Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

MODELING AND SIMULATION OF CONTAINER TERMINAL LOGISTICS SYSTEMS USING HARVARD ARCHITECTURE AND AGENT-BASED COMPUTING

Bin Li

Department of Economics and Management
Fujian University of Technology
University City Campus Road 3
Fuzhou, Fujian 350108, CHINA

Wen-feng Li

School of Logistics Engineering
Wuhan University of Technology
Heping Road 1040
Wuhan, Hubei 430063, CHINA

ABSTRACT

As the highly complex logistics system, container terminal logistics systems (CTLS) play an increasingly important role in modern international logistics, and therefore their scheduling and decision-making process of much significance to the operation and competitiveness of harbors. In this paper, the handling, stacking and transportation in CTLS are regarded as a kind of generalized computing and compared with the working in general computer systems, whereupon the Harvard architecture and agent-based computing paradigm are fused to model the operational processing of CTLS, and the kernel thoughts in computer organization, architecture and operating system are introduced into CTLS to support and evaluate container terminal planning, scheduling and decision-making. A new agile, efficient and robust compound modeling and scheduling methodology for CTLS is obtained consequently. Finally a series of single-vessel simulations on handling and transportation are designed, implemented, performed, evaluated and analyzed, which validate the feasibility and creditability of the systematic methodology effectively.

1 INTRODUCTION

Since introduced in the 1960s, containers represent the standard unit load concept for international freight. Containerized traffic and information network both provide a common basis for international logistics system nowadays. Starting with 50 million twenty-foot equivalent unit (TEU) in 1985, world container turnover has reached more than 350 million TEU in 2004, and the annual growth rate is projected at 10 percent till 2020. Today over 60% of the world's deep-sea general cargo is transported by containers, whereas some routes, especially between economically strong and stable countries, are containerized up to 100% (Steenken, Voss, and Stahlbock 2004). The increasing number of container shipments causes higher demands on the seaport container terminals, container logistics, and management, as well as technical equipments. An increased competition between seaports, especially between geographically close ones, is a result of this development. At the same time, harbors have to gear up to meet the challenge of handling mega-vessels capable of carrying 10,000– 12,000 TEU and beyond (Stahlbock and Voss 2008). To win an advantage in the new round terminal competition, container terminal logistics systems (CTLS) must be systematic rationalized, efficient and robust, and could provide a first-class container logistics handling platform for the loading and unloading of container ships and trucks. The only effective approach to achieve this purpose is optimizing task assignment, resources allocation and scheduling management at container terminals.

Thereupon, much pertinent research has been on the march or educed the corresponding results. For instance, Gunther and Kim (2006) summarized the container traffic, operation and the interrelated planning and scheduling problems. Vis and Koster (2003) gave a classification of the decision problems that

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

arose at container terminals under the background of the ships have been increasing large-scale. Henesey, Davidsson, and Persson (2006), and Kefi, Korbaa, Ghedira, and Yim (2007) applied multi-agent into container terminal scheduling system, which made the scheduling system intelligent and provide the scientific proofs for the scheduling and decision-making of the production and management at container terminals. Lu, Bostel, Dejax, Cai, and Xie (2007) presented an integrated model to schedule the equipment to minimize the service makespan, which was formulated as a hybrid flow shop scheduling problem with precedence and blocking constraints. A tabu search algorithm was proposed to solve this problem. Legato, Gulli, and Trunfio (2008) advanced some queuing network based representations that were on the basis of an integrated simulation model under development and also remarked the benefits of parallel and/or distributed computational frameworks. Boer and Saanen (2008) put emphasis upon terminal operating system (TOS) which supports planning, scheduling and equipment control. They put forward an approach to test and tweak the TOS on a virtual platform. From the above literatures review, we can find that more and more research models and analyses the operation at container terminals from the angle of the holistic to achieve the total system optimization, but not for the part. It is a favorable study trend to CTLS, but hardly any had existed literature to present a set of integrated agile robust compound modeling and scheduling methodology for CTLS.

Aiming at the deficiency of the foregoing study and the current requirements of CTLS, this paper presents to describe and model CTLS using Harvard architecture and agent-based computing (HA-AC) by referring to the kernel ideas of computer organization, architecture and operating system (OS), which is intended to improve the scheduling and decision-making level of CTLS and upgrade the competitiveness of container terminals consequently. All are supposed to bring forward a practical, agile, effective and robust methodology that may model and optimize CTLS, whether for the part or on the whole. So the rest of this paper is organized as follows. Section 2 advances the overall modeling thought of CTLS using HA-AC after analyzing the CTLS operation in brief. Section 3 discusses the structural and operational comparability between CTLS and computer systems (CS) and their mapping relationships. In section 4 the compound modeling and scheduling architecture of CTLS using HA-AC is entered into particulars. Section 5 performs and evaluates a serial of simulations on handling and transportation for a single ship in the CTLS to validate the feasibility and creditability of the above systematic methodology. In section 6, the conclusions and future work is given.

2 OVERALL MODELING THOUGHT

2.1 Operational Analysis of CTLS

Container terminals are the container shipping hubs of land-sea intermodal transportation and also the only switch approach between empty containers and freight ones. Container terminals can be compartmentalized into two main regions in geography: quay side and storage yard. The former is the most crucial part in the whole CTLS and centralizes a mass of handling and transportation equipments, and constructs the combined loading and unloading platform of container ships, which includes handling and transferring of containers. The latter functions as storing and stacking containers temporarily and is a buffer between the loading/unloading of container ships and picking up/delivering of container trucks in substance. The organization and management of storage yard is the most complex portion at container terminals in respect that the import, export and transferring containers want loading and unloading with simultaneity. The optimization of the main resources on the yard, which involve yard space and yard cranes, exerts a direct and important influence on the holistic operational efficiency of CTLS.

