A Dynamic Risk Management in Chemical Substances Warehouses by an Interaction Network Approach

Omar Gaci[1] and Hervé Mathieu[2]
[1]*ISEL, Le Havre University*
[2]*LITIS - ISEL, Le Havre University*
France

1. Introduction

Supply chain is a set of activities involving a group of commercial actors to create a product or a service to satisfy a customer demand. The actors are the ones who form the supply chain, they are suppliers, transporters, manufacturers, distributors, retailers, customers. The objective of every supply chain is to maximize total supply chain profitability.

The Supply Chain puts in interaction a set of entities to provide to the final client the right product (or service) at the right time. Raw material suppliers, manufacturers of parts and components, assemblers, original equipment manufacturers, distributors, retailers, and customers are the main interacting entities of supply chain (SC) systems (Forrester, 1961).

In this chapter, we model Supply Chain as Complex Adaptive System (CAS) (Holland, 1996). CAS postulates that the activities of the constituting entities contribute to a specific emergence which corresponds to a global behaviour. Thus, the system is composed by active and adaptive intelligent agents. Their behaviours, interactions and adaptations lead to the emergence of the system behaviour.

We propose to study activities of storages in a warehouse of chemical substances. Then, this warehouse is subject to restriction in business processes executed every day: operators must respect a segregation strategy which consists in avoiding any mixing of incompatible chemicals. To reproduce the actions of forklift operators, we propose a Multi-Agent System which is the support of CAS modelling. Then, during agent movements for handling pallets from their reception into their storage locations, we define a dynamic graph where the vertices represent agents in activities and edges measure the distance between agents. The study of this dynamic graph shows that the average mean distance remains weak meaning that agent are often close each other. From this observation, we deduce a strategy for a dynamic risk management that gives the priority to agents whose betweenness is superior to the other agents that handle pallets of incompatible chemicals.

This chapter is organised as follows. In section 2, we present the main features of a Complex Adaptive Supply Chain and notably the existing support to simulate such a Supply Chain through Multi-Agent System. In section 3, we describe the existing solutions to reproduce

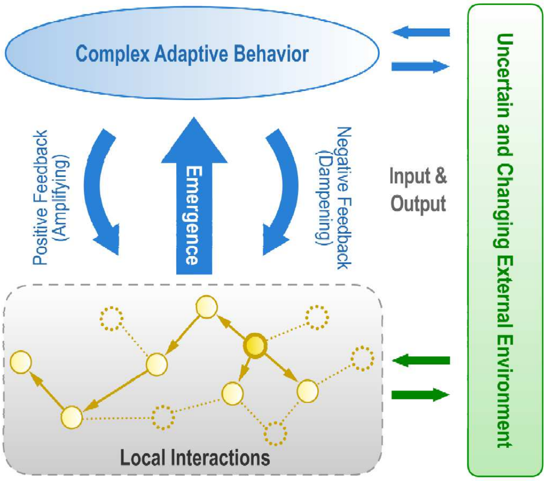


Fig. 1. Emergence from local interactions in Complex Adaptive Systems.

the activities of storages in a warehouse of chemical substances. In particular, we highlight the JADE framework for Multi-Agent simulations. In section 4, dangerous goods in logistics are studied and also the current regulation that warehouse must comply with. In section 5, we present our solution to reproduce the activities of storages in the studied warehouse by implementing a Multi-Agent system. In section 6, we study the dynamic graph resulting from agents handling actions and we deduce a dynamic risk management strategy.

This work is funded by German BMBF and French ANR as part of ReSCUeIT project.

2. Complex Adaptive Supply Chain

The theory of Complex Adaptive Systems (CASs) is presented by Holland (Holland, 1996) as a new paradigm to study the organizations and the dynamics of multi-scale systems whose evolution and adaptability leads to a global behaviour. A CAS can be considered as a multi-agent system with seven basic elements. According to Holland, the first four concepts are aggregation, nonlinearity, flow and diversity. They represent the characters of agents and influence the adaptability and the system evolution. The last three concepts, tagging, internal models and building blocks, are specific mechanics for agents to communicate with each other and also with the environment. The environment is itself subject to evolution notably because of the agent interactions which compete or cooperate from a same resource or for achieving a specific goal. As well, since the environment changes, the agents' behaviour evolve as consequence.

The main remarkable property which characterizes a CAS is the emergence of highly structured collective behaviour over time from the interactions of simple entities (Holland, 1996). The emergence of a complex adaptive behaviour from the local interactions of the agents is illustrated in Fig. 1. Then, the CAS and its environment evolve in the same time in order to maintain themselves in a state of quasi-equilibrium.

Considering a CAS consists in studying non-linear phenomena, non-exhaustive knowledge, a large state numbers and dynamic changes in environment. The main challenges when we reproduce a CAS are to produce a global behaviour by emergence under unpredictable conditions.

2.1 Complex Adaptive Systems as Multi-Agent Systems

In a Complex Adaptive System, CAS, a global behaviour emerges over time into a coherent form, adapting and organizing themselves without any singular entity controlling or managing the global structure or node interactions (Holland, 1996).

Complex Adaptive Systems are commonly implemented and simulated by Multi-Agent System, MAS, (Julka et al., 2002; Kwon et al., 2006; Swaminathan et al., 1997) which represents a general and flexible framework to describe and model autonomous systems including their interactions. An agent is basically a self-directed entity with its own goals and has a means to interact or to communicate with other agent.

2.1.1 Multi-Agent Systems

A MAS is formed by a network of computational agents that interact and typically communicate with each other.

The approach described by MAS consists in representing explicitly the individuals or the entities which compose the studied population. The system can then be ecological, social economic, etc. The goal is to produce a model for the entities, for the environment and for the mutual interactions. When the entities and their interactions are modelled, it remains to study the evolution of the relation through the simulation of their collective behaviour.

