



## Container Safety Storage on a Port Terminal

Mansoriya Hamidou, Dominique Fournier, Eric Sanlaville, Frédéric Serin

### ► To cite this version:

Mansoriya Hamidou, Dominique Fournier, Eric Sanlaville, Frédéric Serin. Container Safety Storage on a Port Terminal. 4th international conference on logistics, systems and supply chain, Aug 2012, Quebec, Canada. hal-00989920

**HAL Id: hal-00989920**

**<https://hal.science/hal-00989920>**

Submitted on 12 May 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Container Safety Storage on a Port Terminal

M. Hamidou<sup>1,\*</sup>, D. Fournier<sup>1</sup>, E. Sanlaville<sup>1</sup>, F. Serin<sup>1</sup>

<sup>1</sup> LITIS Laboratory, University of Le Havre – France ({mansoriya.hamidou, dominique.fournier, eric.sanlaville, frederic.serin}@litislab.fr)

---

**Abstract :** *A container terminal is a complex and dynamic system. Many operations occur within the storage area: containers import, containers export and containers shifting. All these operations require the respect of many rules and even laws in order to guarantee the port safety and to prevent risks, especially for the hazardous material storage. This paper proposes an hybrid architecture, using a Multi-Agent System and a Cellular Automaton, to handle the hazardous container storage problem. It is an optimization problem since the aim is to improve the container terminal configuration, that is, the way hazardous containers are dispatched through the terminal. Simple optimization heuristic methods were tested on a terminal with four hazardous container types. We consider then containers as agents, in order to use a Multi-Agent System for the decision aid software, and a Cellular Automaton for modelling the terminal itself. This approach will improve the naive methods first implemented, and should apply to the actual data and constraints of container terminal management.*

**Keywords :** *Container terminal, cellular automata, multi-agents system, optimization problem, dangerous containers*

---

## 1 Introduction

This paper proposes a dynamic technique to manage the storage of dangerous goods in a container terminal. This work aims at maintaining the safety of a terminal during all the handling operations that can be executed in such areas.

More precisely, our research is about stacking activities and dangerous containers storage in a port terminal. The problem is: how to position hazardous containers in compliance with physical constraints and regulations? The International Maritime Dangerous Goods (IMDG) Code, available on International Maritime Organization (2011) site web, classifies dangerous goods on 9 main classes (table 1). Their stockpiling must respect regulation and separation rules for each class. Our aim is to maintain a safe configuration of the terminal. The management of handling equipments is outside the scope of this paper. Methods for the scheduling of Straddle Carrier (SC) missions, and the subsequent routing, are investigated in other papers, see Lesauvage et al. (2011); Balev et al. (2009).

In the following, we first present more precisely the context of our problem and some related works. Then we detail the

optimization model; the limits of a classical optimization approach (especially through integer linear programming) are presented. Indeed, the size of the linear problem is too large for exact solving, and there are in fact several performance criteria. Furthermore, many additional constraints appear that cannot easily be modelled by such techniques: in real terminal, there are several uncertainty sources, as the possibility of damaged containers, of an already occupied place unforeseen by a centralized system,... and the system is structurally dynamical, with the constant arrival and departure of containers to and from the terminal.

To cope with these difficulties, heuristics based on local decision rules are first proposed. The drawbacks of this approach are illustrated on some examples. We propose in the last part to treat the problem with the help of a model driven approach mixing different tools issued from artificial intelligence: cellular automata, object-oriented design, agent paradigms.

The positioning strategies are limited by the human mind when a traditional and formal approach is adopted. By using a Multi-Agent System, we expect the emergence of original behaviours due to the dynamic created by interactions, negotiations and collaborations among different agents having their own constraints and goals.

---

\*PhD Student. Corresponding Author. Address: Rue Philippe Lebon, BP 540, 76058 Le Havre Cedex, France. E-mail address: mansoriya.hamidou@litislab.fr

## 2 Problem description and related works

### 2.1 Context

Containers are relatively uniform boxes, which can be loaded, unloaded, stacked or transported over long distance. They have been designed for fast and easy handling of freight. Each container has a set of properties like dimensions, weight, destination and type of goods it contains.

