



HAL
open science

A simulation approach for risk modeling and analysis based on multi-agents

H Kanj, Jean-Marie Flaus

► **To cite this version:**

H Kanj, Jean-Marie Flaus. A simulation approach for risk modeling and analysis based on multi-agents. ESREL 2015, 25th European Safety and Reliability Conference, Sep 2015, Zürich, Switzerland. 10.1201/b19094-514 . hal-01276120

HAL Id: hal-01276120

<https://hal.science/hal-01276120>

Submitted on 22 Feb 2016

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.

A simulation approach for risk modeling and analysis based on multi-agents

H. Kanj & J.-M. Flaus

Univ.Grenoble Alpes, G-SCOP, F-38000 Grenoble, France

CNRS, G-SCOP, F-38000 Grenoble, France

ABSTRACT:

Agent-based modeling and simulation is an interesting approach to modeling systems comprised of autonomous, interacting agents. Computational advances have made possible a growing number of agent-based models across a variety of application domains.

This paper describes a generic approach to use agent-based modeling for risk analysis. It presents a novel generic model facet for representing risk analysis and fault tree propagation in an agent model, where the goal is to simulate the behavior of a system in normal and degraded mode by using multi-agents systems.

1 INTRODUCTION

Risk analysis is the discipline to identify and assess risks for a system, activity or process according to the severity of their consequences which may vary from moderate to disastrous. Today, many methods are available and widely used in order to conduct risk analysis [Flaus, 2013a] such as Preliminary Hazard Analysis (PHA) and Failure Mode Effects Analysis (FMEA)[Papadopoulos et al., 2004].

In [Kanj and Flaus, 2014], we have proposed a meta model suited to socio-technical systems. It allows the description of the system to analyze and provides risk analysis results, and required aspects of a dynamical system behavior in order to automatically perform simulation under degraded conditions.

In this work, we propose to analyze the system's performance in degraded mode. For complex systems, this analysis can only be made via simulation. A simulation model may be considered as a set of rules (e.g. equations, flowcharts, state machines, cellular automata) that define how the system being modeled will change in the future, given its present state. Simulation is the process of model "execution" that takes the model through (discrete or continuous) state changes over time.

Many approaches may be used for Simulation modeling, such as :

- System Dynamics (SD), developed by the electrical engineer Jay W.Forrester in the 1950s. It is mainly continuous and is characterized by a high abstraction level, low details and a strategic level ;
- Discrete Event (DE), which roots back to 1960s

when Geoffrey Gordon conceived and evolved the idea for GPSS and brought about its IBM implementations [Gordon, 1961]. This model is based on the concept of entities, resources and block charts describing entity flow and resource sharing, it is mainly discrete and characterized by a middle abstraction level, medium details and a tactical level ;

- Agent Based (AB), which is known by many names. ABM (agent-based modeling), ABS (agent-based systems or simulation), and IBM (individual-based modeling) are all widely used acronyms. These models are essentially decentralized and preferred for complex systems. They can range from high to low abstraction levels.

As the systems we consider are complex in terms of multiplicity of units, decentralization of decision making, and number of relationships between its components, we will use an "ABS".

In this paper, we propose a generic model facet for representing risk analysis and fault propagation in an agent model. This facet can be built in a systematic manner from model based risk analysis and is made of the following elements :

- a set of behavioral modes and an associated activity model represented as an activity UML model.
- a set of events and a dysfunctional model represented by a bow tie model.
- a set of transition rules to describe the interaction between these models

The remainder of this paper is organized as follows. Section 2 we give a basic representation of the multi-agents systems (MAS) and define the agents, their

attributes and their relations. Section 3 presents the second part of our model, which is built while we perform the risk analysis. In this section we explain how to adapt a multi-agents system to risk analysis. Section 4 presents an illustration of this approach for risk analysis for dangerous good transportation. Finally, Section 5 concludes this paper.