The understanding of the global MAS dynamic is viewed according to two levels, the microscopic (the study of the individual dynamics) and the macroscopic (the observation of the collective behaviour resulting of the entities interactions).

In (Conte, 1999), the authors propose two conceptual approaches deduced from the modelling of social phenomena:

- The top-down approach, which enables to deduce the microscopic phenomena, the goals or the individuals' motivations starting from the macroscopic observations;
- The bottom-up approach in which the hypothesis are established on the individual behaviours, their motivation or their way of interacting. The observation of their collective behavior is then compared to the macroscopic phenomena observed in the modelled system to eventually discuss the hypothesis formulated at a microscopic level (Epstein & Axtell, 1996). The bottom-up approach is specific to the most individual based models which propose such models to explain or characterize observed collective behaviours.

2.1.2 The agent properties

The agents considered in MAS are used in a broad variety of applications and are defined by the following way (Ferber, 1999):

The term 'agent' denotes a hardware or (more usually) software-based computer system, that has the following characteristics (Casterfranchi, 1995):

- *Autonomy*: an agent acts without any intervention from its environment and possesses rules to control its action and internal states;
- *Social ability*: an agent interacts and communicates with other agents using a specific agent-communication language;
- *Reactivity*: an agent perceives its environment and is able to answer to any solicitations;
- *Pro-activeness*: an agent is not only reactive to a stimulus from its environment, it is also able to exhibit goal-directed behaviour by taking the initiative.

2.1.3 The type of agents

Agents are defined by their capacities and according to these properties, different levels of complexity characterize agents. Such complexity depends on the task that agents have to carry out and on the environment surrounding them. In (Ferber, 1999), agents are classified according to their architectures:

- *Simple reflex agents*: These agents are basic because their actions depend on stimulus. Their act are then subject to specific conditions. The past is not considered and none memory influence the present reactions.
- *Model-based reflex agents*: These agents cannot perceive their whole environment but keep track of their environment they cannot currently observe. Then, they possesses an internal representation of their environment called 'model of the world' to evaluate the environment evolution and the impact of the agent's actions on this environment. These agents select their action according to condition-action rules. The conditions only depends on the model of the world, and not on the current perception of the environment.
- *Model-based, goal-based agents*: These agents have goals describing desirable situations to choose an action because the current state of the model of the world is not always enough to select an action efficiently. The model of the world represents a state from which the agents evaluate how the world would be after an action. The action is chosen in order to the agent goals are satisfied and the model of the world state is a parameter taken into account.

2.1.4 Applications of Multi-Agent Systems

MAS are commonly exploited to model and simulate one of the three followings types of applications:

- *MAS for studying complexity*. These studies regroup social models such as the segregation model of Schelling (Schelling, 1971) artificial life simulation with the Sugarscape (Epstein & Axtell, 1996) and Reynold's Boids models (Reynolds, 1971) and also logistics models (for example traffic simulations (Burmeister et al., 1997)). These models are built with simple reactive agents and a set of rules without any need of resource planning or coordination. The simulations are monitored relying on qualitative measures (emergent communities, emergent flocking, emergent behaviour) and/or quantitative (average generation, average agent movements, average awaiting time). These models are then studied as well from a top-down approach than with a bottom-up approach.
- *MAS for studying Distributed Intelligence*. These studies are relative to planning (Pollack & Ringuette, 1990) and particularly cognitive social interactions Doran (n.d.); Gilbert (2005); Sun (2001). The main goal is to reproduce human cognition through

cognitive agents (Sloman & Logan, 1999). These developed models use complex, situated and communicating agents to study the behaviour of cognitive formalism (Taatgen et al., n.d.; Wray & Jones, n.d.).

- *Application development with MAS*. Existing toolkits provide technical tools to develop software agents described in (Jennings et al., 1998). Software agents are then Semantic Web agents, Beliefs-Desires-Intention (BDI) agents in expert systems, or agents for network metamanagement. These toolkits include a development environment to implement MAS, it can be considered equivalent to a simulation engine.

Further, in this chapter, software agents are used and the JADE platform is used as framework for the development.

3. Multi-Agent System to study warehouse activities

The activities of modelling and simulating offer applications in scientific and industrial fields. These works improve the understanding and the reliability of design of various systems.

In the context of supply chain, the study of warehouse activities is motivated by the following goals:

- test by a software a virtual version of a warehouse before implementing and using the real system;
- collect information to support discussion with the customer;
- simulate the warehouse activities to improve business or security procedures;
- generate reproducible error situations.

In this chapter, we are interested in simulating storage activities in a warehouse of dangerous goods to evaluate the segregation policies efficiency and to propose a reliable risk management to maintain these segregation strategies.

3.1 Existing softwares in warehouse simulations

Over the years, different tools have been developed to help designers and users to model and simulate warehouse activities. Existing tools can be divided in three groups: GUI-based simulation softwares, framework libraries and specialized programming languages (Colla & Nastasi, 2010).

There is few simulation tools designed for the application of supply chain activities. Among them, we can cite the commercial tools eM-Plant. It can be used for visualization, planning and optimization of production and logistics. FlexSim (*Flexsim Simulation Software*, n.d.) is another commercial software which enables fast and easy modeling, clear visualization as well as reuseability of models.

The agent-based approach appears to be a powerful tool for the development of complex systems and is exploited in industrial applications (Weiming et al., 2000). This approach is used in many fields such as manufacturing, process control, telecommunication, air traffic control, transportation systems, information management, electronic commerce, etc.

Among the existing agent-based applications for the simulation of supply chain activities, we can cite Repast, SeSAm, NetLogo, SDML or AnyLogic.