CTLS are the highly complex open logistics systems made up of the production section, the scheduling section, and the management information system (including information infrastructure). CTLS comprise multifarious modeling and optimization technology, and include parallel, negotiation and competition relationships in operation. CTLS are also the multistage and multidimensional representative discrete element dynamic systems (DEDS) that are provided with the characteristics that are high random that is for the levity of nature conditions and client requirements, low flexibleness that is for the limited handling space at container terminals, operation time uncertainty of equipments and high harmony among handling facilities, which all demand that container terminals possess of the sound plain layout and the efficient handling technics. Container terminals have turned into the kernel transacting establishments of container logistics and their handling technics also becomes more complicated than ever with the extend of scale. The loading, unloading, stacking and transferring equipments are pertinent to their handling technics. At present, tugboats assist container ships to moor the appointed berth from anchorage ground (AG), quay cranes (QCs) are provided to load and unload container ships on quay side, and yard cranes (YCs) are adopted to handle and stack containers on yard, and yard trailers (YTs) are charged with transferring between quay side and storage yard and among the blocks on the yard, and gate houses (GHs) answer for passing in and out of container trucks (CTs), and CTs fulfill the container collecting and distributing tasks for container terminals. We can describe the above containers handling, stacking and transportation mode from the angle of flexible manufacturing system (FMS), whereupon CTLS can be considered as the agile efficient and robust FMS for all the import, export, and transferring containers. That is driven by the information obtained from cargo agency and shipping agency via electronic data interchange (EDI), wireless sensor network (WSN), radio frequency identification (RFID), harbor portals etc ahead of schedule and stored in the database servers.

2.2 Agent-based Modeling and Simulation for CTLS

Agent and agent-based computing provide a new and often more appropriate paradigm for the development of complex systems in open and dynamic environments, especially multi-agent system (MAS), which has already provided faster and more effective methods of resource allocation in complex surroundings (Luck 2006). At the same time, MAS offers strong models for representing real-world environments with an appropriate degree of complexity and dynamism (Macal 2009). Agent-based modeling and simulation (ABMS) for real-world domains may provide answers to complex physical or social problems that would be otherwise unobtainable, the modeling and scheduling of CTLS could be one good example since its state could be changed by many events and its mathematical models are hardly established due to the high complexity of the equations. ABMS is optimum for the domain that possesses the characteristics of distributed and discrete decision support as Wooldridge (2002) points out. ABMS for CTLS is just a feasible and practical means for as much as they are distributed and DEDS as mentioned above.

So we integrates and fuses the ideology of systems engineering, software engineering and industrial engineering, and presents the following ABMS thinking for CTLS. The handling, stacking and transportation in CTLS are regarded as a sort of generalized computing, and the classical precise computational and distributed control architecture --- Harvard architecture and the rapidly developing distributed artificial intelligence (DAI) modeling paradigm --- agent-based computing (HA-AC) are synchronized to describe CTLS to obtain a new, agile, efficient and robust modeling and scheduling methodology for CTLS. Since ABMS is a kind of bottom-up modeling method, we need firstly set up the every individual agent model and adopt the appropriate MAS architecture to assemble the individual agents subsequently to establish the MAS model ultimately. Now we bring forward the agent classification in this section, as for which architecture is going to be applied to assemble the agents and why will be discussed in the next section.

Since we try to model CTLS with a kind of computer architecture, the design ideas of computer organization, architecture and operating system (OS) are used to disassembled CTLS for reference spontaneously. We can discover that the developing history and trend of computer systems (CS) is an evolution process from centralized serial control to distributed concurrency control to a great extent. Now the control units in CS not only include central processing unit (CPU) but also cover interrupt system, graphic processing unit (GPU), channel adapter, and direct memory access (DMA) etc, which aim at that various control units can management the computation, memory and input/output load in CS apart and be absorbed in the respective nuclear task and improve CS holistic operational efficiency. So we decompose CTLS into two kind agents: resources control decision agents and resources task execution agents according to the principles of computer organization and architecture. The former involves AG management agent, tugboat dispatching agent, berth allocation agent, QC scheduling agent, YT dispatching agent, yard allotment agent, YC scheduling agent, block control agent and GH management agent. The latter include AG agent, tugboat agent, berth agent, QC agent, YT agent, block agent, YC agent, bay agent and GH agent. In addition, the main service objects of CTLS are containers, container trucks and container ships, which can also be regarded as the corresponding agents to attend the interaction operation of MAS for CTLS. That is to say, the whole CTLS using HA-AC can be applied in the open and dynamic environments, which is of much significance to the methodology is fulfilled into practice, but not just in theory.

3 SYSTEM COMPARISON ANALYSIS BETWEEN CTLS AND CS

3.1 Comparison in Systematic Hierarchy Model

CTLS are the central transacting platforms of container shipping in the international container logistics network just as CS are the core processing nodes in the Internet. If CTLS systematic hierarchy model in logic concept is contrasted with CS one in particular, the following conclusions may be drawn: first, container terminals plane layout, handling, stacking, transportation equipments and the corresponding operational order set strongly resemble hardware system in CS, which is composed of physical devices, micro architecture and machine language in respect that both provide the working physical basis for respective system and determine system configuration and architecture at root. Secondly, container terminal scheduling systems (CTSS) can be seen as operating system (OS) in CS because both of them improve the system efficiency through sound and effective task organization and resource allotment. Finally, container terminal management information system (CTMIS) including business intelligence (BI) platform may be looked upon as application software in CS, because it can provide the friendly interface to the users based on the hardware system and OS and make it easy to observe, supervise and make the most of the system. The above systematic hierarchy mapping relation is illustrated with Figure 1.

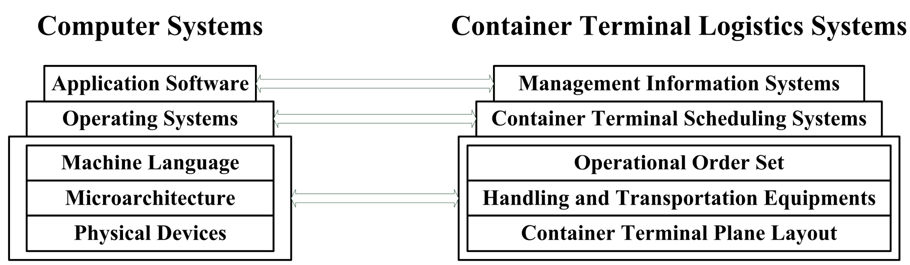


Figure 1: Systematic hierarchy model contrast between CS and CTLS

One important viewpoint must be placed emphasis on in Figure 1. The relations between application software, system software and hardware system in CS are typical vertical calling support. That is to say, application software calls the service provided by system software, and system software drives hardware system to fulfill the material task. By analogy, CTMIS constitutes production planning in advance based on the information on arrival ships, trucks and containers by the corresponding information collecting means. CTSS organizes the jobs that are appointed by CTMIS, and fulfill real-time scheduling to drive various handling, stacking and transportation resources and equipments at container terminal plain layout to perform different container logistics task according to the production plan and the service objects situations. Obviously, CTMIS and CTSS are the information flow receiver, transmitter and processor of CTLS, and container terminal geographical location and plane layout are the operational fundamentality of CTLS, and all kinds of loading, unloading, stacking, transportation mechanical equipments and resources are the container logistics sensors and actuators of CTLS, moreover, form the different functional units.

