Multi-Agent System Model for Container Management Simulation

Houda Zouhaier[1], Fahem Kebair[1], Frédéric Serin[2] and Lamjed Ben Said[1]
[1]*Laboratoire SOIE, Higher Institute of Management of Tunis, Le Bardo, Tunis, Tunisia*
[2]*Laboratoire LITIS, University of Le Havre, Le Havre, France*

Keywords: ATN, Container Management, Multi-Agent System, MESSAGE/UML, Simulation.

Abstract: This paper discusses an approach to build a multiagent system for simulating container management in a hub port logistics. The simulator has as goal to help assessing and defining container management strategies. This allows to plan and to control the management of containers while minimizing the waiting time and the parasite shifts and insuring the consistency of the performed tasks sequence. The proposed model involves the multi-point of view and the emergence of behavior specific to the theory of complex systems. The paper is structured as follows: first we present related works, then we expose the multiagent model of the simulator, after that we present the internal structure of the agents and finally we provide and discuss first implementation and results.

1 INTRODUCTION

Supply chain is considered as a network in which a range of actors involved throughout the chain between the origin point (suppliers) and the destination point (clients) for manufacturing goods taking into account many constraints in time and space. Tasks of the chain must be designed to satisfy the needs of customers. In the delivery phase, goods are redirected through a network of intermodal transport from a logistics hub to another until their destinations at the lowest cost and as soon as possible (Govil and Proth, 2001). A logistics hub treats thousands of daily shifts of which management should be optimized as for the internal movements performed by machines. A shift includes three operations: picking up, transporting and putting-on container. The implementation of a strategy for management and decision making is necessary therefore in order to ensure container treatment.

Decisions for container management take into consideration several constraints such as customer requests (turn-around time, low risk delivery and a minimum of handling possible) and resource availability. To improve the port hub performance, decisions are made by evaluating various planning policies. In this field, researchers have defined some productivity indicators for evaluating the decisions performance as service time[1], container terminal capacity, berth utilization, waiting time[2] and dwell time[3], etc. (Henesey, 2006). A logistics hub is characterized by its dynamicity and the intervention of several actors (eg. containers, engines, planner, etc.). The dynamicity is defined by the movements number that can occur, the containers arrival and departure, the unexpected appearance of new missions, etc. (Psaraftis, 1995).

To deal with this dynamicity and given the necessity of coordinating between different actors, we propose in this paper a distributed approach using the Multiagent Systems (MAS) technology (Wooldridge, 2009). This approach allows us to share the plan for tasks accomplishment and to treat the dynamic aspect of containers in the terminal. The multi-agent approach proved their efficacity in container management for port terminals. Indeed, Bin in (Bin et al., 2011), proposes a MAS model of container terminal scheduling and management system, focusing on how to coordinate the scheduling of different resources in a container terminal. Lawrence in (Henesey et al., 2003) describes an approach (supported with MAS-CommonKADS) enabling decision makers to simulate various port policies and analyze the multitude of "what if" scenarios to model the system architecture and the role of different parts which compose it. We cite also the work of Kefi in(Kefi et al., 2009)who uses an informed algorithm to describe the behaviour of

[1]Service time is a period of time in which the ship is berthed. (Henesey, 2006)

[2]waiting time is time during which the vessel must wait for an available berth. (Henesey, 2006)

[3]dwell time is the time spent by the container in the port. (Henesey, 2006)

Zouhaier H., Kebair F., Serin F. and Ben Said L..
Multi-Agent System Model for Container Management Simulation.
DOI: 10.5220/0004453404980505
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 498-505
ISBN: 978-989-8565-59-4

the container agents in storage areas introducing intelligence degree to reduce significantly the total number of unproductive movements.

The work presented in this paper enhances these researches by proposing an innovative model which integrates the multipoint of view and the emergence of behavior specific to the theory of complex systems. We propose to study the evolution of container agents behaviour from their arrival until their shipments. The final goal is to build a container management simulator that enables decision makers to analyse different port policies and to define container management strategies. We opt for an individual-centered approach to focus the management on the container life cycle, and to let emerge consequently a global behavior. Each agent of the system is a microscopic representation. We consider that the container is an individualized agent provided with an active behaviour. It seeks its satisfaction by moving optimally in the terminal to reach their destinations. The engines are also considered agents to ensure the mobility of containers and respond to their requests. To model the system architecture and the role of different parts which compose it, we use the agent-oriented methodology MESSAGE / UML (Caire et al., 2001).