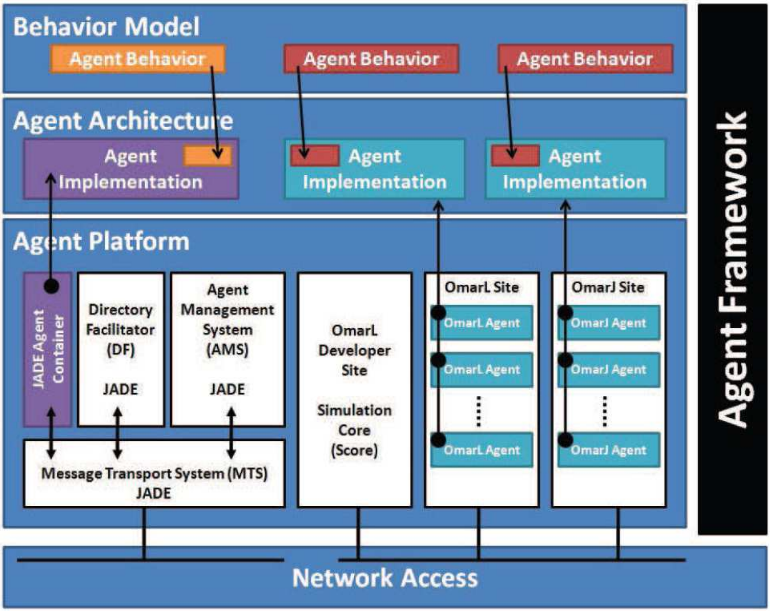


Fig. 2. Description of Agent Framework.

3.2 Existing frameworks for Multi-Agent warehouse simulations

It exists many different types of frameworks dedicated to the agent development. They are built from different theories and principles and allows then their classification. The definition of an agent framework is the following: an agent framework is a dedicated structure or platform to the development of software agents and based on a specific technical architecture.

An agent framework covers a set of missions relative to agents development, platform development, agent architecture and agent behaviour models as shown by Fig. 2. Then, an agent development platform is a structure which encompasses and support the entire life cycle of agents and provide in the same time a communication interface for agent interaction. This agent development platform commonly provides an API that defines the manner an agent communicate within the platform (Bellifemine, Caire, Trucco & Rimassa, 2007). As well, the agent architecture constitutes itself a framework for creating behaviour models. A behaviour model represents the architecture content and usually represents different forms of knowledge. The behaviour can be viewed as the result of the architecture and its content (Lehman et al., 2006).

The Java Agent Development Environment (JADE) is an agent platform used further in this chapter to implement a Multi-Agent System. JADE is a platform for the creation of MAS and contains a message transport system (MTS). This MTS, constitutes a network interface for developing distributed agent networks. As well, an Agent Management System (AMS) is available and allows the supervising of agent access control to the MTS and a directory facilitator (DF) for creating distributed services. In JADE, an agent is an instant that runs in the agent platform, then the agent has a determined life cycle by the AMS. Each agent is able to communicate with the other and possesses a queue for sending and receiving messages. An agent instance represents a container for the agent internal structure. The JADE agent platform is a middle-ware that complies with the specifications of the Foundation for Intelligent Physical Agents (FIPA) (Bellifemine, Caire, Trucco & Rimassa, 2007).

3.3 JADE platform for warehouse activities simulations

JADE is a software development framework fully implemented in JAVA language aiming at the development of multi-agent systems and applications that comply with FIPA standards for intelligent agents (Bellifemine, Caire & Greenwood, 2007). JADE is an agent framework and provides then a set of technical features to the development of MAS such as:

- *A distributed agent platform*. The platform can be easily shared and hosted in different machines when each machine possesses its own Java Virtual Machine;
- *FIPA-Compliant agent platform*. This means that the platform provides a set of functionalities such as Agent Management System, a Directory Facilitator and an Agent Communication Channel;
- *Communication with ACL messages*. The standard ACL ensures efficiently in the message transport between agents.

Communication of agents consists in sending and receiving messages, the FIPA ACL language is used to represent the messages. Each agent possesses an incoming message box and messages can be blocking or nonblocking during a determined blocking time. As well, JADE offers the possibility of filtering messages: it is possible to utilize advanced filters relative to different fields of the incoming message such as sender or ontology.

To build agent conversations, FIPA defines a set of standard interaction protocols such as FIPA-request and FIPA-query that can be exploited as standard for agent communication. When a conversation starts between two agents, JADE distinguishes two roles: the initiator who is the agent that starts the conversation and the responder who communicates with the previous one. This protocol architecture implies that the initiator sends a message and the responder can potentially reply by refusing the message indicating the incapability to continue with the conversation. The responder can also answer with a agreed message indicating that the communication between the two agents is established and can continue. After receiving a message, the responder performs potentially an action and must send back a message to describe such an action. In case that the action has failed, a failure message indicates that the action was not successful. JADE provides behaviour for initiator and responder roles according to FIPA interaction protocols. Then, the classes *AchieveREInitiator* and *Responder* provides homogeneous implementation of interaction protocols with methods for handling the different communication phases.

In JADE, agents actions or missions are implemented by the implementation of *behaviours*. These behaviours are defined as threads that can be composed or not, and allows agents to achieve their intentions. Such behaviours can be initialized, suspended and spawned at any given time. Then, the agents possess an action list that is executed through their behaviours. The JADE platform uses one thread per agent and not one thread per behaviour due to resource concerns (the number of running threads is limited). As well, a scheduler (unreachable for developers) organizes via a round-robin strategy the behaviours already created and instantiated in the queue. The coding of a behaviour offers the possibility of releasing the execution control when blocking mechanisms are used. The behaviours are executed in the method action().