3.2 Comparison in Systematic Architecture

Almost all CS are based on Von Neumann concept and architecture, which indicates that calculator, controller, memory, input and output device five constitute the whole CS. The memorizer is the systematic center whether in architecture or in function, furthermore, and the same memory can store not only data but also instructions. It is obvious that the storage mode has limitations and goes against the improvement on system efficiency, which is also not propitious to describe and model CTLS distinctly. Whereupon a new kind of architecture, namely, Harvard architecture is proposed, which stores data and instructions in different memory and access them by diverse bus. Based on Harvard architecture, the system components and their mutual relationships of CTLS are illustrated by Figure 2. Container terminals are considered as open and efficient container logistics hubs with two input and output interfaces. One is quay side, the other is gate house. The former bears responsibility for handling task of container ship, the latter answers for fulfilling the collecting and distributing assignment of containers. Container terminal storage yard is a working buffer between container ship handling and container fetching/distributing in fact. That is to say, memorizer and storage yard is the operational centre of the respective system whether in architecture or in function. Moreover, if container logistics in CTLS are mapped as data stream in CS, and information flow are reflected instruction stream in CS, and the handling, stacking, transportation are considered as a sort of generalized computation, upon that the whole CTLS may be regarded as a macroscopic and physical container logistics and information flow processing systems.

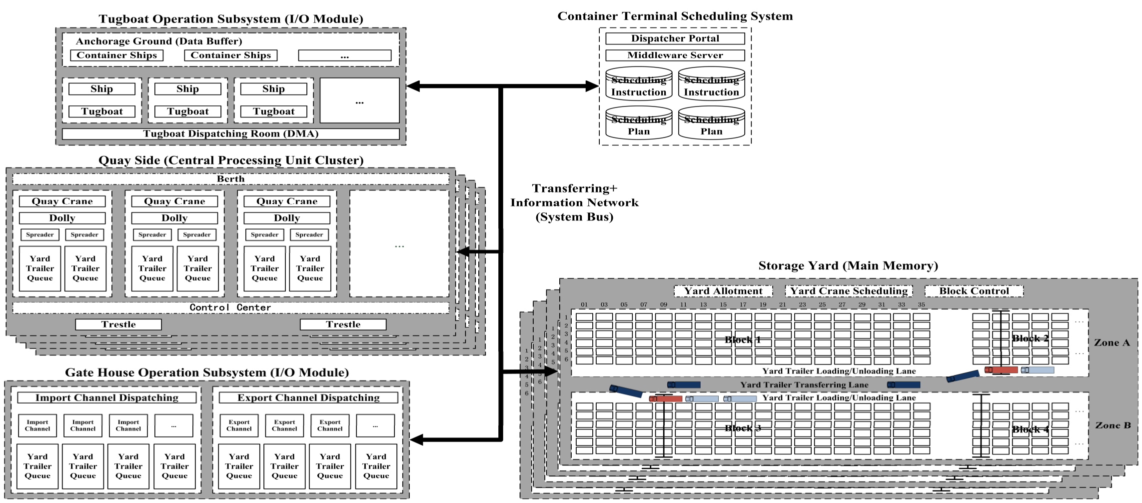


Figure 2: Top view of CTLS organization and architecture

The central control room can be looked upon as controller of CTLS and each berth is regarded as one CPU, thereupon quay side with several QCs can be considered as CPU cluster that is in charge of loading and unloading of arrival container ships, Whereupon YTs, quay trestles and roadways into container terminals form bus systems of CTLS. Container storage yards are just main data memory of CTLS that accomplishes stacking and storage of import, export and transferring containers by cooperating in harmony with YCs. The database servers in CTMIS are the main instruction memory. Container ships, anchorage ground, tugboats, container trucks, gate house and so on are the input/output devices of CTLS that take full responsibility for container getting in/out of harbors. The peripheral equipments also form the container logistics import and export port of CTLS, by connecting the collecting and distributing subsystems of container terminals. One keystone to emphasize on is that the entire CTLS are just a multiprocessor and multi-core container logistics processing systems. This summary can be further comprehended from the sequent three facets, firstly the berth equipped with one QC can be taken for mononuclear container logistics processor, and the berth equipped with two QCs can be regarded as binuclear container logistics processor, and the berth equipped with three QCs can be considered as trinuclear container logistics processor and so on. Secondly, CTLS win the evident advantage over traditional multiprocessor and multi-core CS because QCs can be moved from one berth to adjacent berth in order to improve on working efficiency and holistic throughput of CTLS. Thirdly, it is a clear development trend in the container terminal handling technics that one QC furnished multi-spreader, and each spreader may be equivalent to one thread of processing core in CPU. The QC equipped with mono-spreader can be taken for single-threading processing core, and the QC equipped with bi-spreader can be regarded as hyper-threading processing core, and the QC equipped with tri-spreader can be considered as triple-thread processing core, and so on. So CTLS are the multi-processor and multi-core container logistics processing systems that are parallel computing platform in nature.

Therefore CTLS are provided with a typical multiprocessor parallel architecture. In computer science domain, people usually adopt Flynn classification, which is based on that the processing element with the most constraints in CPU holds the parallel degree of instructions calling data flow and instruction stream, to sort the computer systems four kinds: SISD, SIMD, MISD and MIMD. That classification method establishes a rudimental framework for computer architecture design. As MIMD model can implement the thread level parallelism mechanism, MIMD becomes the chief choice of multi-processor system. Besides, there are two knockout causes, one is that the MIMD takes on outstanding flexibility, the other is that MIMD may make full use of the ratio of performance to price predominance of the existing microprocessors (Henenessy and Patterson 2007). CTLS just accord with MIMD model in the Flynn classification framework, namely, CTLS are the representative multi- instruction stream and multi- data flow systems. As each berth is a CPU with the corresponding task on the arrival container ships, the whole CTLS are the MIMD system in the macroscopic scale. Since each quay crane that are the handling core at berths are seized of instruction stream and container logistics respectively by reason of vessel loading depot bills, container handling and transportation subsystems at each berth is a MIMD system, namely, CTLS are the MIMD system in the medium measure. For there are several spreaders in a quay crane, upon that there are many container logistics and instruction stream, CTLS are the MIMD system in the microcosmic level. In brief, CTLS are the pure multilevel MIMD system, so their scheduling and decision-making are highly complex problems.