This paper is structured as follows: first, container management in hub port logistics and their related problems are discussed. Then, the multi-agent modeling using the agent-oriented methodology MESSAGE / UML as well as the internal structure of the agents are presented. In the next section, an implementation of the model using the NetLogo simulation platform and the different results are detailed. Finally, we provide a general conclusion which includes the contributions of this work and the prospects that enrich it.

2 MANAGEMENT OF CONTAINERS IN A HUB PORT LOGISTICS

Mocellin in (Mocellin, 2006) defines a hub as a "*place where we receive the goods for the resend in a very short time.*" Unlike a warehouse, Mocelin (Mocellin, 2006) considers that it is infrequent to apply repackaging operations on a platform as it aims to tranship goods from one mode to another and redirect them to another destination. From the introduction of containers in the 50s, the international network has given rise to a multitude of modes, as the rail, road and sea mode. Thus, we can design transport systems in two conceptual ways including more than one transport mode: The multimodal transport which is the goods transport using at least two modes. The intermodal transport which is, according to The United Nations Economic Commission for Europe (ECE), the delivery of goods in one and the same loading unit (such as a container) or one and the same vehicle using successively several transport modes without loading or unloading (CEMT, 2006).

2.1 Delivery of Containers via Intermodal Network

Intermodal transport containers is maintained by a set of operations. In fact, the displacement of containers typically begins with the road mode from the freight terminal to a multimodal transport terminal where containers are transferred then from road to rail. Containers are transported for a long distance to the arrival terminal where they are transshipped onto a carrier. Containers are transferred by road to the port of the exporting country. After customs clearance operations and temporary storage, the containers are loaded onto container ships and then transported to the arrival port. Upon their arrival, the containers are unloaded from the container ships and transported by road in the importing country. The same procedure of transport is repeated from the arrival terminal until the arrival at the final customer (E.C.M., 2005).

2.2 Container Life Cycle Management

The life cycle of a container includes a set of processes that operate together. Their rapid chaining improves the delivery of goods in an appropriate time which progressively improves productivity. In this paper, we consider the mode ship to ship for delivery and receipt of containers. The four main processes are receipt, transfer, storage and delivery containers.

When a ship arrives, it must be assigned to an optimal position along the quay which is equipped with a number of cranes for the tasks of loading and unloading containers. This number varies depending on the number of containers to be treated and the length of the ship. Containers are transferred from a berthing area to a storage area near to the place where they will be transshipped next using a set of trailers or straddle carriers. After their transfer, containers are raised by stacking cranes or straddle and piled one on top of other. Operations of stacking provide the performance of a terminal by the segregation of containers in various strategies. To identify a container, a planner uses a computerized management system based on four coordinates which are the number of storage area, the bay number, the line number and the level of stacking. Additional moves are performed between empty stock, sheds, and import and export container stocks if sheds and/or empty depots exist within a terminal (Steenken et al., 2004). According to the request of a customer, the container will be delivered out of the terminal respecting the desired delivery date. A container is removed from the storage area, transported then by a transport vehicle to the ship operation area and finally loaded on a ship. The ship leaves the terminal after loading containers.

2.3 Types of Decisions in Container Management

Researchers use a set of indicators to evolve performance decisions. They contributed by the invention of several strategies for improving various tasks in the terminal. There are two types of decisions: planning and control. A planning decision is more concerned for the design and development process to be carried out in the implementation effective containers management. A control decision aims to ensure a high level of productivity and to monitor the process. Decision problems are divided into three levels: strategic, tactical and operational. Strategic decisions are long-term decisions that include the structure of the terminal, handling procedures and resource types to be chosen. Tactical decisions are medium term decisions which include the resources number to allocate. Operational decisions are short term which include the process to be followed by resources (Henesey, 2006).

2.4 Management and Decision Problems of Container Management

Given the large number of treated movements daily in a terminal, the decision is made by a set of constraints of time or space to plan and control the containers handling. We include the emergence of new missions during the day although planning is already established. The departure and arrival unpredictable of containers which complicates their location and distribution in the storage areas. This affects the strategic decision-making for planning storage. The unexpected appearance of breakdowns affects the allocation of resources to do the tasks which generates a rectification on the tactical decision to plan and control the containers handling. This leads an increase in the waiting time in the supply chain. The limited resources availability, which complicates the synchronization of tasks sequence. This leads to a problem of making operational planning. Due to the imperfection of classic tools to solve and to schedule the dynamicity and the complicated managing of containers in a terminal, we have used a distributed resolution using agent-oriented methodology.