A container terminal is a part of a port where containers are stored and handled. The storage area (yard) is divided in blocks. On each rectangular block containers are arranged in rows and slots (piles of at most 4 containers high). Space between two rows allow the handling equipment circulation.

Handling equipments are required for terminal management. They transfer containers within terminal and transship them. Common equipments are chassis based-transporter, straddle carriers SC, quay crane, rubber tired gantry crane and rail mounted gantry crane (Stahlbock and Voß, 2008).

In a terminal, there are three main activities concerning containers:

**Unloading** containers are discharged from a ship or other transport mode like trucks or train, to be transferred to the storage area using handling equipments.

**Staking** containers are stored on the area dedicated to them, respecting physical constraints and regulations.

**Loading** container leave storage area and are loaded to be transported on train or ship.

This paper focus on the stacking activities, and the storage area containers are moved by Straddle Carriers. When a container is moved from one place to another, within the storage area, we talk about a "transfer", and a "shift" is a set of grouped transfers, for example, moving a container and the container above it.

### 2.2 Dangerous goods

Containers are boxes which contain goods. These goods can be dangerous and are then called hazardous materials or dangerous goods. This means articles or materials potentially dangerous for people or environment. It includes items of common use, such as aerosol cans, perfumes, and paints (BusinessDictionary.com, 2011).

The nine IMDG classes of dangerous goods are listed in table 1. Some of these classes are subdivided on divisions or subclasses. There exists a total of 20 classes or subclasses.

Class #	Dangerous Goods
1	Explosives
2	Gases
3	Flammable liquids
4	Flammable solids; substances liable to spontaneous combustion; substances which, in contact with water, emit flammable gases
5	Oxidizing substances and organic peroxides
6	Toxic and infectious substances
7	Radioactive material
8	Corrosive substances
9	Miscellaneous dangerous substances and articles

Table 1: IMDG Classes

Storage constraints exist for each class. The containers of some class cannot be stored next to another, or must be separated from them by a fixed distance. An example of separation rules is cited below, see Institut du Droit International des Transports (2011).

A **flammable liquids** containers (class 3) must be separated by:

- Distance F from explosives (class 1). F equals to

$$F = 4.8 \times \frac{Q}{3}$$

where:

$F$  is a separation distance in meters; and  
 $Q$  is the explosive net weight in kilograms.

- 30 meters from gases (class 2).
- 7 meters from radioactive.
- etc.

### 2.3 Related works

As far as we know, there is no work dealing with the storage of containers with dangerous goods in a terminal. However, many research papers use agent-based approach to simulate or solve transport logistics problems (Davidsson et al., 2005). Some of them study the container terminal management problem using Multi-Agent System and their aims focus on various aspects of terminal planning and management Rebollo et al. (2001); Henesey et al. (2003); Thurston and Hu (2002).

In Kefi et al. (2007) a MAS approach was used for storing containers respecting their departure time. The authors use two kinds of agents (Container Agents and Interface Agent) in order to optimize the container storage area on a port terminal, their goal was to reduce the transportation cost within the terminal.

All works cited reinforce our idea to use a MAS approach to model the management of a port terminal and to solve our problem. Moreover, Kefi et al. (2007) used such an architecture to perform container storage optimization which has a spatial aspect like our problem.

Other papers used Operational Research techniques to solve container storage problem in a terminal. Kim and Hong (2006) proposed two methods for determining the relocation of containers: a branch-and-bound algorithm and a decision rule, but it was limited to only 6 stacks by 5 containers high (5 tiers).

In Kim and Lee (2006), constraint satisfaction technique was used for space allocation to export containers. The objective was the maximization of the equipments efficiency.

More recently, Salido et al. (2010) resolved both allocation berth problem and container stacking problem by a set of Artificial Intelligent based heuristics. In the container stacking problem, the objective was the minimization of number of relocation. In this paper, dangerous containers were considered but the constraint was: two dangerous containers must maintain a minimum security distance, but different existing classes and rules of dangerous containers were not considered.