3.3 Comparison in Systematic Operation

CTSS can be considered as OS in CTLS as its principal function is also improving the system operational efficiency and the holistic throughput by efficient and rational system resources management and allotment. CTLS can be carved up five task subsystems by analysis of CTLS organization model: anchorage grounds and tugboats, quay side, horizontal transferring, storage yard, and harbor gate. Now we introduce the main functions of OS into the five subsystems in CTLS and bring forward container terminal scheduling hierarchical structure according to system organization and operational characteristics by Figure 3.

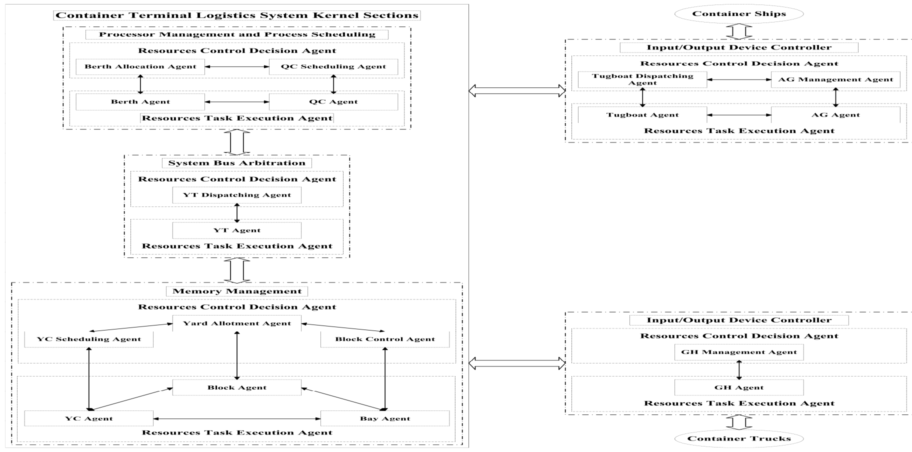


Figure 3: CTLS scheduling administrative levels

Obviously, the scheduling at quay side, in transferring and on storage yard are the core parts of CTLS operation and can determine the traffic and throughput capacity of CTLS directly. On the other hand, the periphery anchorage ground allotment, tugboat dispatching and gate house management exert an great influence on the running of CTLS. If anchorage ground and tugboats are scheduled improperly, the container ships berthing and departing efficiency will descend, which has an negative impact on the combined loading and unloading at quay side to debase the traffic capacity of CTLS. And if the import and/or export channels under gate house are not supervised validly, the container terminal collecting and distributing system is going to be weakened, which may also cut down the throughput capacity of CTLS.

4 COMPOUND MODELING ARCHITECTURE OF CTLS USING HA-AC

Based on the above-mentioned modeling thinking, agent classification and the structural and operational comparability between CS and CTLS, we present the modeling architecture of CTLS using HA-AC whose holistic framework and interaction structure are showed in Figure 4. Here CTLS are regarded as a “huge” data processing systems, namely, quay side and the interrelated equipments are the “central processing unit cluster”, and storage yard and the correlative facilities are the “main memory”, and anchorage ground, tugboats and gate house are the “input/output equipments”. The yard trailers, roadway and traffic rules in port constitute the container logistics bus, and the information infrastructure network forms the information flow bus, which both structure the system bus and the backbone of CTLS. This modeling thinking and framework is also a extension and expansion of the thinking of computer integrated manufacturing system (CIMS) which is intended to integrate the software and hardware resource across-the-board. It makes full use of the precise computational structure to model the complex system based on multi-agent, thereby it serves the turn to the queuing theory and FMS inherently and possesses the top-ranking flexibility, agility and robustness. This architecture makes it easy to set down production planning and real-time scheduling, moreover, and can support the important decision-making in harbors.