2 MULTI-AGENT SYSTEM (MAS)

2.1 Definition of an agent

In literature, a variety of definitions exist to describe what an agent is. One example, described in [Ferber and Perrot, 1995], defines an agent as a physical or virtual entity :

- which is able to act in an environment,
- that can communicate directly with other agents,
- which is driven by a set of trends (in the form of individual goals or function of satisfaction, even survival), it seeks to optimize,
- which has its own resources,
- which is able to receive (to a limited extent) its environment,
- which may or may not have only a partial representation of this environment,
- which is expert and provides services,
- which can eventually reproduce,
- whose behavior tends to meet its objectives, taking into account the resources and expertise available to it, and according to its perception, its representations, and communications it receives.

Another definition is that proposed in [Wooldridge and Jennings, 1995] which defines the following characteristics of an agent :

- Autonomy : an agent operates alone without the direct intervention of a third party (human or otherwise), and undergoes no control over the actions carried out or on its internal state.
- Social capacity : agents interact with each other (and probably with humans) through communication languages and common sociability rules.
- Reactivity : the agent perceives its environment (physical or model) and response in a timely manner to changes that occur in it.
- Pro-activity : agents do not simply react to stimuli from their environment, they must be able to show behavior directed by internal goals and this by taking initiatives.

2.1.1 Properties of an agent

The concept of agents is closely related to the object oriented approach used in modern programming languages such as Java or C++. Thereby an object is defined by its states and behavior, whereas agents can be seen as objects with more extended capabilities (see Figure 2.1) (e.g., rules of behavior, autonomy, cooperation (e.g., perception, action, communication),

mobility, memory, learning ability, among others. Cooperation, which is considered as a core capability of an agent, comprises, e.g., perception and action (interaction) and communication.

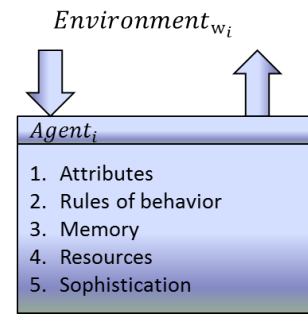


FIGURE 2.1: Agent properties

2.2 Definition of MAS

The multi-agent system is a software technology in great demand to model and simulate the dynamic behavior of complex and decentralized systems.

Historically, these systems are positioned at the intersection of programming (software), artificial intelligence (decision-making autonomy) and distributed systems (decentralization). The field of multi-agent systems has emerged in the 80s under the name of artificial intelligence (Huhns, 1987) in the USA and decentralized artificial intelligence (Demazeau and Müller, 1990) or multi-agent systems Europe.

Agent-based modeling (ABM) is an upcoming approach in complex systems science to model structures comprising autonomous and interacting elements. Some scientists even denote this computer simulation based modeling approach "A New Kind of Science" and argue, that besides deduction and induction, ABM and simulation is a third way of doing science.

The basic idea of ABM is to model only the units - called agents - of a specific system and to simulate their interplay in order to derive and analyze the total system behavior.

Building an agent based simulation is composed of three parts :

- Define the set of agents and the environment which contains all agents : we define for each agent a set of attributes, behavior modes, and resources, and for the environment, we define its characteristics, responsibility and functions ;
- Identify for each functions, its characteristics and attributes ;
- Specify the interactions in the system.

2.3 Interactions between agents

In a multi-agent system, an agent is in mutual interaction with other agents. Two types of communication can be observed for an agent :

- agent-agent (direct communication) : is carried in an intentional way by sending messages to one or more well defined recipients ;
- environment-agent or indirect communication : is carried either through the environment (not intentional action), leaving traces or signals, or through a chalkboard (intentional action) filing and reading information filed in a shared data area. In this type of communication, the recipients are not defined.

3 AGENT-ORIENTED RISK

To adapt the model-based agents to our risk analysis, we added some additional characteristics to our classical agent in order to obtain a new risk oriented agent architecture (see Figure 3.1) :

1. behavioral modes : described by a list of successive blocks
2. failures modes : for expressing the risk and dysfunction analysis of our agents
3. relationships between failure modes and patterns of behavior of our agents.