3 MULTIAGENT MODEL FOR SIMULATING CONTAINER MANAGEMENT

3.1 Environment Characterization

We consider several assumptions in the problem resolution. Indeed, An elapsed time for the container treatment is measured to determine if the service is provided in a determined time. Moreover, ship departure date should be known in advance. The first agent resource that responds to the container agent request is the most appropriate. The containers number to be processed simultaneously is obtained by the quay cranes number allocated per vessel. The container satisfaction is to perform the unloading, the transfer and the loading operation as soon as possible. The containers priority depends on several factors.

- The containers number to be processed by quay cranes is obtained by dividing the containers total number (to be loaded and unloaded by vessel) by quay cranes number.
- The service time per container is obtained by dividing the service time by the containers number to be processed per quay cranes. It corresponds to the deadline for processing container.
- The containers priority depends on the ship service time whereby it arrived ts_1 and the ship service time whereby will be shipped ts_2.
- Therefore, the containers priority is defined by the shortest sum of service time per each container.

$$p = ts_1 + ts_2$$

We also identified several parameters in our models which are:

- c: means the resource capacity. It is measured by the containers number handled in the same process;
- p: is the container priority;
- d: it the distance between two points and is measured by a time interval;
- t: the service time is the time interval set between arrival date and departure date of a ship. It corresponds implicitly to the limit time of the processing of all containers;

- *tc*: the current processing time of a container that is initiated to zero and evolves in time until the end of treatment.
- *ts*: the average service time per container is the period fixed for processing each container.

3.2 System Design and Architecture

MESSAGE / UML is a methodology for agent-oriented software engineering for describing and modeling agents. It allows to study the dynamic evolution of the behaviour of each agent by the incremental development of links between the agent and its properties (goals, services, roles, resources, etc.). We use this methodology in order to define and to characterize the multiagent model that we propose. Indeed, figure 1 shows the structural relationships of the organization model. An *Organization* is represented by an isosceles triangle. The *Role* that represents an agent is schematically drawn by ellipses. The *Class* formulates objects that are used in the port hub. In fact, the container terminal is the main organization in which run the port operations. It is composed of a set of containers with different types and dimensions. Containers to treat, require services from a resource organization comprising handling equipment and transport vehicles. A machine is a resource agent. Storage areas and communication lines are treated as objects to be consumed in the terminal. It should be noted that the appropriate icons have been associated with different stereotypes.

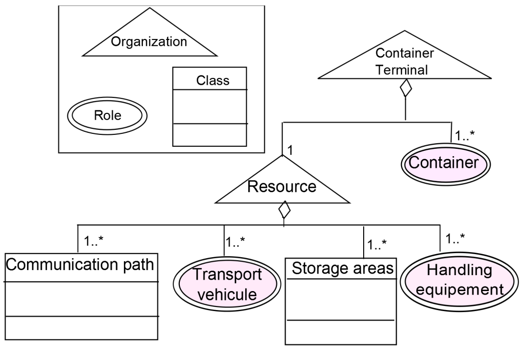

Figure 1: Structural relationships.

The inter-agent interaction is an important concept in multiagent systems and the establishment of social organizations. According to (Ferber, 1995), it is expressed by a set of actions that have an influence on the future behavior of agents. Interaction can provide the possibility of communication between agents in order to coordinate and to avoid the encounter that leads to a collision (such as moving vehicles causing a collision).

Figure 2 shows the acquaintances relationship between the system entities where different stereotypes are schematized by the same icons of figure 1. An *acquaintance* is shown by a double arrows. A container agent interacts with resource agent to determine the set of tasks of a requested service in two cases. It communicates with a resource agent to establish the process of resource sharing using the *Contract Net protocol* or to determine the allocation resource through the planner agent using the *request protocol*. According to (Ferber, 1995), the type of interaction between the resource agent and the container agent is *"Collaboration coordinated* ". It is characterized by a compatible goals between entities (answer to the containers satisfaction) and insufficient skills. Agents must coordinate their actions in order to ensure the realization of all requested services by a container agent.

Agents resources communicate with each other using the *request protocol* to ensure the consistency and the synchronization of tasks to accomplish.