The spatial aspect also appears in works on cellular automata (Wolfram, 2002). Cellular automata are in particular used by geographers and economists to model the evolution of a population inside a given space. (Schelling, 1978) was first to study the racial segregation mechanism in a urban area by CA, and he showed it could be very accurately simulated using some cellular automaton with very simple rules. A cell of the automaton is an accommodation (flat or house). Its state is the group of its inhabitant, if any. The inhabitant decides to leave if the percentage of foreigners (relatively to his group) in its neighbourhood exceeds a given threshold. He then moves to any free accommodation. Under very weak initial conditions and a high tolerant threshold, a segregation appears between the different groups of inhabitants. It is however difficult to build the transition function of such an automaton as the arrival of an individual in a given empty cell cannot be entirely predicted. As we shall see later, we use a similar model, where inhabitants are replaced by containers.

### 3 Problem modelling

$N$  Containers of different types  $T_1, T_2, \dots, T_r$  are packed together on one terminal. For simplicity, it shall be supposed in that presentation that the terminal is composed of one unique block of  $n$  rows. According to its type and the typology presented before, the *well-being* of one container can be evaluated. For instance, considering a dangerous container of radioactive type as in previous section, its well-being depends on the number of containers of any dan-

gerous type present in its neighbourhood (defined in terms of euclidean distance). Generalizing this observation, it is easy to derive a *well-being value* for each container of the terminal, which can be normalized according to all container types. The *total well-being value* of a whole terminal configuration can be computed as the worse of the well-being values of all containers it contains (an alternative criterion is the sum of well-beings). This is also called fitness function in section 4.

Consider now some initial terminal configuration, associated with its well-being value. The problem consists in changing the configuration through a sequence of transfers (moves of a container from one place to another) so as to optimize the total well-being. This optimization problem is not simple to solve because of the different types of containers, and the spacial dimension of the problem. It is also clear that the optimal configuration does not depend on the initial configuration, but only on the number of containers of each type. Finding this optimal configuration is a problem of placing objects in a three-dimensional environment, so as to allot each type at best.

#### 3.1 Example of linear model

For the simplest variants, there might be an analytical solution. For other simple variants, the optimization problem may be modelled by an integer linear program. To illustrate this, let us consider the following decision problem: suppose there are only two types  $T_1$  and  $T_2$ , and the well-being of a container is the number of containers of the other type close to him (Moore neighborhood, see later). Given a block size and a fixed number of containers of both types, does a configuration exist such that each container has a maximum well-being (that is, no container of the other type in its neighborhood)? An integer model may be built, with for each place  $k$  of the block one binary variable  $x_k$  that states whether the place  $k$  contains a container, and binary variables  $y_{kt}$ , one by dangerous type;  $y_{kt} = 1$  if and only if the container present at place  $k$  is of type  $t$ . There is for each place and for each dangerous type at least one constraint to ensure that the container neighborhood is indeed free of undesirable containers. This constraint may be written for type  $T_1$ :

$$\forall \text{ place } k, \frac{1}{N_k} \cdot \sum_{k' \in \mathcal{N}(k)} y_{k'T_2} + y_{kT_1} \leq 1$$

where  $\mathcal{N}(k)$  is the set of places in the neighborhood of  $k$ , and  $N_k$  is the cardinality of that set (at most 26). There are also additional constraints to ensure feasibility (for instance a container occupies a place only if it is on the ground or just above a place containing itself a container). Considering a typical block in European ports will have around 10 rows, 20 slots per row and be 4 container high; our integer program has around 2400 binary variables, 1600 constraints of the kind above, plus additional constraints due to

the structure of the block. From these observations made on a simplified version of the problem, it follows that an integer programming optimization approach will be intractable.

### 3.2 More complications

However, we are not really interested in solving exactly the above problem. Indeed, at least four other points need to be investigated.

1. The number of transfers to achieve the desired configuration cannot be too large, as any move of a straddle carrier for instance has a non negligible cost. In fact, the problem is *bi-criteria*: the goal is to achieve the best configuration, but through a minimum number of transfers.
2. The life of the port does not stop while the configuration of the terminal is undergoing some changes: new containers are added to the terminal stock; meanwhile, some containers are removed, picked up by lorries, trains, maritime or river ships. Hence the problem is dynamic.
3. Computing a neighborhood is more complicated, as the security distance is not just euclidian but also takes into account obstacles like other containers. Thus the neighborhood itself is dynamic.
4. Some uncertainties are present : some containers may be at another place than they are supposed to be, some may be damaged and need immediate care,...