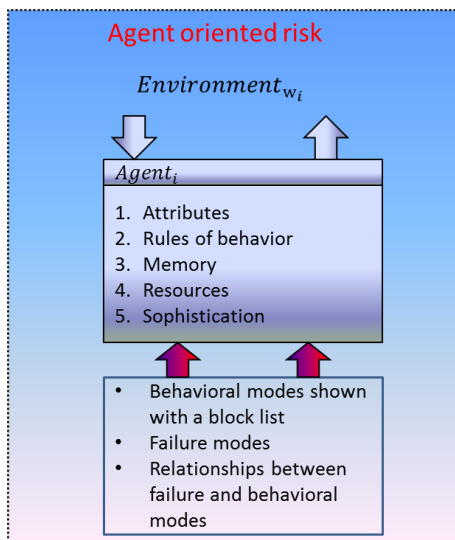


FIGURE 3.1: Agent oriented risk

Then we are used the FIS modeling tool to model our multi-agent system, The structure of each agent is composed of three views :

1. Structural view
2. Dysfunctional view
3. Behavioral View

3.1 Structural view

This part describes the structure of an agent, by showing its relations with its environment (other agents) and the characteristics of each of them. This basic view, allows us to enter into the dysfunctional or behavioral view.

3.1.1 Structure of the agent

The basic element is the agent, seen as an intelligent entity capable of interacting with its environment and with other agents, execute tasks described by an organized set of activities that use resources or agents (personnel, equipment and machinery, ...) to transform inputs into outputs agents. It is described internally by :

1. Structural elements that are defined with a list of properties of accumulated resources and behavioral rules.

The property list of an agent may contain :

- a type, that defines the class under which the agent belongs ;
- a name, which specifies the name to identify the agent from the other agents in its environment ;
- a list of variables or characteristics : for each characteristic, define a name, a type and a domain of validity.

The rules of behavior vary by agent. They include :

- The rules of sophistication
- The cognitive load
- The internal models of the external world
- The memory used

2. The *functional elements* (FE), called *activities* or *tasks* for an agent. An activity can be either active or inactive at a given time t. Its behavior is characterized by a set of variables, and eventually by a behavioral model describing the temporal evolution.

3.1.2 Links between activities and resources

Relationships can take place between the functional elements of an agent and other agents in its environment, which is necessary to perform an activity (see Figure 4.1). These relationships are of three types :

- input relations : sometimes a number of agents is requested in entering an activity. These agents are either consumed or provide information that is used ;
- outputs relations : other agents are either generated or their status are changed, it is the case of a material agent that is produced, or object that is assembled agent, ... ;
- relationships of use : in this case, the agents are considered supports for the function. For example, to advance, a truck agent use a driver agent.

3.1.3 Relationships between agents

Many relationships can be defined between agents. We distinguish :

- composition : during their lifetime, agents can be joined to form a super agent who possess all of their properties, memory resources and rules of behavior ;

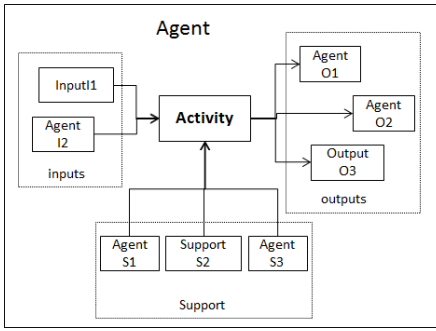


FIGURE 3.2: Relations

- communication : since agents are intelligent systems, they can communicate with their environment, respond to changes and exchange knowledge with other agents, information and data ;
- decomposition : an agent can decompose to give birth to several other agents possess a copy of the knowledge and father Agent resources with different attributes ;
- influence : agents can influence each other especially if they are at a certain point in the same geographical area, the relationship of influence can lead to other types of relationships such as : loss of an agent (composition), creation of an agent (decomposition) or a data change agent (communication).

3.2 Dysfunctional view

Dysfunctional view describes the propagation of failure between model elements. For example, a failure of an input or support resources for an activity may cause the failure of the activity or may produce an output resources in a degraded state. Each failure is represented by a fault tree (see Figure 3.3), where events are related to each others by links representing causal relationships and connectors such as AND and OR connectors. Two categories of failure sources for an activity can be observed :

1. Failure related to the activity : in this case, the failure becomes from its inputs, supports resources ;
2. Exterior failure related to others agents or to the environment : usually, this kind of failure requires a common resource or a communications between the failing agent and the agent executed the activity, for example, during the advancement of a truck, a collision between two vehicles (exterior failure mode) at the road (common resources between the truck agent and the vehicles) may lead to a delay during the truck advancement ;

Once, this view is built, we can define the set of failure modes for each agent's activity in the system. Afterward, we add the set of exterior failure in order to obtain for agent's activity a set of failure mode.