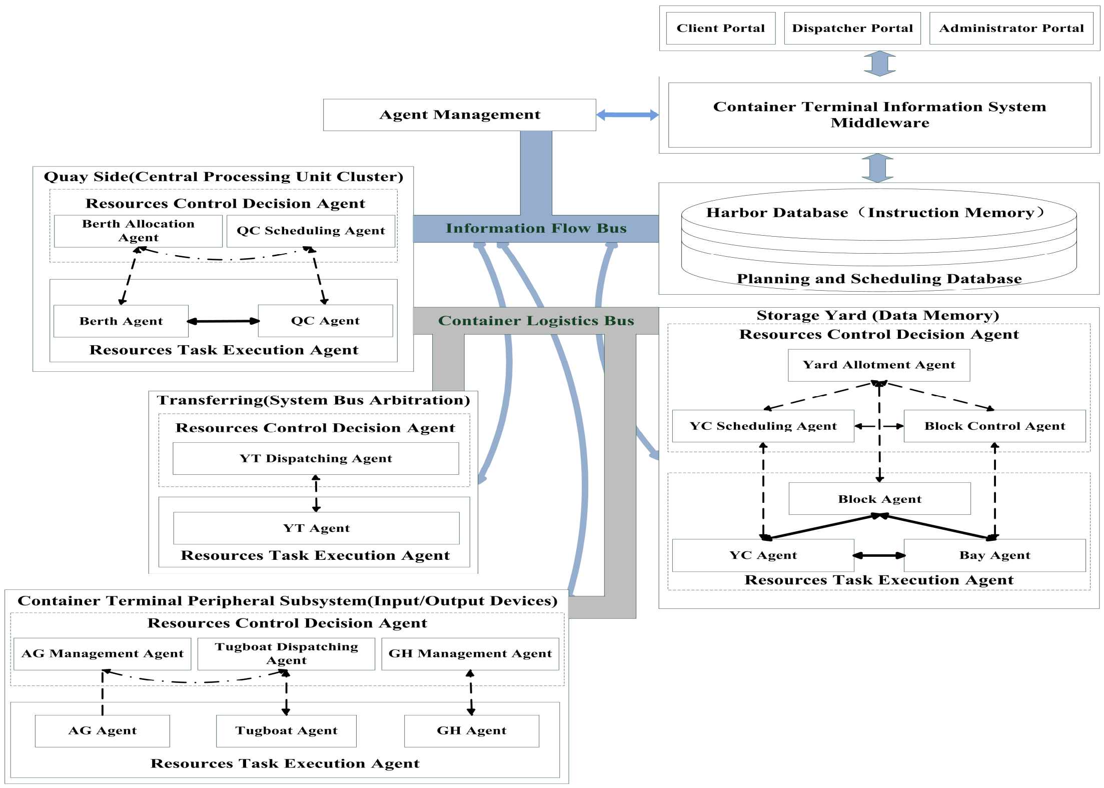


Figure 4: Modeling framework of CTLS using HA-AC

After setting up the holistic model framework of CTLS based on the Harvard architecture, we mine the scheduling and optimization mechanism in computer science to describe and optimize CTSS. CPU is the most important resource in CS, and an intriguing point of upgrading its speed is just parallel processing. Most modern CPUs utilize superscalar architecture and pipeline operation to improve the performance, and therefore the ideas of superscalar and pipelining are considered as the core modeling ideas in CTSS. The fusion of superscalar architecture and pipeline operations can be abstracted as hybrid flow shop scheduling problem with blocking in mathematical model. On the other hand, the parameters on processed workpieces (vessels and containers) in the flow shop can be seen as the attributes of workpieces and have a strong impact on container terminal scheduling. So we use the hybrid flow shop scheduling problem with blocking based on attributes (HFS-BA) to model CTSS and present an integrated bi-level scheduling model, which is shown in Figure 5. The whole model is made up of three parts: the upper-level scheduling model, the lower-level one and the public communication system. The upper-level is the quay side scheduling model, which includes the AG, tugboats, berths, QCs etc., loading and unloading facilities and resources and provides the services for container ships. Minimizing the total time in harbor of arriving vessels and maximizing the revenues of the port or terminal operators are the ultimate objectives of the upper-level scheduling. The lower-level is a transfer dispatching model, which involves QC, YT, YC, yard etc. equipments and resources. In the same manner, minimizing the total handling time is the ultimate objective of the lower-level dispatching. The public communication system is composed of the blackboard system and the mailbox system, so is implemented on top of a database (DB). One point needs to be stressed: there are five processes or work pieces which may be referred to as flow shops for scheduling the container ships and the containers respectively. However, only eight kinds of equipment and resources are set in the ten phase scheduling. This is because tugboats require scheduling twice and the QC in the upper-level and the lower-level are the same set, i.e.. The DB will track record and update two sorts of contents on-line. One is the task information which is concerned with the arriving vessels and the corresponding handling of containers; the other records the busy/idle states and functional parameters of the production resources and equipment at the terminal which is an important basis for scheduling. The approach and solution for container terminal scheduling are based on the feedback and reciprocity between the two sub-models, where the container ships and corresponding containers are work pieces respectively, As the bridge between two kinds of work pieces, QC is the joint of the bi-level scheduling model, and its allocation is of utmost significance for the overall scheduling.

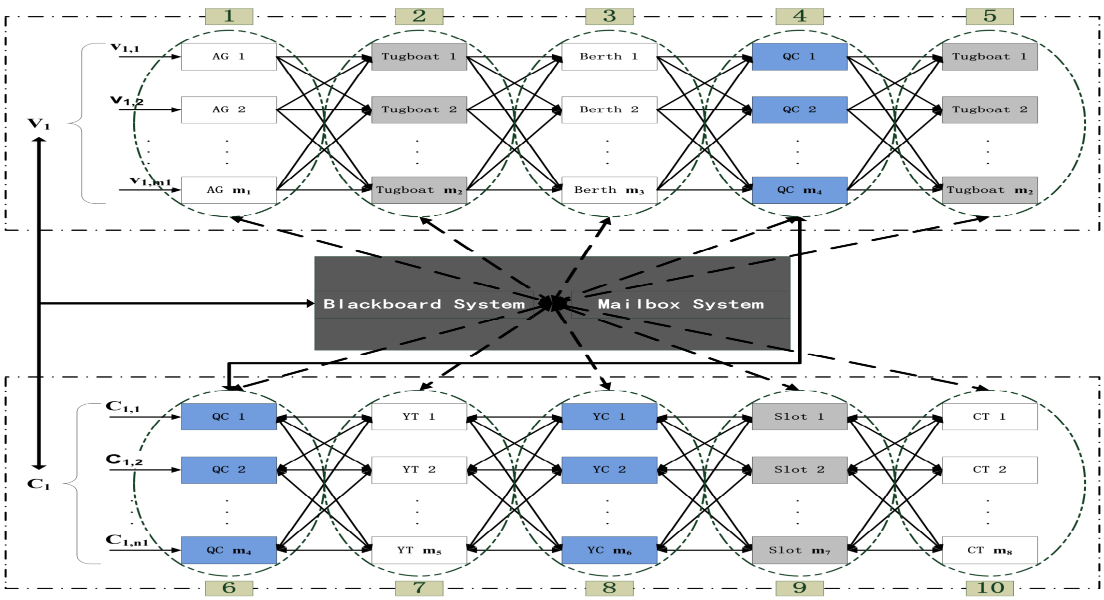


Figure 5: Bi-level scheduling model for CTLS based on HFS-BA

5 SIMULATION AND ANALYSIS

5.1 Problem & Scenario

The above modeling architecture can be exemplified by a simulation scenario of handling and transportation of a single ship at a container terminal. Suppose that the vessel has been moored to a berth, the pertinent QC have been situated on the appointed berth, and all the handled containers are export containers and have been collected on the yard. This is representative of dispatching for containers as the scheduling on container ships has been specified. The involved resources and equipments at the container terminal are described as follows: The whole storage capacity is about 22000 TEU and the storage yard consists of 36 stacks of containers and is divided into two sections as shown in Figure 6. The import storage area where the import containers are stored is closer to the gate and the export storage area is closer to the quay side. Each stack has 216 containers when containers are stacked 3-high and 288 if stacked 4-high. Assuming stacking of 4-high, the maximum capacity of the storage yard is 10,368 containers. We assume that containers are 40-feet which gives a total capacity for the storage yard of 20,736 TEU. In addition to the storage yard, containers can also be stored at the gate buffer whose maximum storage capacity is 1,728 TEU giving a total storage capacity for the terminal of 22,464 TEU. The terminal dimensions are calculated to be 1,633*1,875 ft^2 (70.29 acres). As for the handling facilities, there are six QC on the appointed berth and each of them is assumed to be able to perform up to 42 moves per hour per crane for combined loading/unloading. there is one YC in each container stack and each is assumed to perform up to 37 moves per hour per crane. In addition, there are 42 YT for transfers. The number of export containers is 300 and the containers are distributed in the export container storage whose amount is 18. The speeds of empty and loaded YT are 16 km/h and 8km/h respectively. The layout and handling equipment of this terminal is a very typical one to most terminals around the world.

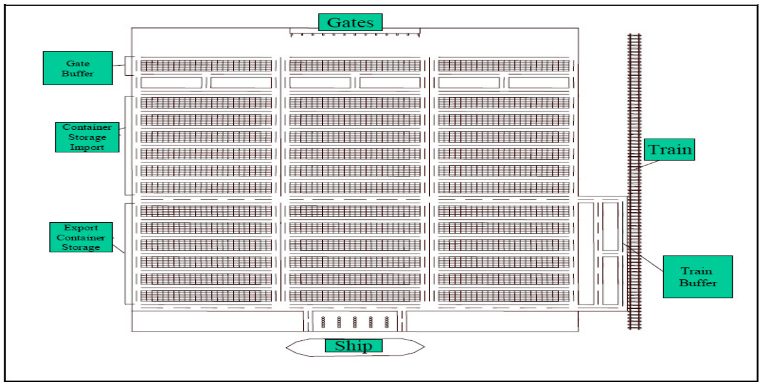


Figure 6: Container terminal layout on simulation instance

5.2 Design and Implement

We simulate the container terminal handling and scheduling system on an advanced dynamic simulation platform AnyLogic 6.5.0. AnyLogic is based on Java and the Eclipse framework that make it possess of outstanding open and compatibility, and its script language is Java too, which brings sufficient flexibility and enables the user to capture the complexity and heterogeneity of business, economic and social systems at any desirable level of detail.

To 5.1 simulation instance, the above compound modeling architecture can be predigested to be a yard trailer dynamic dispatching (YTDD) problem, where the configuration of QC and YC have been confirmed. So the involved nuclear agents in the model are QC agents, YC agents, YT agents and YT dispatching agent in Figure 4, and that the corresponding working procedures are just 6, 7 and 8 in Figure 5. we adopt the fundamentals law of simulation implement, which can be summarized as follows: the practical operational data stored in the database are engaged to drive the simulation, and the blackboard and mailbox system and message/conversation are fused to form the communication and interaction mechanism among agents, and the resource allocation policies and algorithms in OS are introduced to dispatching handling equipments. In this instance, we adopt semaphore mechanism to manage the correlative resources on the understanding that the other conditions hold the line. Here only scheduling principles are emphasized on. We model the single ship handling and transportation using Harvard architecture, and then we introduce the resource allocation policies and algorithms of OS to schedule yard trailer. In our modeling thought, the YT are resource in substance and the exclusive equipments for the scheduling of yard trailers the semaphore mechanism is introduced which is a highly effective tool in OS for the synchronization and mutex of process. We set up the resource record table in background database to note the idle and busy status of YT. The yard trailers are noted by the semaphore with integer variables. The physical meaning of semaphore is the usable number for yard trailer resources when the semaphore is greater than or equal to zero; when the semaphore is smaller than zero the absolute value of the semaphore is the blocked yard trailer requests. In the practical handling at container terminals, we should avoid to have QC and YC suspended for waiting the yard trailers, especially QC. When the waiting yard trailers queue length under QC or YC is smaller than certain valve value, the QC or YC sends a request for yard trailers. By doing So the possibility that QC or YC waits the yard trailers for transferring will be very low. In the case of export containers, the yard trailers are dispatched to YC according to a rank of these YC, which is the descending sort of the differences between the waiting yard trailer queue length and the corresponding request numbers of each YC. (Li, Li, and Voss 2009)This scheduling method could also be used in The case of import containers. In scheduling simulation, there are a mass of parallel requests and read-write to database whether in the simulation or the communication between AnyLogic and database (SQL

Server 2008). In the simulation, we define the relevant dynamic events that are defined in AnyLogic in advance and multiple instances of the same dynamic event can be scheduled concurrently in the model. SQL Server 2008 provides the powerful mechanism to guarantee the rationality of parallel read-write between AnyLogic and background database as the above mentions.

5.3 Result and Evaluation

Zhang and Wang (2008) simulates the same instance on the eM-Plant. The ultimate completion time in static scheduling policy is two hours eleven minutes and twenty seven seconds, and the time in dynamic scheduling based on genetic algorithm (GA) is one hour fifty six minutes and thirty one seconds. We simulate the instance by utilizing HA-AC architecture on the AnyLogic 6.5.0 platform. Considering the need of pre-stowage plan in practice, the handling number of each quay crane is 52, 49, 52, 43, 52 and 52 respectively, and the number of waiting handled containers in 18 yard block is 18, 18, 17, 14, 18, 18, 18, 9, 18, 18, 16, 16, 18, 18, 18, 17, 16 and 15 respectively, and the initialization distributing of YT in 18 yard block is 1, 4, 4, 1, 4, 4, 2, 2, 1, 3, 1, 2, 4, 0, 0, 4, 3 and 2 in the following simulation. We regulate the semaphore quantity from 1 to 5, and get the simulation interface as Figure 7 and the relevant result as Table 1 shows. The following pertinent simulation time all is expressed in the format of Hour: minute: second.

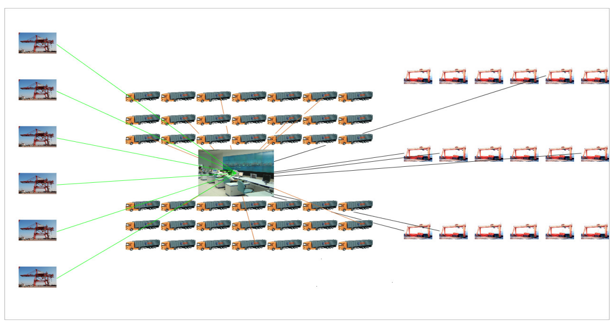
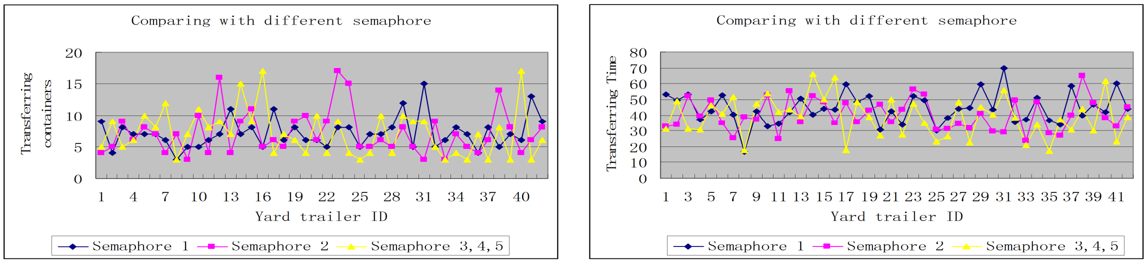


Figure 7: Simulation interface using AnyLogic

Table 1: Simulation result

Semaphore Quantity	Completion Time	Average Waiting YT under QC	Average Waiting YT under YC	Max Waiting YT under QC	Max Waiting YT under YC
1	01:43:51	2.804	1.154	8	3
2	01:50:00	2.37	1.31	8	3
3	02:05:00	1.979	1.463	8	3
4	02:05:00	1.979	1.463	8	3
5	02:05:00	1.979	1.463	8	3

Through the above simulation, we can conclude that the scheduling based on Harvard architecture and semaphore excels yard trailers static scheduling obviously and gain the advantage over dynamic scheduling based on GA in the completion time, moreover, the max waiting yard trailers under QCs and YCs is not much more than the yard trailers static scheduling and is less than dynamic scheduling based on GA distinctly. The transferring containers and time of every yard trailer are illustrated by Figure 8. The data is same to the semaphore 3, 4 and 5 in the simulation.



(1) Transferring containers using different semaphore **(2) Transferring time using different semaphore**

Figure 8: Comparison of transferring containers and time in different semaphore

From Figure 8, we can find that the effective transferring time of YT are 30 hours fifty five minutes and thirty two seconds, 27 hours fifty minutes and four seconds, and 26 hours forty minutes and twenty seven seconds apart on the understanding that the semaphore is 1, 2 and 3 (4,5). It is evident that the whole transferring time is seized of the maximal effective transferring time, which means that the YT resource is utilized most adequately. The maximal transferring container quantity of single yard trailer is 14, 16 and 18 when the semaphore is 1, 2 and 3 (4,5), and the minimal one all is 3, the standard deviation is 2.543, 3.354 and 3.68 respectively. With the above statistical data, we can conclude that the minimal holistic transferring time is the minimal standard deviation in respect that the task load on every yard trailer is the most balanced. This point can be deduced from the effective transferring time too. The maximal effective transferring time is 62.79, 66.98 and 74.81 minutes on condition that the semaphore is 1, 2 and 3 (4,5) apart, the minimal one is 16.88, 23.44 and 17.29 minutes, the standard deviation is 9.942, 10.285 and 12.806. It is same that the minimal standard deviation gets the minimal holistic transferring time. It is indicated that the equilibrium load to the handling and transportation facilities and resources is of much significance to CTLS, which is different in approach but equally satisfactory in result with the operation in CS and imply the comparability between CTLS and CS again.

5.4 Simulation Analysis

In 5.3 section, we simulate and evaluate a task instance on the basis of the practical case. Now we make the instance more generalized by introducing a dynamic rule for dispatching the yard trailers, which is dependent on whether the containers are assigned to QC or YC with equilibrium and whether the trailers are distributed to each block with equality. This generalized instance is a sensitive analysis for 5.3 simulation, where the handling task is still 300 export containers. there are eight situations depending on whether the QC tasks, YC tasks and YT are distributed uniformly or not. We set up semaphore as 2 and start the simulation with 512 MB the maximum available memory of AnyLogic, and it runs1000 times for each situation. the following result is obtained as Table 2 shows.

Table 2: Comparison of Simulation Result 1

Simulation group	Task in QC uniformly	Task in YC uniformly	YT distributing uniformly	Minimal transferring time	Maximal transferring time	Average transferring time	Standard deviation
1	No	No	No	1:24:51	2:26:00	1:42:16	11.096
2	Yes	No	No	1:23:00	2:28:00	1:40:46	10.528
3	No	Yes	No	1:24:43	2:17:00	1:40:03	9.124
4	Yes	Yes	No	1:28:00	2:27:00	1:52:08	11.075
5	No	No	Yes	1:21:51	1:35:00	1:26:31	1.819
6	Yes	No	Yes	1:21:00	1:31:00	1:25:49	1.637
7	No	Yes	Yes	1:20:09	1:33:34	1:24:36	1.851
8	Yes	Yes	Yes	1:24:26	1:24:26	1:24:26	0

Based on the data in Table 2, we can conclude that the initialization distribution of YT is of much significance to YTDD by contrast with eight group experimental data. If yard trailers are scattered on the yard zone with uniformity, the final simulation result is identical and optimal, at least much better than the average scheduling, moreover, this result does not change with the distribution of handled task.

It is almost same idea to have balanced load in substance, which is an important YTDD rule for the transferring at container terminals. Under this rule, we set the quantity of semaphore from 1 to 5 and run the simulation again. The simulation results of 1000*2*5 times, which are according to the different semaphore value and the task is assigned by QC and YC equably or not, show the parallel ultimateness. Here the second line is the case that the task is not allotted to QC and YC equably and the third line is the case that the task is distributed to QC and YC with uniformity. One point to be mentioned here the experimental results do not change when the semaphore quantity is greater than or equal to 2, as the quantity of YT divides the quantity of YC is just 2 (reserving integer section).

Table 3: Comparison of Simulation Result 2

Semaphore Quantity	1	2	3	4	5
Average Completion Time 1	1:27:44	1:26:31	1:26:31	1:26:31	1:26:31
Average Completion Time 2	1:19:26	1:24:26	1:24:26	1:24:26	1:24:26

5.5 Further Discussion

Due to the container ship of increasing large-scale and quick, CTLS are required to handle arrival of ships as soon as possible. We still adopt 5.1 section container terminal plane layout and equipment configuration, moreover set the value of semaphore as 2. The simulation remains the same to load export containers, but the quantity is not 300, but 1500. According to 5.4 section analysis, YT and export containers are in equilibrium on all the blocks. Though the demand of pre-stowage plan, every quay crane acquires different working load, we simulate the instance with 1000 times and obtain the following result as Figure 9 shows. The maximal YT transferring time is seven hours forty nine minutes and twenty six seconds, the minimal one is six hours fifty two minutes and zero seconds, the average one is seven hours fourteen minutes and forty seven seconds, the standard deviation is 8.31. If the export containers are still in equilibrium on all the blocks, but the number of YT is adjusted to 20, 25, 30, 35, 40 and 45, and we simulate the each cases with 1000 times and obtain the following result as Table 4 shows.

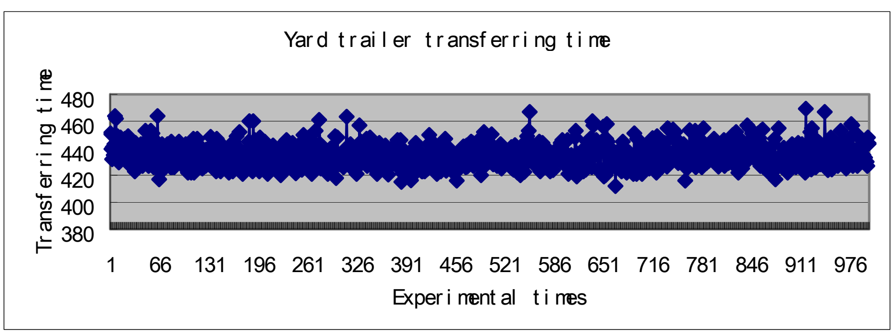


Figure 9: Simulation result for 1500 containers

Table 4: Comparison of Simulation Result 3

YT Quantity	Minimal Transferring Time	Maximal Transferring Time	Average Transferring Time	Standard Deviation
20	12:04:00	14:49:00	13:25:39	27.246
25	09:54:00	12:58:00	11:06:57	23.752
30	09:03:00	13:24:00	10:23:59	30.549
35	09:17:00	11:21:00	10:27:44	18.316
40	07:21:00	08:10:00	07:41:46	7.931
42	06:52:00	07:49:26	07:14:47	8.31
45	06:57:00	07:37:00	07:13:26	6.251

We utilize computer architecture to model CTLS, so we introduce one of computer quantitative approaches --- Amdahl's Law to give the optimization configuration of YT. Amdahl's Law states that the performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used, which also defines speedup that tells us how much faster a task will run using the computer with the enhancement as opposed to the original computer. If the yard trailer is regarded as transferring processor, the average transferring performance of 20 YT is regarded as evaluation benchmark, upon that the speedup is 1.14267, 1.10575, 1.30546, 1.81425, 1.89379 and 1.94525 while the number of YT is 25, 30, 35, 40, 42 and 45. Through the data, we can discover that the transferring performance is improved greatly when the number of YT is increased from 35 to 40, but when the number of YT keeps on enhancing, the improvement on transferring performance is very limited. So the number of YT is between 40 and 45 is rational and efficient to the above container terminal.

6 CONCLUSIONS AND FUTURE WORK

In the context of the increasing large-scale and high-speed of container ships and the cut-throat competition among harbors day by day, the scheduling and decision-making level of CTLS is of great significance to improve service competence and competitiveness of container terminals. This paper syncretizes the classical precise computational framework---Harvard architecture and the rapid developing agent-based computing to form the new and promising compound modeling methodology for CTLS. With the development of pervasive computing and automated container terminals, the container logistics and information flow are synchronized absolutely, the methodology am going to be further studied and applied. The proposed methodology could be developed into a practical and powerful intelligent decision support system (IDSS) for CTLS. This platform can integrate the total resources of harbors and incarnate the kernel of CIMS, which is applicable to the local and holistic of CTLS operation. This is supposed not only to help dispatcher to constitute the production plan and perform real-time scheduling daily but also to support the momentous strategic decision-making in the momentous occasions.

ACKNOWLEDGMENTS

This work is partially supported by the National Key Technology R&D Program of China during the 11th Five-Year Plan Period Grant #2006BAH02A06 and the Scientific Research Foundation for High Level Talents of Fujian University of Technology in China Grant #GY-Z10005.

REFERENCES

Li, B., W. F. Li, and S. Voss. 2009. Modeling container terminal scheduling systems as hybrid flow shops with blocking based on attributes. In *Logistik Management*, 413-434. Berlin: Springer-Verlag.

Boer, C. A., and Y. Saanen. 2008. Controls: Emulation to improve the performance of container terminals. In *Proceedings of the 2008 Winter Simulation Conference*, ed. M. E. Kuhl, N. M. Steiger, F. B.

Armstrong, and J. A. Joines, 2639–2647. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Macal, C. M., and M. J. North. 2009. Agent-based modeling and simulation. In *Proceedings of the 2009 Winter Simulation Conference*, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 2639–2647. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Steenken, D., S. Voss, and R. Stahlbock. 2004. Container terminal operation and operations research – a classification and literature review, *OR Spectrum*, Vol. 26, No. 1.

Gunther, H. O., and K. H. Kim. 2006. Container terminals and terminal operations, *OR Spectrum*, Vol. 28, No. 4.

Vis, F.A. and R. D. Koster. 2003. Transshipment of containers at a container terminal: An overview, *European Journal of Operational Research*, Vol. 147, No. 1.

Henenessy, J. L., and D. A. Patterson. 2007. *Computer architecture: A quantitative approach*. 4th ed. Maryland Heights: Elsevier Science.

Henesey, L., P. Davidsson, and J. A. Persson. 2006. Agent based simulation architecture for evaluating operational policies in transshipping containers, *Lecture Notes in Computer Science*, Vol. 4196. 73-85. Berlin Heidelberg: Springer-Verlag.

Lu, C., N. Bostel, P. Dejax, J. G. Cai, and L. F. Xie. 2007. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*, Vol. 181, No. 1.

Kefi, M., O. Korbaa, K. Ghedira, and P. Yim. 2007. Container Handling using multi-agent architecture, *Lecture Notes in Computer Science*, Vol. 4496. 685-693. Berlin Heidelberg: Springer-Verlag.

Luck, M. 2006. Agent-based computing. *GEOconnexion International*, Vol. 5, No. 5.

Legato, P., D. Gulli, and R. Trunfio. 2008. Simulation at a maritime container terminal: Models and computational frameworks. In *Proceeding of 22nd European Conference on Modeling and Simulation*, 261-269. European Council for Modelling and Simulation.

Stahlbock, R., and S. Voss. 2008. Operations research at container terminals: a literature update, *OR Spectrum*, Vol. 30, No. 1.

Wooldridge M. 2002. *An introduction to multi agent systems*. West Sussex, England: John Wiley and Sons Ltd.

Zhang, Y., and S. Wang. 2008. The research of trailer scheduling based on the hybrid flow shop problem with blocking. In *Proceedings of the 7th World Congress on Intelligent Control and Automation*, 3936-3940. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

BIN LI is a lecturer in the Department of Economics and Management at Fujian University of Technology in China. He received a Master of Software Engineering from Huazhong University of Science and Technology, Wuhan, China, and a Doctor of Mechanical Manufacture and Automation from Wuhan University of Technology, Wuhan, China. His research interest is in the area of discrete event modeling, simulation and optimization applied for manufacturing industries and complex logistics system, computational intelligence, pervasive computing and software engineering. His email address is <mse2007_lb@whut.edu.cn>.

WEN-FENG LI is a Professor and subdecanal in the School of Logistics Engineering at the Wuhan University of Technology. He received a Doctor of Mechanical Design and Theory from Wuhan University of Technology, Wuhan, China. His research interests are in complex logistics system modeling and simulation, wireless sensor network and robot. His email address is <liwf_cn@126.com>.