Hence our basic optimization problem is in fact: bi-criteria, dynamic, and subject to uncertainties. From all these considerations, it seems that the problem can hardly be tackled by the classical tools of static optimization. What is needed is a completely different model, a decision aid tool which should be reactive to expected and unexpected changes, consider both the number of transfers and the global well-being of the terminal. Finally, it is also desirable that it propose decisions with incomplete informations.

## 4 First solving approach and preliminary results

Here we propose first simple approaches, based on an heuristic optimization method.

To illustrate solutions, we consider a toy block composed of 2 rows. Each row is 5 containers long and at most two containers high, as shown in figure 1. We used only two dangerous container types, with a simple separation rules, and one neutral container type. We mean by neutral, all containers that are not classified as dangerous.

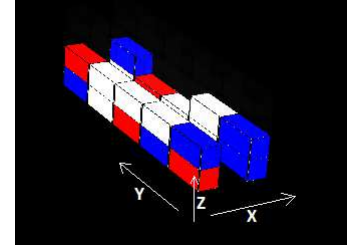


Figure 1: Toy terminal

The method purpose is to rearrange containers within the block in order to optimize the block fitness function. We consider only containers shifts inside the block and suppose there is no container entering or leaving. Dangerous containers are coloured with red and blue, while neutral are white boxes.

- *Red boxes*: are containerized combustible material.
- *Blue boxes*: are containers transported fuel.

Separation rules for each type are:

- Red containers must not have another red containers in the neighborhood.
- Blue containers must not have red containers in the neighborhood.

Let us consider the simple 3D Moore neighborhood. It is composed of the twenty six cells surrounding a central cell on a three-dimensional grid, like shown in figure 2:

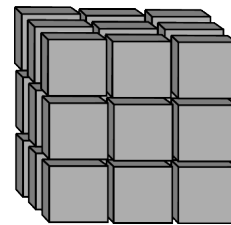


Figure 2: Moore 3D neighborhood

A fitness (well-being) is associated to each container. It depends on the separation rules of container type:

- The fitness of some red container equals the number of blue containers in its 3D Moore neighborhood.
- The fitness of some blue container equals the number of red containers in its 3D Moore neighborhood.

The aim is to ameliorate terminal configuration by decreasing the fitness function of the block. Its value is the maximum of all containers fitnesses.

Two resolution methods were tested. In both of them, the container with the maximum fitness value is moved from its cell, but the moving strategies differs from each other.

For example, for the terminal (block) illustrated in figure 1, at the beginning, the fitness of each container is calculated. The container with the higher fitness value, is the container A  $C_A$  with coordinates  $C_A(0, 0, 0)$ . It has 4 blue containers in neighborhood, so its fitness equals to  $Fit(C_A) = 4$ , so the terminal fitness function equals  $Fit_T = 4$ .

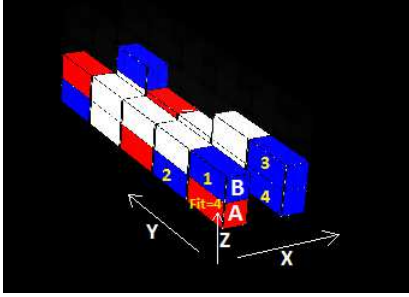


Figure 3: Container terminal at  $t$

Fitness of container  $C_A$  can be optimized by a shift (remember it is a set of transfers), in order to optimize terminal configuration. The two optimization strategies tested are explained below.