Each failure mode is described by :

- a name ;
- a set of events, causes for this mode, connected using AND and OR gates represented in the fault

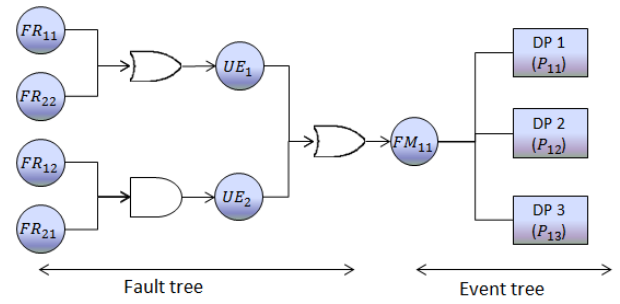


FIGURE 3.3: fault and Event Tree

tree (see Figure 3.3) ;

- a set of events ,consequences of the failure, representing in the event tree (see Figure 3.3) ;
- and a time t representing the activation time of this mode.

Failure mode is denoted by $fmode_i$. The normal mode of an activity is defined as :

$$ok \bigwedge_i \neg dfmode_i$$

3.3 Behavioral view

This view describes the dynamic behavior of a multi-agent system. Once, we have this view, it is possible to simulate the operation of the system, in normal or degraded modes. In addition, some elements of this view can be exploited for diagnosis or prognosis [Flaus, 2013b, Giap et al., 2009].

3.3.1 Activity Diagram

The behavior of each agent is described by an activity block diagram (see Figure 3.4). For an agent, many behavioral modes are possible, however, at a precise time t , only one mode is active while the others are inactive. A behavioral mode represent a program for the agent with many input parameters. A mode has

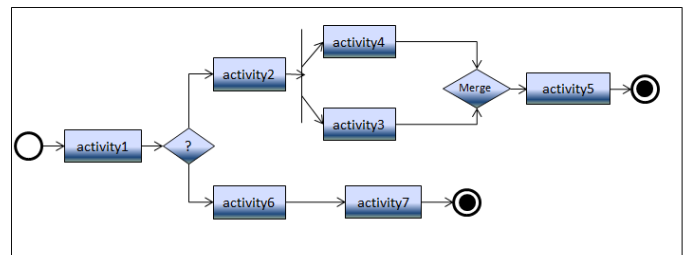


FIGURE 3.4: Behavioral mode of an agent

a current state (active, inactive) which is updated in each step of the simulation. It may contain many types of blocks :

1. A start event block : This is the first block executed by an agent, it indicates the existence of a new agent in the environment ;
2. An end event block : This is the last block executed by an agent, it indicates the end of life of an agent ;

3. An activity block : Each behavioral mode is associated with one or more activities, which can be of the following types :
 - a simple modification activity of the informational state, described by a mathematical relationship and an execution duration (may be null). In general, this type of block is intended for the execution of simple tasks (an affecting statement, increment of a variable) as follows : $x = 1$.

- a continuous action : this type of block is generally used in simulation with continuous events , where it is important to represent the change of variables over time. The following equation shows a usage example of using of this block : $x(k+1)=x(k)+b$

More precisely, this kind of activity is defined by $a = name, x \leftarrow f_a(x), \delta$ where x is a vector of variables defined as the model element or the set of model elements via a structural link $N_s(me)$, and δ is a positive integer representing the time $t = \delta.T_s$.