Container agents interact with each other using the *Contract Net protocol* to negotiate the resources sharing on condition that they have the same destination and the same size. The interaction between container agents is defined by goals incompatibility which requires negotiation to achieve their satisfaction. So, we can name the interaction type by *"Pure individual competition*" (Ferber, 1995).

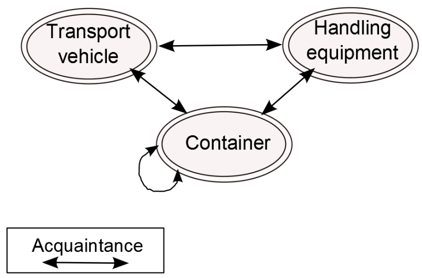

Figure 2: Acquaintance relationships.

In MESSAGE methodology, each agent owns tasks, interactions and goals in its environment. A task is executed when a precondition is valid and when is completed by a post-condition. A composite task is expressed in terms of causality between subtasks. It can be modeled by a state machine. We modelise it therefore by a state machine using the ATN (Augmented Transition Network) (Woods, 1970), we discuss this in the next section.

3.3 Internal Structure of Agents

ATN is the main internal element of an agent. ATN is a finite state machine that describes the agent behaviour where each transition is marked by one or more conditions and corresponding actions. Its role is to reflect the dynamicity and to insure the agent reactivity and proactivity. According to (Cardon, 2003), an agent is structured by the following modules:

- Knowledge module: contains the agent knowledge which may be static or dynamic. Static knowledge represent environmental data whereas dynamic knowledge represent environmental variables that are used by agents to negotiate with other system entities.
- Communication module which is composed mainly of "acquaintances Network" and that enables agent to interact with other agents.
- Behaviour module, which is composed of an ATN, manages the agent behavior based on his knowledge.

3.3.1 Container Agent

The container is the principal agent of the system. It changes state while executing a set of actions to achieve its internal goal. An ATN is used to model its behaviour and which comprises an initial state, a final state and a finite set of states as shown in figure 3.

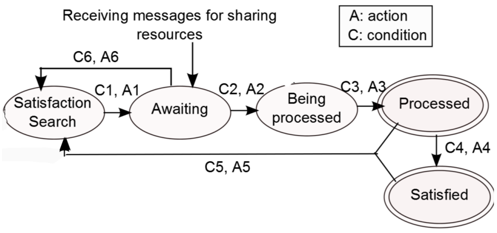

Figure 3: ATN of a container agent.

Satisfaction Search State. When a container agent needs a resource to be treated (C1), it starts negotiation with other agents having a resource. It broadcasts therefore a call for proposals indicating its destination, its dimension and its priority (A1) and goes to state "*Awaiting state*". In return, the assigned container agent, that receives the message, checks whether the resource sharing will affect its remaining processing time. This process is performed using the following equation to select calls. We note that ts_A is the service time per affected container and ts_{NA} is the service time per non affected container.

$$Z = ts_A - (ts_{NA} + d_{A,NA})$$

There are two cases:

- If $Z < 0$, ie if the remaining processing time of unaffected container adding the elapsed time of the distance between the affected and unaffected container agent exceeds the remaining processing time of affected container , then it rejected the call.
- If $Z \geq 0$, then the affected container agent asks the resource agent for sharing. The latter accepts or rejects the request indicating its capacity. Next, the container agent chooses among the selected calls the highest priority calls and sends proposals proportionally to the capacity of agent resource.

Awaiting State. It waits proposals (C2), then collects them and finally accepts the most appropriate one (A2). In return, the affected container agent sends a confirmation to resource agent indicating the containers number that have accepted the proposal of sharing resource and their identifiers. Then, the resource agent broadcasts an information indicating that it is ready to container agents sharers the resource agent (C2).

On failure (it does not find proposals for its request (C6) or it refuses all proposals that do not possess an adequate destination (A6)), it returns therefore to "*Satisfaction Search state*" (A6). In this state, it proceeds the coordination with resource agents through a planner agent by sending a message to the scheduler indicating its destination, the operation type and the remaining processing time and switches after that to "*Awating state*" (A1). When it receives an information from planner indicating its assignment to a resource (C2), it sends a confirmation message to the resource agent (A2). It receives an information from resource agent indicating that it is ready (C2) and it transits to "*being processed* ".

Being Processed, Processed and Satisfied States. In this state the container agent is being processed. As soon as the treatment is finished (C3), it informs the scheduler about the end of treatment (A3). Finally, the container passes from "*Processed state*" to "*Satisfied state*" that presents the final state in which the container will be processed before the deadline to ensure their satisfaction (A4) and in which it tries to stay. If it is satisfied (C5), it returns to the initial state (A5).