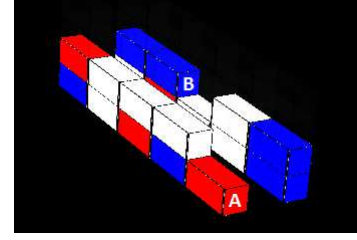
#### 4.1 First strategy

The strategy of the first method consists in:

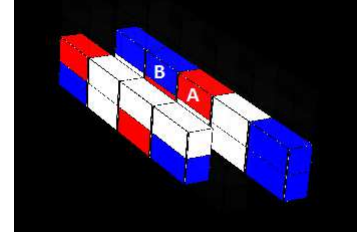
First, find a new place for the container chosen to be moved, which decreases its fitness, and reserve this place. Second, if other containers are above the chosen container, find the best places for them, among free places in terminal excluding the reserved place. Next, move containers one by one, beginning by the stack top (the chosen container must be moved to its reserved place). Finally, recalculate the fitness of each container and the global fitness function of the terminal.

Let us apply this strategy to our example (figure 3). First we search a place for the red container  $C_A$ . If it is placed on  $Cell(1, 2, 1)$  then its fitness function will be equal to 1, because it will have one blue container in neighbourhood. And, if it is placed on  $Cell(1, 3, 1)$  then its fitness will be equal to 2, because it will have two blue containers in neighbourhood. So the reserved place is  $Cell(2, 1, 1)$ . Then next, we move the container above  $C_B(0, 0, 1)$ . It cannot be moved to  $Cell(1, 2, 1)$  because it is reserved, so, we move it to  $Cell(1, 3, 1)$ .

When transfers are carry out, fitness functions are calculated.  $Fit(C)$  is the local fitness function value of container  $C$  and  $Fit_T$  is the global fitness value of terminal (which is the maximum local fitness value). In our case,  $Fit(C_A)$  decreased but  $Fit(C_B)$  increased (from 1 to 4), so  $Fit_T = 4$



(a) Container terminal at  $t + 1$



(b) Container terminal at  $t + 2$

Figure 4: A shift within storage area - Strategy 1

because the maximum fitness value is  $Fit(C_B) = 4$ , and it does not decrease.

	Before shift (at $t$ )	After shift (at $t + 2$ )
$Fit(C_A)$	4	2
$Fit(C_B)$	1	4
$Fit_T$	4	4

Table 2: Fitness value evolution - Strategy 1

#### 4.2 Second strategy

The second strategy consists in:

If the container chosen to be moved is not on the stack top, then, first, find places which decreases fitness value, for each container above, beginning by the stack top, until the chosen container. Next, calculating the future global fitness function, if it decreases then carry out transfer containers, one by one.

In our example, (figure 3), The method begins by finding the best place for blue container  $C_B$ , because it is the stack top. If we move it to  $Cell(1, 2, 1)$  its fitness will be equal to 2 and if we move it to  $Cell(1, 3, 1)$  its fitness value will equal 3, so the best place for it, among the free places on terminal, is  $Cell(1, 2, 1)$ . For the container  $C_A$  it will be placed on  $Cell(1, 3, 1)$ . Its fitness will equal 3.

After the shift was done, the fitness values are computed. As we see in table 3,  $Fit(C_A)$  decreased and  $Fit_T$  decreased but  $Fit(C_B)$  increased from 1 to 3.

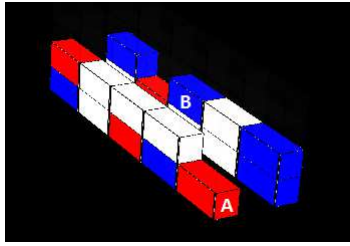
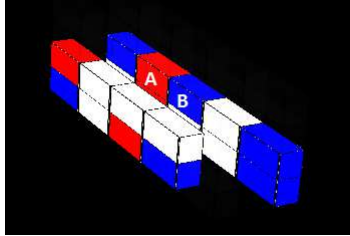

 (a) Container terminal at  $t + 1$ 

 (b) Container terminal at  $t + 2$ 

Figure 5: A shift within storage area - Strategy 2

	Before shift (at $t$ )	After shift (at $t + 2$ )
$Fit(C_A)$	4	3
$Fit(C_B)$	1	3
$Fit_T$	4	3

Table 3: Fitness value evolution - Strategy 2

### 4.3 Weaknesses of heuristic approaches

As we saw in the example, the heuristics approaches were not very efficient. In the first strategy, global fitness is tried to be decreased by decreasing the maximum local fitness value, but it does not always work. When the container with the higher fitness value is chosen to be moved and has other containers above, by transferring containers above, we can largely increase their local fitness value, if there is not other places better, consequently the global fitness value is increased. So the shift is carry out without improving the storage area configuration, and this shift has a cost on the terminal managing.