- a transformation activity : this type of activity involves interaction between an agent and other agents in its environment or between the agent and its environment, and can lead to changes in the characteristics of the agents and the environment. It can be considered as an entity processor and used to represent the processing flow diagrams (see Figure 3.5). It is defined by :
 - a transformation relation :

$$\sum n_i me_i | \phi_i \dots \rightarrow \sum n_s me_s | \phi_s \rightarrow \sum n_o me_o | \phi_o \{ m_i \}$$

where me_i is an input model element, required in number n_i and which must satisfy the condition ϕ , which a logical relation expressed with respect to the variables, the attributes and the events available in the scope. me_o is an output and me_s is a model element representing a support ;

- a positive integer duration δ representing the time $t = \delta.T_s$, and/or a set of final conditions ;
- a set of input actions ;
- a set of output actions ;

It is also possible for an interaction of this kind to create new agents in the environment or kill existing agents. This type of activity is sometimes complex. We can decompose it into a list of simple activities.

4. A gateway block : it used for testing the value of a Boolean equation. This type of block contains just the expression to be tested and returns a Boolean value that indicates whether the expression is true or false, which determines the next block of the agent. Usually, this block is followed by 2 blocks, the first block is active when the equation is true and the second is active when it is false.

5. A block with many successors : this type of block is used to represent the position of an agent that executes more than one block after the execution of a block. In this case, once the execution of the block is completed, all successor blocks are activated .
6. A block with many predecessors : this type of block is used to represent the position of a block that is active only when several blocks are already executed.

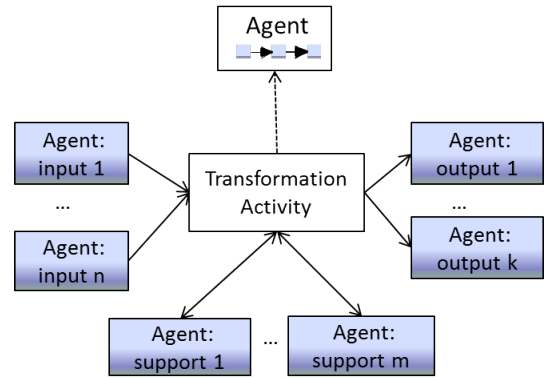


FIGURE 3.5: Transformation Activity

When a behavioral mode for an agent becomes active, we execute its first block (usually the start event). We then identify its successor block and check if it is an activity block. If not, we propagate the activation. If the next block is of type gateway we test its expression to determine its successors. When a block has an activity block as successors, we test all input and support elements (agents) of this activity block. If they are all available, this activity block becomes active running :

- we execute its input actions and make time $t =$ current time ;
- we execute its task when $t +$ duration of block is less then current time ;
- Once, current time = $t +$ duration of block, we execute the output actions of the block, we destroy its input, we release its support and output agents, and this block becomes inactive.

A behavioral mode of an agent is finished when we execute its final block (end event).

3.3.2 Relations between dysfunctional and behavioral views

During the construction of our multi-agent model :

1. We define the set of our agents and a behavioral mode for each agent called default behavioral mode ;
2. For all agents, we define the set of failure modes associated for each activity in the default mode ;
3. For each failure mode, we define a corresponding behavioral mode (see Figure 3.7) (for example : if failure mode FA_{11} is active we activate behavioral mode Mode 2). Sometimes many failure modes can have the same behavioral mode with

some interior changes in the behavior of the activity. A behavioral mode can be defined by a combination of failure modes (if FA_{11} and FA_{12} are active we activate behavioral mode Mode 4).

Now, for each agent in the environment, we have :

- a set of variables ;
- a set of behavioral modes : with one representing the normal mode (defined by default) and the others representing the dysfunctional modes ;
- a set of dysfunctional modes : which contains the set of failure modes of all activities agent ;
- a set of relations between the functional and behavioral mode ;

It should be noted that a behavioral mode for an agent is normal, if all its failure modes are false.

The number of behavioral modes of an agent may be up to 2^n where n : is the number of its failure modes. The number is, however, usually much less as a same behavior mode is associated to several failure modes.

3.3.3 Transition rule from a mode to another

At each step in the simulation, a test is carried out on the entire agent failure modes. This allows us to identify which failure modes should be true in the next step. Based on the value of each failure mode, we can determine if the current behavioral mode remains active for the next step or replace it by another mode. If it is to be replaced, we must stop all running activities that correspond to the old mode and activate the new mode. This can be established by :

- stopping the current activity which is the old mode (by putting it in inactive state, destroying its inputs elements and releasing its support and outputs elements) ;
- activate the first block in the novel mode.

Figure (3.6) shows an activity of an agent A, which has two failures modes FA_{11} and FA_{12} . Each failure mode is represented by a fault tree as shown in (figure 3.6). For each failure, we define a corresponding behavioral mode (see Figure 3.7). For each of the values FA_{11} and FA_{12} , we obtain a behavioral mode of the agent (if FA_{11} and FA_{12} are false, the behavioral mode of the agent is Mode 1, else if FA_{11} is true, the behavioral mode of the agent is Mode 2,...).

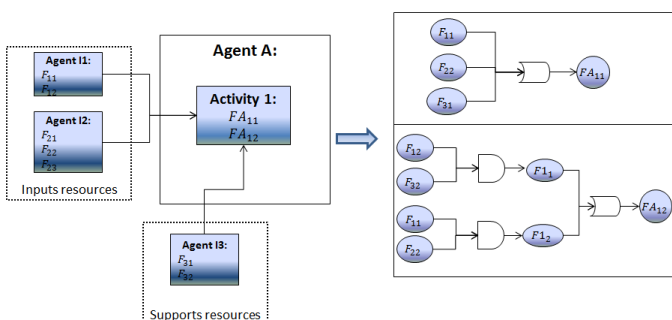


FIGURE 3.6: Failure modes for an activity

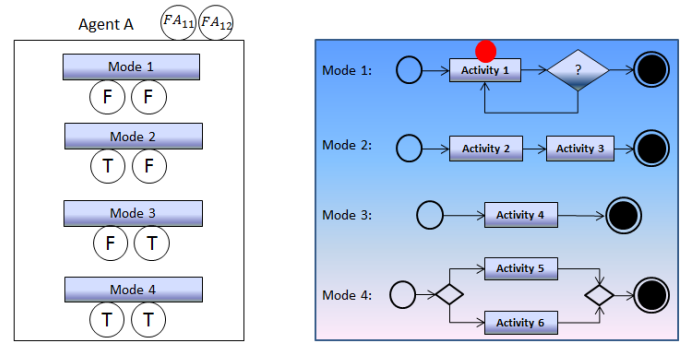


FIGURE 3.7: Relation between functional and behavioral view

4 EXAMPLE OF APPLICATION

In this section, we apply the proposed approach on a hazardous material road transportation system. We use the multi-agents System to simulate our system under Repast [Collier, 2003]. We lay out the *moving* activity. We construct the structural, the dysfunctional and the evolution views. This activity presents the advancement of the truck agent along a predefined path. To achieve such advancement, many entry agents are required such as, the truck loaded with hazardous materials in position 1, the driver, the road segment (to advance the truck), and the weather provider, the traffic density provider (to calculate the probability of a failure). The needed output agents is the truck in position 2. The truck is seen as modified resources. *Moving* will change the truck position after advancement. The driver, the road segment, the provided weather, and traffic density are seen as supports for the *Moving* function. They are reserved prior to *Moving* and released at the end. Figure 4.1 shows the relations of *Moving* describing in the above.

To secure the advancement, we should take note about the quantity and the quality of loaded products, and eliminate any loss or accident that may occur during the advancement (e.g., accident between two vehicles on the road or accident between the truck and a fixed object on the road). A *Moving* failure may come from two cases [Flaus and Granddamas, 2002][Wirth et al., 1996] :

- loss of containment during the truck advancement caused by a structural failure or a containment bypass ;
- accident during moving (related to weather conditions, road characteristics and moving conditions along the advancement) ;

Figure 4.2 shows the failure mode loss of containment of a moving activity described in a fault tree and figure 4.4 shows the failure mode accident for the activity described in a Fault tree [Karagiannis et al., 2010]. Figure 4.3 presents the set of behavioral modes of truck agent. Table 4.1 represents the FMEA of the Moving activity and table 4.2 illustrates the behavioral mode related to each failure mode. Initially, we operate in the normal mode of the truck agent.

TABLE 4.1: FMEA of Moving activities

Agent or element	failure mode	cause	effect	action
Agent driver	LOC truck	Container containing water Wrong closure of the container.	evacuation toxic inhalation	calculate the number of impacted people
Agent truck	LOC truck	Cooling malfunction Failure of the heating system	evacuation toxic inhalation	calculate the number of impacted people
Agent segment	accident (related road)	Type of segment area, Roadways characteristics.	death injuries	calculate the number of impacted people
Agent provider traffic density	accident (related to traffic density)	Congestion in the road.	death injuries	calculate the number of impacted people
Agent provider weather	accident (related weather)	Bad weather conditions during the advancement.	death injuries	calculate the number of impacted people

TABLE 4.2: Failure modes of Moving activity and the related behavioral modes of the truck agent

failure mode	related behavioral mode	equations	severity calculation
No failure	normal mode	$p_{k1}(x, y) = p_k(x, y) + (v \times t_e)$ $p_{k1}(x, y)$: position of the truck at step k+1, $p_k(x, y)$: position of the truck at step k, v : average speed of the truck, t_e : sampling time, $q_{k1} = q_k$. q_k : quantity of Materials in the truck at step k, q_{k1} : quantity of Materials in the truck at step k+1.	
LOC truck (f1)	Mode 2	$p_{k1}(x, y) = p_k(x, y)$, $q_{k1} = q_k - (t_d \times t_e)$ if advancement with LOC $p_{k1}(x, y) = p_k(x, y) + (v \times t_e)$ $q_{k1} = q_k - (t_d \times t_e)$	$Gr = f(p_{k1}(x, y), q_{k1}, c_d, d_p, t_d, cm, t_e)$ Gr : severity of the failure c_d : code of danger of Materials in the truck, d_p : population density in the area of the truck, cm : weather conditions in the area of the truck, t_d : leakage rates.
accident (f2)	Mode 3	$p_{k1}(x, y) = p_k(x, y)$, if we have thermal or explosive effect (case 1) : $q_{k1} = 0$ if toxic effect (case 2) : $q_{k1} = q_k - (t_d \times t_e)$	if thermal or explosive effect : $Gr = f(p_{k1}(x, y), q_{k1}, c_d, d_p, t_e)$ if toxic effect : $Gr = f(p_{k1}(x, y), q_{k1}, c_d, d_p, t_d, cm, t_e)$
LOC followed by accident (f3)	Mode 4	$p_{k1}(x, y) = p_k(x, y)$, case 1 $q_{k1} = 0$ case 2 : $q_{k1} = q_k - (t_d \times t_e)$	if thermal or explosive effect : $Gr = f(p_{k1}(x, y), q_{k1}, c_d, d_p, t_e)$ if toxic effect : $Gr = f(p_{k1}(x, y), q_{k1}, c_d, d_p, t_d, cm, t_e)$

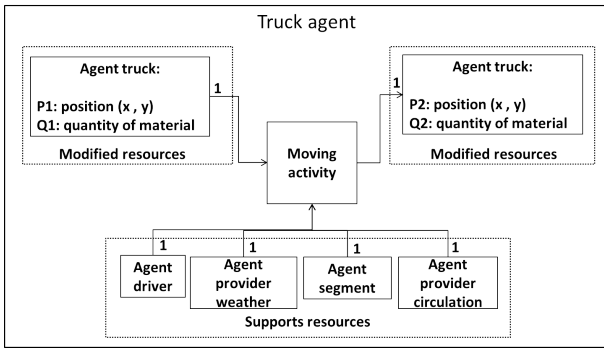


FIGURE 4.1: Relations of the moving activity of the truck agent

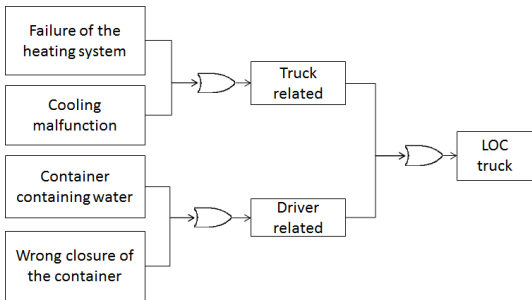


FIGURE 4.2: Fault tree loss of containment (LOC truck)

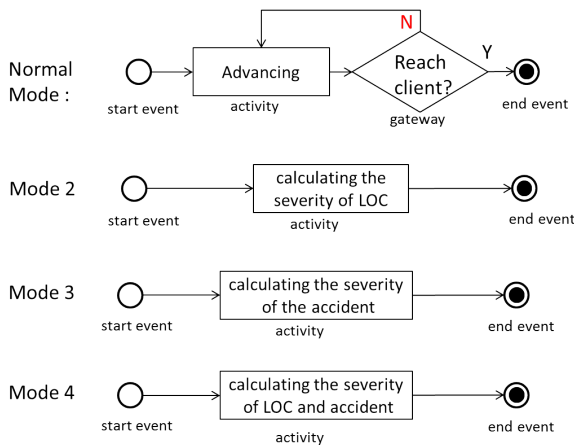


FIGURE 4.3: Behavioral modes of the truck agent

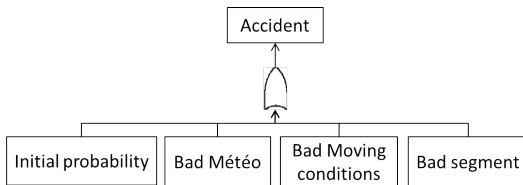


FIGURE 4.4: Fault tree accident

5 CONCLUSION

In this paper, we have presented a new approach for risk analysis based on multi-agents. For each agent in the system, we define its activities, attributes, a list of failure modes and a list of behavioral modes. The main interest of this meta model is that it allows the representation of risk analysis and dynamical behavior in a coherent manner, which can be used to simulate the behavior of a system in normal or degraded conditions ...This was illustrated throughout an example of a moving of a truck agent loaded with hazardous materials. One of our future goals is to apply our meta model to

a system containing a large number of active agents.

RÉFÉRENCES

- [Collier, 2003] Collier, N. (2003). Repast : An extensible framework for agent simulation. *The University of Chicagos Social Science Research*, 36 :2003.
- [Ferber and Perrot, 1995] Ferber, J. and Perrot, J.-F. (1995). *Les systèmes multi-agents : vers une intelligence collective*. InterEditions.
- [Flaus and Granddamos, 2002] Flaus, J. and Granddamos, O. (2002). Towards a formalisation of mads, system failure analysis model, in lambda-mu 13. ESREL.
- [Flaus, 2013a] Flaus, J.-M. (2013a). *Risk Analysis : Socio-technical and Industrial Systems*. John Wiley & Sons.
- [Flaus, 2013b] Flaus, J.-M. (2013b). *Risk Analysis : Socio-technical and Industrial Systems*.
- [Giap et al., 2009] Giap, Q. H., Ploix, S., and Flaus, J.-M. (2009). Managing diagnosis processes with iterative decompositions. 296 :407–415.
- [Gordon, 1961] Gordon, G. (1961). A general purpose systems simulation program. In *Proceedings of the December 12-14, 1961, eastern joint computer conference : computers-key to total systems control*, pages 87–104. ACM.
- [Kanj and Flaus, 2014] Kanj, H. and Flaus, J. (2014). A meta model framework for risk analysis, diagnosis and simulation. In *Safety and Reliability*, pages 2043–2049. CRC Press.
- [Karagiannis et al., 2010] Karagiannis, G., Piatyszek, E., and Flaus, J. (2010). Industrial emergency planning modeling : A first step towards a robustness analysis. *Journal of Hazardous Materials*, 181 :324–334.
- [Papadopoulos et al., 2004] Papadopoulos, Y., Parker, D., and Grante, C. (2004). A method and tool support for model-based semi-automated failure modes and effects analysis (fmea) of engineering designs. *Research and Practice in Information Technology, ACM, ISSN 1445-1336*, 47 :89–95.
- [Wirth et al., 1996] Wirth, R., Berthold, B., Krämer, A., and Peter, G. (1996). Knowledge-based support of system analysis for the analysis of failure modes and effects. *Engineering Applications of Artificial Intelligence*, 9(3) :219–229.
- [Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents : Theory and practice. *The knowledge engineering review*, 10(02) :115–152.