The behaviour of agents is defined by a *Behaviour* class that can be specialized to defines a set of other behaviours. A behaviour is composed by several methods so that it is possible to describe the different state transitions. From this root behaviour, children behaviours can be deduced and notably the *SimpleBehaviour* and *CompositeBehaviour*. Behaviours that specialize or descend from *SimpleBehaviour* represent atomic simple tasks that can be executed several times according to the developer coding. As well, behaviours from *CompositeBehaviour*, are able to use multiple behaviours according to the children behaviours. Then, the agent tasks are executed not directly through the current behaviour but inside its children behaviours. For that purpose, the *FSMBehaviour* class executes the children behaviours. The *FSMBehaviour* class is able to maintain the transitions between states and to select the state coming after the current one. It is possible to register some of the children of an *FSMBehaviour* as final states. This type of behaviours terminates once one of its children has finished its execution.

4. Dangerous goods in logistics

A good is considered as dangerous when it may present a danger on the population, the environment or on the infrastructures according to its physicochemical properties or because of the reactions it can imply. A dangerous good can be flammable, toxic, explosive, corrosive or radioactive. According to the new CLP regulation, dangerous goods are considered as chemical substances in the European Union.

4.1 Dangerous goods identification

Considering the important number of substances, there is a clear need for dangerous goods classification. Amongst the existing classification of dangerous goods, the following distinctions exist:

- chemical family (acid, alcohol, amide, etc.);
- chemical reaction (oxidation, reduction, combustion).

We remark that the vocabulary becomes quickly specialized. To avoid this technical aspect, the dangerous goods are described in function of their reactions. Thus, the danger that represents the manipulation of dangerous goods depends on the properties of each product. Some goods represent only one risk whereas others regroup several.

The CLP (Classification, Labelling, Packaging) regulation is relative to the chemical substances imported or commercialized in the European Union. This regulation entered into force in January 2009 and will be totally applied in 2015.

4.1.1 Obligations under CLP

CLP provides a global obligation for all suppliers in the supply chain to cooperate. This cooperation is necessary to make the different suppliers meet the requirements for classification, labelling and packaging.

4.1.2 Terminology

A new terminology is used, terms of existing regulation are kept whereas news are adopted. The term substance is used to designed hazardous material and the transformation of these substances into a new one is called mixture.

As well, the properties of substances are described according to three properties: physicocochemical, toxicological and ecotoxicological. According to these three criterion, the definition of hazard classes helps to classify a substance. Then, a hazard class defines the nature of a hazard, it can be physical, on health or on the environment.

4.1.3 Classification of substances

CLP possesses specific criteria of classifications that are rules that allow associating a substance to a class of hazard or a category in this class. In particular, the classification process is based on the substance concentrations to establish the effects of those substances on the health and the environment.

CLP defines three hazard classes and 28 categories, such as:

- 16 categories for physical hazards;
- 10 categories for health hazards;
- 2 categories for environmental hazards.

For example, the physical hazards regroup explosives, flammable gases, solids, aerosols, liquids. The health hazards are relative to acute toxicity, skin corrosion, irritation and sensitization. The environmental hazards address hazardous to the aquatic environment and hazardous to the ozone layer.

4.1.4 Labelling

A substance contained in packaging should be labelled according to the CLP rules with the following information (called labelling elements):

- the name, address and telephone number of the supplier of the substance;
- the quantity of the substance in the packages;
- hazard pictograms;
- signal word;
- hazard statements;
- appropriate precautionary statements;
- supplemental information.

A substance contained in packaging is labelled according to the CLP rules and contains a set of information such as name of the supplier of the substance, quantity of the substance in the packages or hazard pictograms, see Fig. 3.

The CLP regulation helps then the identification of chemical substances through the supply chain since it provides a standard framework for the classification, the labelling and the packaging of substances.

Fig. 3. Pictograms used in CLP regulation.

4.2 Dangerous goods storage

Among dangerous goods, products can react violently when they are in contact. For these reasons, they must be stored in separate places. The strategy of storage consists in avoiding incompatible products to be neighbours. To avoid any storage of incompatible goods and risks of chemical reactions in case of wrong manipulation, segregation policies are established. Fig. 4, summarizes the incompatibilities between chemical substances.

Segregation policies in dangerous good warehouses consist in storing products according to their physicochemical properties. This strategy is static and doesn't take into account the possible movements of incompatible goods (by forklifts for example) that can be present in the same place at the same time.

The segregation can be achieved by the use of an impervious barrier or by a separation distance sufficient to prevent mixing. The segregation policies are also subject to constraint storages. According to the nature of goods, specific storage conditions must be respected. Among storage constraints, we can cite the most obvious such as storage conditions (humidity, heat and light). The respect of these constraints is ensured by safety equipments: sprinkler, smoke detector, particles detector or temperature probe.

Consequently, to make a warehouse of dangerous goods secured, different types of safety equipments are needed and a reliable segregation is used. In this chapter, we propose to simulate such a warehouse and to study the emergent collective behaviour of the MAS constituted by the warehouse actors.

Danger Code	F	F+	T	Xi	O	Xn	N	C
F	+	+	-	+	-	+	-	-
F+	+	+	-	+	-	+	-	-
T	-	-	+	+	-	+	-	-
Xi	+	+	+	+	-	+	+	-
O	-	-	-	-	+	-	-	-
Xn	+	+	+	+	-	+	-	-
N	-	-	-	+	-	-	+	-
C	-	-	-	-	-	-	-	+

Fig. 4. Identification of compatibilities between dangerous goods. The letter F means inflammable, F+ means very inflammable, T means toxic, Xi means very irritant, means O oxidizing, Xn means noxious, N means polluting and C means corrosive.

5. Simulation of warehouse activities by a Multi-Agent System implemented with JADE

Agent considered represent forklift drivers and the warehouse structure is then the agent environment. Fig. 5 shows the warehouse architecture which is composed of a forklift base, corridors, docks where pallets are temporally stored and five racks.

5.1 Forklift agent

Forklift agents are simple reactive agents that are positioned in their base and wait for a message from the central warehouse scheduler. This scheduler is actually a random generator which creates truck arrivals and sends messages to forklift driver agents so that they go to docks to unload the truck. Once the truck is completely unloaded, forklift driver agents continue their actions and store pallets in their rack position.

As shown by Fig. 6, agents react and communicate through messages. Firstly, agents are in their base and when they receive a message *GoToDock* they receive also the dock number and they consecutively move according to the *moveToDock(dockNum)* method. The agent motion follows warehouse corridors and this method provides to agents the set of corridors to use in order to reach the dock number *dockNum*. When the agents is in position, he confirms his position to the centralized scheduler and replies a message *atDock-DockNum*. This means that he is operational and the scheduler communicates with him to ask him to begin the unloading with an *Unload* message. If the scheduler has sent another message type, the agent would move back to the base. To unload the truck, the agent executes the *unloadTruck* method and confirms the end of handling operations with a message *EmptyT-dockNum*. Once more, the scheduler can choose to call back the agent to the base or to send him a message *Storage* so that the agent uses the method *storePalletsFrom(dockNum)*. This last method indicates to the

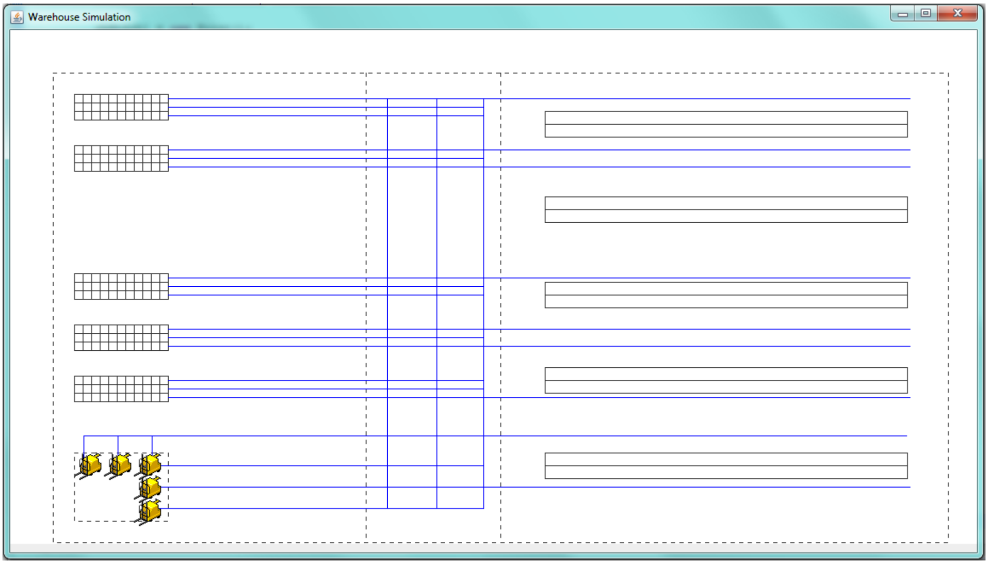


Fig. 5. Representation of the studied warehouse. Agents are located in their base and follow corridors to go to docks or racks.

agents the corridors to follow until the storage place in rack. When, the agent finishes the storage of pallets, he sends a message *StoredP-DockNum* and move back to his base.

The forklift driver agents evolve in a warehouse which represents their environment. They interact with a set of objects enumerated in the class diagram presented in Fig. 7. As shown, a forklift agent is a software agent defined according to an *ID* which is typically a number. His position is monitored in two dimensions and the current *Corridor* where he is evolving is given. As well, the *status* variable provides a means to know if the agent is in activity or if he is waiting in the base. In the class *Pallets*, the *hazardType* attribute gives the type of dangerous goods present on the pallet and it is the same for *Racks* that stores only restricted types of dangerous goods.

6. An interaction network approach for a dynamic risk management

The interaction network approach proposed in this section consists in monitoring in real-time the forklift agent movements and to detect a risk of incompatible chemicals mixing. To achieve such a goal, the warehouse is viewed as a dynamic graph where forklift agents who are active in the warehouse represent vertices that can be removed when they move back to their base. Edges link the vertices which represent agents and are weighted by the euclidean distance between agents in the warehouse. By this way, it is possible to consider a dynamic graph that puts in interaction forklift agents whose edges represent distance between them. The goal is then to detect the risk of incompatible chemicals mixing when the distance between agents is insufficient.

```
addBehaviour(new CyclicBehaviour(this) {
private static final long serialVersionUID = 1L;

public void action() {
int dockNum;
String msgType1, msgType2, msgType3;
ACLMessage msg1, msg2, msg3;
ACLMessage rep1, rep2, rep3;

// Waiting message from scheduler
msg1 = receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
if (msg1 != null){
  msgType1 = msg1.getContent().substring(0,8);
  if (msgType1.equals("GoToDock")){

    dockNum = Integer.parseInt(""+msg1.getContent().substring(8,9));

    moveToDock(dockNum);     //move to dock
    rep1 = msg1.createReply();
    rep1.setContent("atDock"+dockNum);
    send(rep1);              // moved at dock

    msg2 = receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
    if (msg2 != null){
      msgType2 = msg2.getContent().substring(0,8);
      if (msgType2.equals("Unload__")){

        unloadTruck(dockNum);   //unload pallets from a truck
        rep2 = msg2.createReply();
        rep2.setContent("EmptyT"+dockNum);
        send(rep2);             // pallets unloaded from the truck
        }

        msg3 = receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
        if (msg3 != null){
          msgType3 = msg3.getContent().substring(0,8);
          if (msgType3.equals("Storage_")){

            storePalletsFrom(dockNum);// store pallets from a dock
            rep3 = msg3.createReply();
            rep3.setContent("StoredP_"+dockNum);
            send(rep3);             // pallets stored from the dock
         }
        }
       }
     }
goToBase();
}
```

Fig. 6. Behaviour of forklift agents. They react after receiving messages and confirm their action by replying.

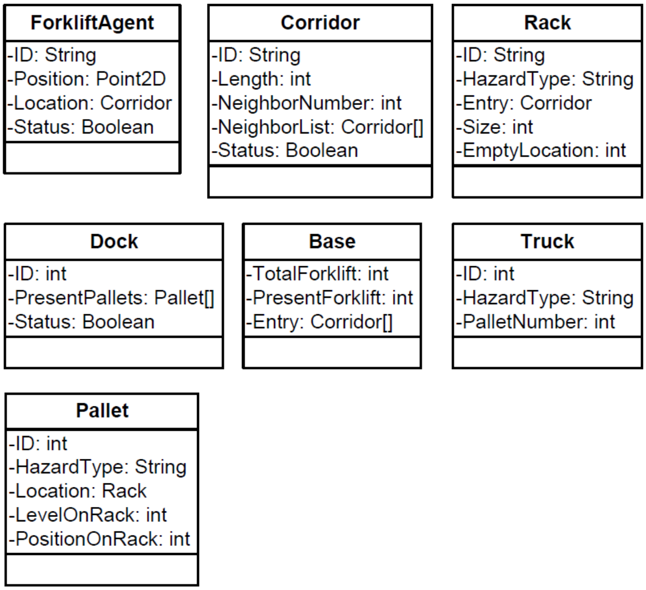


Fig. 7. Classes used to reproduce the activities of storage in a warehouse of chemical substances.

6.1 Dynamic graphs

Many systems, both natural and artificial, can be represented by networks, that is by sites or vertices bound by links. The study of these networks is interdisciplinary because they appear in scientific fields like physics, biology, computer science or information technology. The purpose of these studies is to explain how elements interact inside the network and what are the general laws which govern the observed network properties.

From physics and computer science to biology and the social sciences, researchers have found that a broad variety of systems can be represented as networks, and that there is much to be learned by studying these networks (Broder et al., 2000). Indeed, the study of the Web (Albert et al., 1999), of social networks (Wasserman & Faust, 1994) or of metabolic networks (Jeong et al., 2000) are contribute to put in light common non-trivial properties to these networks which have *a priori* nothing in common. The ambition is to understand how the large networks are structured, how they evolve and what are the phenomenon acting on their constitution and formation (Watts & Strogatz, 1998).

Nevertheless, to study the dynamic of a phenomenon through a graph, we need tools able to describe the graph topology evolution over time. The works relative to random graphs (Erdos & Rényi, 1959) provide a generic dynamic model which describe graphs whose edges are added according to a specific probability.

More recently, the interest for dynamic graphs has increased notably because of their potential application in communication, urban traffic or social sciences. The dynamic graphs allow studying the graph topology evolutions relying on dynamical metrics able to describe the graph properties when it evolves over time.

Now, we give some graph theory definitions to propose a definition of dynamic graphs.

A graph G is formally defined by $G = (V; E)$ where V is the finite set of vertices and E is the finite set of edges each being an unordered pair of distinct vertices.

Let f be a function defined on the vertex set as $f : V \to N$, then the triple $G = (V; E; f)$ is a node weighted graph. As well, let g be the function defined on the edge set as $f : E \to N$, the triple $G = (V; E; g)$ is an edge weighted graph.

In (Harary & Gupta, 1997), the authors classify the dynamic graphs as a function of the graph evolution:

- Node dynamic graphs, the vertex set V changes over time
- Edge dynamic graphs, the edge set E is modified over time
- Node weighted dynamic graphs, the f function evolves over time
- Edge weighted dynamic graphs, the g function varies over time

6.2 Graph metrics

Different graph measures allow characterizing graphs. Here, the proposed metrics provide measures for global description and also for individual vertices so that it is possible to identify the influence of a vertex in the modelled warehouse.

6.2.1 Distance and diameter

The distance in a graph $G = (V, E)$ between two vertices $u, v \in V$, denoted by $d(u, v)$, is the length of the shortest path connecting u and v.

A graph diameter, D, is the longest shortest path between any two vertices of a graph:

$$D = \max\{d(u, v) : \ u, v \in V\}$$

The mean distance is defined as the average distance between each couple of vertices:

$$L = \frac{2}{n(n-1)} \sum_{u,v \in V} d(u, v)$$

6.2.2 Mean degree

A degree of a vertex u, k_u, is the number of edges incident to u. The mean degree, z, of a graph G is defined as follows:

$$z = \frac{1}{n} \sum_{u \in V} k_u = \frac{2m}{n}$$

6.2.3 Node betweenness

The betweenness of a node is defined as the total number of shortest paths between pairs of nodes that pass through this node. It measures the influence of a node in a network. The betweenness of a node t, denoted $B(t)$ is defined as follows:

$$B(t) = \sum_{u \neq v, u \neq t, v \neq t} \frac{\sigma_{uv}(t)}{\sigma_{uv}}$$

where σ_{uv} is the number of shortest paths between the nodes u and v, and $\sigma_{uv}(t)$ is the number of shortest paths between u to v that pass through t.

6.3 Simulation and results

We assume that the warehouse is well dimensioned and at last one agent is available to perform a truck unloading or a pallet storage. Once a single or several agents react, they perform handling operations according to their behaviours. Then, agents are located in their forklift base and wait for a truck arrival. When a truck is in position, agents react and move into a dock. Another reaction is needed in order to agents unload the truck. The movements of agents follow the warehouse corridors. The time spent by agents to unload a truck or to store pallets is the average time observed in the real warehouse. The objective is to simulate a behaviour close to the reality.

In this case study, we consider that the warehouse stores three types of chemical substances denoted A, B and C. Each product must be stored only in its rack location and the segregation consists in avoiding that a product is stored in another rack. We consider 6 forklift agents and 10 trucks with a cargo of 33 pallets by truck. We launch a simulation and we study the dynamic graph resulting for agents activities.

Fig. 8 shows the evolution of the mean distance denoted l and the diameter, D. It appears that the means distance evolves between 30 and 150 which is the consequence of the warehouse dimensions. As well, the diameter being an upper bound of distances in interaction networks, we expect that the mean distance l will be lower than D. The mean degree is studied in Fig. 9 and shows that it evolves between 1 and 6. This means that at least two agents are in activities in the warehouse and when they are all outside their base, the mean degree will be 6.

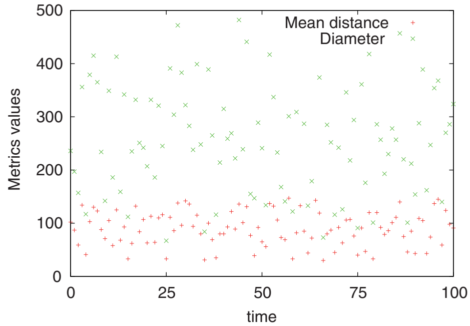


Fig. 8. Mean distance and diameter of the resulting graph from agent activities.

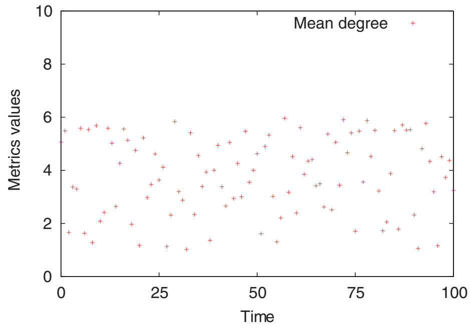


Fig. 9. Mean degree of the resulting graph from agent activities.

This first study about the dynamic graph resulting from agents handling operations put in evidence that they are not all present in same time in the warehouse. The average distance between agents is still weak in front of the upper bond expressed by the diameter.

Our goal is to develop a dynamic risk management strategy to maintain a segregation during the agent movements. In front of results presented above, we deduce a strategy presented in Algorithm 1. Then, when two agents are present in the same corridor, the type of handled goods determines if these forklift agents can share this corridor. In case that incompatible products are transported by agents, a topological measure is exploited, typically the node betweenness, to determine the priority between these agents. We consider that the agent

Algorithm 1: Algorithm to maintain a dynamic segregation between agents

Input:
F_a: set of forklift agents
Data:
$f_a, f_a^{'}$: forklift agent
$B(f_a)$: betweenness of the current agent
$HazType(f_a)$: hazard type of chemicals handled by the current agent
$Corr(f_a)$: corridor length of the current agent
$reorienteAgent(f_a)$: current agent is reoriented into another corridor

```
begin
    foreach f_a ∈ F_a do
        foreach f'_a ∈ F_a do
            if dist(f_a, f'_a) < Corr(f_a) then
                if HazType(f_a)andHazType(f'_a) are incompatible then
                    if B(f_a) > B(f'_a) then
                        reorienteAgent(f'_a)
                    else
                        reorienteAgent(f_a)
```

whose betweenness is superior, has the priority and the other agent is reoriented into another corridor. If the next corridor is in the same configuration, the agent will change again until be in presence of incompatible chemicals. Therefore, the dynamic risk management strategy is defined as a prevention of incompatible chemicals crossing in corridors. Any risks of crossing or mixing is mitigated by the routing of agents into another corridor when this agent possesses a weaker betweenness that the other one.

7. Conclusion

In the global context of logistics and supply chain management, we are interested in the manner to model the SC. A Complex Adaptive Model, CAS, approach is then well studied for modelling supply chain systems considering the structural and behavioural dynamics. In a CAS, the interactions of the agent population and the environment evolution contribute to the emergence of a global behaviour.

This chapter presents an approach to study warehouse of chemical substances involving human actors. We have modelled the activities and the actors to implement a Multi-Agent System, MAS from which we want to reproduce segregation violation during the goods movements. Then, the warehouse becomes a CAS where agents accomplish their goals (typically handling operations) and whose mutual interactions are susceptible to violate segregations.

We propose a dynamic graph to describe the agents movements in the warehouse. Then, vertices represent agents when they are in activities and removed once they move back t their base. Edges are defined between vertices and are weighted by the distance between agents. The study of this graph by topological measures such as the average distance, the diameter and the mean degree show that agents are effectively close each other during their handling operations. We deduce a dynamic risk management to maintain segregation even when chemical substances are handled by agents. Thus, when the distance between incompatible goods is insufficient, a study of the two involved agents node betweenness determine what agent is redirected into another corridor. By this way the crossing and the mixing of incompatible goods is mitigated.

8. References

Albert, S., Jeong, H. & Barabasi, A.-L. (1999). The diameter of the world-wide web, *Nature* Vol. 401(No. 4): 130–131.

Bellifemine, F., Caire, G., Trucco, T. & Rimassa, G. (2007). Jade programmer's guide.

Bellifemine, F. L., Caire, G. & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*, Wiley, MA, USA.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. & Wiener, J. (2000). Graph structure in the web, *Computer Networks* Vol. 33(No. 1): 309–320.

Burmeister, B., Haddadi, A. & Matylis, G. (1997). Application of multi-agent systems in traffic and transportation, *IEE Proceedings of Software Engineering*, IEEE Computer Society, London, pp. 51–60.

Casterfranchi, C. (1995). Guarantees for autonomy in cognitive agent architecture, *in* M. Wooldridge & N. R. Jennings (eds), *Lecture Notes in Computer Science: Intelligent Agents*, Springer-Verlag, Heidelberg, Germany, pp. 56–70.

Colla, V. & Nastasi, G. (2010). Modelling and simulation of an automated warehouse for the comparison of storage strategies, *in* G. Romero Rey & L. Martinez Muneta (eds), *Modelling Simulation and Optimization*, InTech, Rijeka, Croatia, pp. 471–486.

Conte, R. (1999). Social intelligence among autonomous agents, *Computational & Mathematical Organization Theory* Vol. 5(No. 3): 203–228.

Doran, J. (n.d.). Can agent-based modelling really be useful?, *in* N. J. Saam & B. Schmidt (eds), *Cooperative Agents: Applications in the Social Sciences*, Springer, pp. 57–81.

Epstein, J. & Axtell, R. (1996). *Growing Artificial Societies, Social Science from the bottom up*, Cambridge MA MIT press, MA, USA.

Erdos, P. & Rényi, A. (1959). On random graphs I, *Publicationes Mathematicae* Vol. 6(No. 1): 290–297.

Ferber, J. (1999). *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*, Harlow: Addison Wesley Longman, MA, USA.

Flexsim Simulation Software (n.d.).
URL: *http://http://www.flexsim.com/*

Forrester, J. W. (1961). *Industrial Dynamics*, Cambridge MA MIT press, MA, USA.

Gilbert, N. (2005). When does social simulation need cognitive models, *in* R. Sun (ed.), *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, Cambridge University Press, Cambridge, UK, pp. 428–432.

Harary, F. & Gupta, G. (1997). Dynamic graph models, *Mathematical and Computer Modelling* Vol. 25(No. 7): 79–87.

Holland, J. H. (1996). *Hidden order: how adaptation builds complexity*, Reading, MA, Addison Wesley, MA, USA.

Jennings, N. R., Sycara, K. & Wooldridge, M. (1998). A roadmap of agent research and development, *Autonomous Agents and Multi-Agent Systems* Vol. 1(No. 1): 7–38.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabasi, A.-L. (2000). The large-scale organization of metabolic networks, *Nature* Vol. 6804(No. 407): 651–654.

Julka, N., Karimi, I. & Srinivasan, R. (2002). Agent-based refinery supply chain management, *Computer Aided Chemical Engineering* Vol. 26(No. 12): 1771–1781.

Kwon, O., Im, G. P. & Lee, K. C. (2006). Mace-scm: A multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties, *Expert Systems with Applications* Vol. 33(No. 3): 690–705.

Lehman, J. F., Laird, J. & Rosenbloom, P. (2006). A gentel introduction to soar, an architecture for human cognition.

Pollack, M. E. & Ringuette, M. (1990). Introducing the tileworld: Experimentally evaluating agent architectures, *Proceedings the Eighth National Conference on Artificial Intelligence*, AAAI Press, Boston, MA, USA, pp. 183–189.

Reynolds, C. W. (1971). Flocks, herds and schools: A distributed behavioural model, *Computer Graphics* Vol. 21(No. 4): 25–34.

Schelling, T. (1971). Dynamic models of segregation, *Journal of Mathematical Sociology* Vol. 1(No. 2): 143–186.

Sloman, A. & Logan, B. (1999). Building cognitively rich agents using the sim agent toolkit, *Communications of the Association of Computing Machinery* Vol. 42(No. 3): 71–77.

Sun, R. (2001). Cognitive science meets multi-agent systems: A prolegomenon, *Philosophical Psychology* Vol. 14(No. 1): 5–28.

Swaminathan, J. M., Smith, S. F. & Sadeh, N. M. (1997). Modeling supply chain dynamics: A multi-agent approach, *Decision Sciences* Vol. 29(No. 3): 607–632.

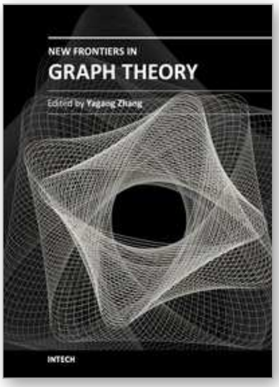
Taatgen, N., Lebiere, C. & Anderson, J. (n.d.). Modeling paradigms in act-r, *in* R. Sun (ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modelling to Social Simulation*, Cambridge University Press, pp. 29–52.

Wasserman, S. & Faust, K. (1994). *Social network analysis: methods and applications*, Cambridge University Press, Cambridge, UK.

Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks, *Nature* Vol. 393(No. 6684): 440–442.

Weiming, S., Qi, H., Hyun, J. Y. & Douglas, H. N. (2000). Applications of agent-based systems in intelligent manufacturing: An updated review, *Advanced Engineering Informatics* Vol. 20(No. 4): 415–431.

Wray, R. E. & Jones, R. M. (n.d.). An introduction to soar as an agent architecture, *in* R. Sun (ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modelling to Social Simulation*, Cambridge University Press, pp. 53–78.



New Frontiers in Graph Theory
Edited by Dr. Yagang Zhang

ISBN 978-953-51-0115-4
Hard cover, 526 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

Nowadays, graph theory is an important analysis tool in mathematics and computer science. Because of the inherent simplicity of graph theory, it can be used to model many different physical and abstract systems such as transportation and communication networks, models for business administration, political science, and psychology and so on. The purpose of this book is not only to present the latest state and development tendencies of graph theory, but to bring the reader far enough along the way to enable him to embark on the research problems of his own. Taking into account the large amount of knowledge about graph theory and practice presented in the book, it has two major parts: theoretical researches and applications. The book is also intended for both graduate and postgraduate students in fields such as mathematics, computer science, system sciences, biology, engineering, cybernetics, and social sciences, and as a reference for software professionals and practitioners.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Omar Gaci and Hervé Mathieu (2012). A Dynamic Risk Management in Chemical Substances Warehouses by an Interaction Network Approach, New Frontiers in Graph Theory, Dr. Yagang Zhang (Ed.), ISBN: 978-953-51-0115-4, InTech, Available from: http://www.intechopen.com/books/new-frontiers-in-graph-theory/a-study-of-dangerous-goods-warehouses-by-an-interaction-network-approach

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821