Knowledge module: consists of acquaintances of container agent are composed of:

- Static knowledge: container id, dimension and nature.
- Dynamic knowledge: Destination, service time, number of containers to be treated, current processing time and operation type.

3.3.2 Resource Agent

The resource agent, which is responsible to provide services is of two types: handling equipment and transport vehicle. A container treatment is a coherent sequence of a set of tasks. Each state carries roles and brings a specific version of each service in different processes. The ATN of the resource agent has two states which are (figure 4):

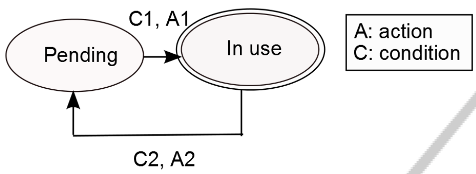

Figure 4: ATN of resource agent.

Pending State: presents the initial state in which the resource agent is still waiting container requests to ensure its satisfaction (C1). When it receives a request from container agent, it responds to requests according to its capacity, the container dimensions and the type of authorized terrain. In case resource agent accepts the request (A1), it receives confirmation to start treatment (C1) and goes to state "*In use*".
In Use State: presents the final state in which a resource agent moves to the location of the container and, sends information indicating that it is ready to the container (A1) and then begins processing. Finally, when it finishes its processing (C2), it returns to the initial state "*Pending*" and informs planner about the end of treatment (A2).

The knowledge module of a resource agent is composed of:

- Static knowledge: which are the resource Id, the resource type, the speed and the capacity.
- Dynamic knowledge: which contains the container coordinates, the type of operation and the location of the ship.

4 RESULTS

Figure 5 shows the entities of the simulation environment. Entities are composed of straddle carriers that represent the resource agent, containers that represent the container agent, ship, communication routes and junctions. Entities are coded in color to help understanding the evolution of the state entities as well as the system state.

To treat the coordination among agents, we have established a dialogue between the container agent and the resource agent using the formalism FIPA / ACL. We chose to simulate the loading operation to validate the model and to understand the different policies in the containers management by altering some parameters in the purpose of taking and evaluating different decisions. The instance is repeated in each hour to control the evolution of container behavior as well as the rate of its satisfaction during the simulation by changing the speed of resource agent and the number of containers to be processed. The rate of satisfaction (RoS) is obtained by dividing the number of satisfied requests per the number of required requests. We note that N is the number of required requests and Ns is the number of satisfied requests.

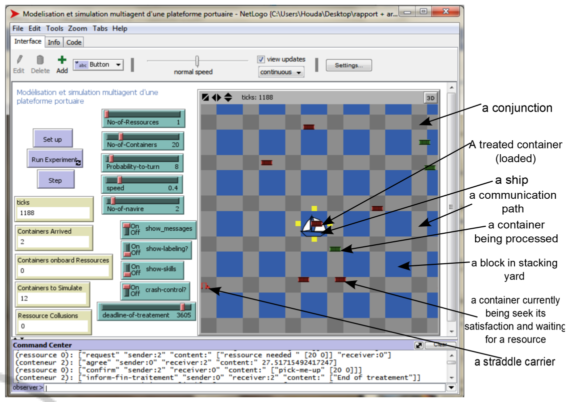

Figure 5: Simulated container management Layout.

$$RoS = \frac{Ns}{N} * 100$$

Moreover, given the dynamicity of the container terminal, we investigate the perturbations that can be made during the treatment of containers. We evaluated , for this purpose in two scenarios, the number of collisions depending on the speed of resource agent. According to the results in table 1, scenario 1 and 2 has successively a satisfaction rate of 60% and 36%. We note, basing-on scenario 1 and 2 that the number of containers to be treated has a great impact on the rate of containers satisfaction. In other hand, according to a study done by ECMT on 34 terminals, the number of containers to be treated has a big influence on the cranes productivity and on the quay capacity (Chambreuil, 2011). So we can interpret that the greater a number of containers to be processed per ship is, the more a number of unsatisfied containers increases and the more the waiting time increases. In practice, a quay is considered saturated when it is used more than 60%. So we have a waiting time at berth (queuing theory). Also, the number of allocated cranes depends on the size of container ship and the number of container to be treated, so the number of allocated cranes is proportional to the number of con-

Table 1: Quality of the solution for the instance # 1.