Another problem appears, the problem of cycles. An example of a cyclic movement: the method chooses to move container  $C_1$  placed on place  $P_1$  because it has the maximum fitness value, but container  $C_2$  placed in place  $P_2$  is above. First  $C_2$  is moved, even if a place with less fitness is not found. So it is moved to a place  $P_3$  witch increases its fitness function. If the best place for  $C_1$  on the terminal is the place  $P_4$ , which is above  $P_3$ , the container  $C_1$  is moved to  $P_4$ .

At the next iteration, the container with the maximum fitness function is the container  $C_2$ , and for moving it, container  $C_1$  must be moved. The place chosen is  $P_1$ . It's not the optimal place but there is not better. the transfer increases its fitness, and  $C_2$  is moved on  $P_2$  ... so the strategy leads to infinite cycles.

In addition to the issues we have encountered, it is difficult to apply these methods to nine dangerous container classes and complex separation rules. It is why integration of MAS approach will be necessary to solve the problem. In particular, negotiation between agents will avoid pitfalls like the cyclic problem above. Considering each container as an agent, interacting on a Cellular Automata environment, should allow fast, intelligent and efficient terminal management.

## 5 Hybrid model using cellular automaton and software agents

Our model is guided by three levels of design. The first layer is the object based representation of the physical situation. We reify the container terminal using an object model previously validated. The second layer uses the similarity between the block structure and the architecture of Cellular Automata. This second viewpoint merges the physical reality with a representation of information. The third and last layer is the adding of "intelligence" in our system. We introduce the agent approach considering containers as partially autonomous elements.

### 5.1 Object-oriented modelling

The object-oriented approach permits to distribute the properties and behaviours then to enhance progressively the model. So, the first step of our work consists of an object oriented modelling of a container terminal.

A block is composed by rows, and rows own slots (columns of 4 places). Each container can be stored into a place situated in a block or can be transported by a carrier. These ones move upon ways, pass by gates or interface items (quay cranes for example).

Some parameters in a block configuration are considered: the number of rows, the number of ground places by row, the height of a row that is to say the number of containers it is possible to stack at a given place. The height depends on the physical capability of a straddle carrier to lift a container; commonly this height is comprised between two and four containers.

In the model, we consider two super classes of objects: the statics and mobiles ones. The static objects are used to describe the terminal structure. We enumerate them as follow : blocks, rows, stacks, and places.

The mobile objects are containers and straddle carriers. In a first time, we only consider straddle carriers as constraints for stacking. Note that these classes are associated to the notion of place: a container must be stored into a place, a straddle carrier is a special place.

## 5.2 Cellular Automaton structure

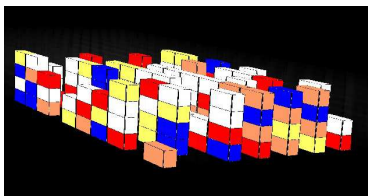
A container terminal is a set of three-dimension cubic cells arranged in rows. These properties inspire us to introduce, by similarity of structure, the notion of 3D Cellular Automaton (CA). This is the second layer of our design.

A Cellular Automaton is a complex and dynamic system. It is a collection of cells on a grid. Each cell has a "state" among a finite set of states, and evolves through a number of discrete time steps according to a set of rules based on the states of neighbouring cells. The grid can be in any finite number of dimensions (Wolfram, 2002). If state updates occurs synchronously, we speak about synchronous cellular automata, i.e the states of every cell in the model are updated together. In contrast, in an asynchronous cellular automaton cells are updated individually and independently, in such a way that the new state of a cell affects the calculation of states in neighbour cells.

Thus, each cell of our cellular automaton corresponds to a container place on terminal. It can be free or occupied. The neighbourhood of each cell depends on the container class and its separation rules, it is defined in terms of euclidean distance, but a transition function is not simple to be expressed. It will correspond to agents' decisions.



(a) Container terminal



(b) Cellular automaton

Figure 6: Container terminal structure and cellular automaton

At each transfer, the state of one cell, or a set of cells but not the totality, changes. So our CA is asynchronous.

People who studies CA, are interested by: How the system evolves? Does the system converge after a finite time? Can a set of configurations be repeated? in other words, can cycles be generated? These questions are typically the ones we wish to answer for the terminal management.

## 5.3 MAS approach

### 5.3.1 MAS model

A Multi-Agent System is a set of physical or virtual autonomous entities, located on an environment. They can coordinate, communicate, negotiate and interact with each other, using their resources and skills, in order to fulfil common and individual goals. Our project aims to avoid a coordination center and consequently to introduce local and neighbourhood consideration to proceed the placing of hazardous items.

As dynamic and complex system, requiring many decision makers with different objectives, dangerous containers storage problem is suitable for distributed solving techniques. The specification of mobility attached to our agents engaged us to use situated agents in the grid and to precise that elements are not fixed in a definitive cell into the CA. Nevertheless, the agents come in, depart, and move into the CA.

Design pattern MESSAGE (Methodology for Engineering Systems of Software AGent) is used to model our MAS Caire et al. (2001). In this pattern each agent is described with mental state entities, activities and concrete entities .

The two objectives of an agents are Mental state entities. At the strategic level, they are called purposes. The first one is to leave the system, the second one is to respect safety rules. The model is dedicated to focus on the second one. At the tactical level, to reach this objective, each agent owns a goal. This goal is to decrease the local fitness value (cf. section 4).

Following the model, to satisfy the agents' goals, processes must be attached to agents. They define an action. Indeed, each process is composed by tasks. Task is an activity and can be executed by another agent; it provides a service to achieve the action through interactions. Some agents can be considered as reactive agents and they are actually viewed like resources.

Resources are concrete entities, aiding the fulfilment of tasks.

### 5.3.2 Description

The aim is to satisfy container objective, that is why container centred model is developed. Consequently, containers are considered as agents and they attempt to comply their goals. Each agent have to be placed in a cell, in which its safety rules are respected. They also contribute to reach the global objective.

Container agents have to execute two processes. The first one is the negotiation phase, the second one is the movement phase. The negotiation phase is composed by the following tasks.

First, each agent computes its fitness value. We can restrict the number of partners (containers) interacting in the negotiation phase. Candidates are chosen as function of the fitness value. The next step consists in finding a destination for elected agent(s), the chosen container can be selected before this step or after to consider the fitness enhancement.

For example, agents compare their fitness with the global fitness value. Then, candidates agents with local fitness value equals to global fitness value, negotiate with each other in order to decide which agent of them will be moved.

Among the strategies intervening to decide the winner of the negotiation, the handling and moving cost can be considered. Handling includes the operating time and the number of shifts. The moving cost depends on the distance and also of the quality of attributes concerning speed and facilities of moving of different type of apparatus and ways. For example, moving into a block, through a row, is harder than driving into an avenue. This parametrization is difficult because the negotiation begins before deciding of the chosen equipments allocated to proceed the mission.

After that, the agent selected to be moved will execute the movement process. In this process, container searches new place better than its current position, and moves using resources.

The “search new place” task, can be in the first process or the second process. It depends on the strategy chosen and the agent situation.

Resources of this models are: empty places and straddle carriers. These resources can be viewed as reactive agents

This model allow us to test various strategies for the dangerous container placement or displacement on a terminal. These strategies depends on processes execution of agents.

To summary our model, first we reify the items composing a terminal, secondly we structure these objects using the CA architecture then, finally, we introduce agent based modelling to add communication protocols and behaviours.

## 6 Conclusion and perspectives

Positioning containers and, especially, hazardous ones in a terminal is a complex problem. Classical optimization approaches seem limited and/or intractable.

This paper proposes to study this situation with a container centred viewpoint mixing different approaches: cellular automaton structure for the space, object-oriented modelling for the architecture, and agent paradigm for the dynamicity.

This hybrid model driven development will permit us to test different strategies to place dangerous containers with an acceptance of security rules, and to permit local accommodations due to some uncertain events.

The principle to consider a container as an intelligent agent is not original but seems interesting, even if It is not the reality (by now) and it is not a goal in a close future. The aim is to study strategies with this model and, with successful solutions, to adapt them to present management systems.

We focus on the dynamicity and localization of the container transfers inside a cellular automaton representing the environment. The object-oriented architecture permits us to enhance the model adding new actors and resources. Then, the agent meta-model permits, considering the dynamicity and modularity of the processes, to introduce a rich panel of services and tasks.

In a first step, we consider our model as a decision support system to help experts to propose new strategies for container positioning. In a second step, the model can guide engineers to enhance the actual management systems used in port terminals introducing intelligent agents in the community of the containers.

## References

- Balev, S., F. Guinand, G. Lesauvage, and D. Olivier (2009, june). Dynamic handling of straddle carriers activities on a container terminal in uncertain environment - a swarm intelligence approach. In *ICCSA 2009 : the third International Conference on Complex Systems and Applications*, Volume 2, pp. 290.
- BusinessDictionary.com (2011). Dangerous goods: definition. <http://www.businessdictionary.com/definition/dangerous-goods.html>.
- Caire, G., W. Coulier, F. Garijo, J. Gómez-Sanz, J. Pavón, F. Leal, P. Chainho, P. Kearney, J. Stark, R. Evans, and P. Massonet (2001). Agent oriented analysis using message/uml. In *AOSE*, pp. 119–135.
- Davidsson, P., L. Henesey, L. Ramstedt, J. Tornquist, and F. Wernstedt (2005). An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies* 13(4), 255–271.
- Henesey, L., F. Wernstedt, and P. Davidsson (2003, June 24). Market-driven control in container terminal management. In *Proc. of the 2nd Int. Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT '03), Hamburg, Germany*, pp. 377–386.
- Institut du Droit International des Transports (2011). Réglementation sur l’entreposage des conteneurs de marchandises dangereuses dans les ports. <http://www.idit.fr>.
- International Maritime Organization (2011). International maritime dangerous goods (imdg) code.

[http://www5.imo.org/SharePoint/mainframe.asp?topic\\_id=158](http://www5.imo.org/SharePoint/mainframe.asp?topic_id=158).

- Kefi, M., O. Korbaa, K. Ghédira, and P. Yim (2007). Container handling using multi-agent architecture. In N. T. Nguyen, A. Grzech, R. J. Howlett, and L. C. Jain (Eds.), *KES-AMSTA*, Volume 4496 of *Lecture Notes in Computer Science*, pp. 685–693. Springer.
- Kim, K. and J.-S. Lee (2006). Satisfying constraints for locating export containers in port container terminals. In M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. Lagana, Y. Mun, and H. Choo (Eds.), *Computational Science and Its Applications - ICCSA 2006*, Volume 3982 of *Lecture Notes in Computer Science*, pp. 564–573. Springer Berlin / Heidelberg.
- Kim, K. H. and G.-P. Hong (2006). A heuristic rule for relocating blocks. *Computers & Operations Research* 33(4), 940 – 954.
- Lesauvage, G., S. Balev, and F. Guinand (2011, September). D<sup>2</sup>CTS : A dynamic and distributed container terminal simulator. In *The 14th International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation*.
- Rebollo, M., J. Vicente, C. Carrascosa, and V. Botti (2001). A mas approach for port container terminal management. In *3rd Iberoamerican workshop on DAI-MAS*, pp. 83–94.
- Salido, M. A., M. Rodriguez-Molins, and F. Barber (2010). Integrated intelligent techniques for remarshaling and berthing in maritime terminals. *Advanced Engineering Informatics* 25(3), 435–451.
- Schelling, T. (1978). *Micromotives and macrobehavior*. Norton, Toronto.
- Stahlbock, R. and S. Voß (2008). Operations research at container terminals: a literature update. *Or Spectrum* 30(1), 1–52.
- Thurston, T. and H. Hu (2002). Distributed agent architecture for port automation. In *COMPSAC*, pp. 81–90. IEEE Computer Society.
- Wolfram, S. (2002). *A new kind of science*. Champaign IL USA: Wolfram Media.