Sc	N	speed	NoC	RoS(%)
Sc 1	20	0.3	0	60
Sc 2	30	0.3	0	36
Sc 3	20	0.6	0	70
Sc 4	30	0.6	0	53
Sc 5	30	1	61	46
Sc 6	30	1.2	73	17

N: number of required requests
NoC: number of collusions
RoS: rate of satisfaction

tainer to be treated. Moreover, according to the hypotheses, priority of container p depends on the number of containers and the number of cranes. Thus, we can take a decision concerning the setting of the number of cranes with the aim to increase the productivity of cranes and consequently to improve the rate of satisfaction. We take Scenario 2 and 4 which integrate two different speeds, Scenario 4 possesses a higher satisfaction rate comparing with scenario 2. We were able to validate that the vehicle speed is an important factor in planning. It varies according to the vehicle type and the terrain type where it moves. This explains that the choice of equipment plays an important role in strategic decision making for an efficient management of containers. The work of Henesey (Henesey et al.,) implement the different policies for sequencing, berthing, and stacking on the performance of CTs includes additional variables such as the number of equipment used in a terminal and the allocated road by the transport vehicle. Finally, if we take scenario 5 and 6, we note that more the speed increases, greater the number of collusions increases in the terminal. On the other hand, although the speed is higher, the rate of satisfaction is lower due to the disturbances that can happen during treatment of containers.

5 CONCLUSIONS

In this paper, we modeled a multiagent model to simulate container management. The model aims to deliver goods to customers in time to satisfy them. It involves only container agents and resource agents that cooperate and negotiate with each other to distribute tasks among resources and to organize their achievements over time according to containers priority. We choose the ATN to describe the internal structure of agents and MESSAGE method to describe the system architecture. We implemented a first prototype and we extracted a first results in order to test and to validate the proposed approach. Currently, we proceed to extend our model using cellular automata for modeling the location of the containers in a container terminal.

REFERENCES

Bin, S., Junqing, S., and Qiushuang, C. (2011). On container terminal logistics operational and scheduling management system based on mas. In *Control Conference (CCC), 2011 30th Chinese*, pages 5484 –5489.

Caire, G., Garijo, F., Gomez, J., Pavon, J., and Vargas, E. (2001). Agent oriented analysis using message/uml. pages 119–135. Springer-Verlag.

Cardon, A. (2003). *Modéliser et concevoir une machine pensante : approche constructible de la conscience artificielle*. Automates intelligents. Automates intelligents, Paris. Lexique des définitions.

CEMT (2006). Terminologie en transports combinés (francais, anglais, allemand, russe).

Chambreuil, A. (2011). Productivité des terminaux à conteneurs. Le CETMEF.

E.C.M., T. (2005). *La sûreté du transport intermodal de conteneurs*. OECD Publishing.

Ferber, J. (1995). *Les Systèmes Multi Agents: vers une intelligence collective*. Paris, France, interedditions edition.

Govil, M. and Proth, J. (2001). *Supply Chain Design and Management: Strategic and Tactical Perspectives*. Academic Press Series in Engineering. Academic.

Henesey, L. (2006). *Multi-agent systems for container terminal management*. Blekinge Institute of Technology doctoral dissertation series. Blekinge Institute of Technology.

Henesey, L., Davidsson, P., and Persson, J. Evaluating container terminal transhipment operational policies: an agent-based simulation approach. *WSEAS Transactions on Computers*, 2006:2090–2098.

Henesey, L. E., Notteboom, T. E., and Davidsson, P. (2003). Agent-based simulation of stakeholders relations: An approach to sustainable port and terminal management. In *International Association of Maritime Economists Annual Conference, 2003(a*, pages 314–331.

Kefi, M., Ghedira, K., Korbaa, O., and Yim, P. (2009). Container handling using multi-agent architecture. *Int. J. Intell. Inf. Database Syst.*, 3(3):338–360.

Mocellin, F. (2006). *Gestion des entrepôts et plates-formes: Assurez la performance de votre supply chain par la maîtrise des zones de stockage*. Fonctions de l'entreprise. Série Performance industrielle. Dunod.

Steenken, D., VoB, S., and Stahlbock, R. (2004). Container terminal operation and operations research, a classification and literature review. *OR Spectrum*, 26:3–49.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Commun. ACM*, 13(10):591–606.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons.