International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:13, No:10, 2019

Dependability Tools in Multi-Agent Support for Failures Analysis of Computer Networks

Myriam Noureddine

***Abstract*—**During their activity, all systems must be operational without failures and in this context, the dependability concept is essential avoiding disruption of their function. As computer networks are systems with the same requirements of dependability, this article deals with an analysis of failures for a computer network. The proposed approach integrates specific tools of the plat-form KB3, usually applied in dependability studies of industrial systems. The methodology is supported by a multi-agent system formed by six agents grouped in three meta agents, dealing with two levels. The first level concerns a modeling step through a conceptual agent and a generating agent. The conceptual agent is dedicated to the building of the knowledge base from the system specifications written in the FIGARO language. The generating agent allows producing automatically both the structural model and a dependability model of the system. The second level, the simulation, shows the effects of the failures of the system through a simulation agent. The approach validation is obtained by its application on a specific computer network, giving an analysis of failures through their effects for the considered network.

***Keywords*—**Computer network, dependability, KB3 plat-form, multi-agent system, failure.

I. Introduction

CURRENTLY, to be competitive all systems must be operational during their activity and to avoid an interruption of their function, it is necessary to identify potential failures. Failure is defined as the alteration or cessation of a required function [1], and dependability is considered as the science of failures [1], [2]. So, one definition of dependability [3] is the ability to avoid service failures.

The dependability concept integrates many attributes [1]-[4] as the reliability to assure the continuity of correct service, the maintainability to able the undergoing reparations, the availability to deliver correct service under the aspect of reliability and maintainability, and the safety to assure protection against natural and industrial disasters. The dependability attributes describe the expected properties of a system and computer networks are systems with the same dependability requirements.

This article deals with an analysis of failures for a computer network and the proposed approach is based on a methodology supported by a multi-agent system, giving a modeling and simulation framework. The methodology integrates specific tools issued from the KB3 plat-form [5], which are usually applied in dependability studies of industrial systems. To show that these tools can also be applied to computer networks, the approach is validated on a computer case study. For this system, first a knowledge base is given, followed by the structural model and dependability models. Analysis of failures is obtained by an interactive simulation, providing the effects of breakdowns on the network operation.

M. Noureddine is with the Department of Computer Science, University of Sciences and Technology of Oran, Algeria (e-mail: myriam.noureddine@univ-usto.dz).

The paper is organized as follows. Section II presents methods, languages and tools in a dependability context. In Section III, the adopted approach is presented with description of the multi-agent system and its application on a network example, giving the knowledge base and the two models. Section IV presents the interactive simulation on the network, by a failure assignment on one or many components. Results are discussed around the impact of failures on the network functioning.

II. Literature Review of Dependability Techniques

A. Dependability Methods

Many methods [2], [4], [6] are available to perform dependability analysis, such as Preliminary Hazard Analysis (PHA), Failure Mode and Effect Analysis (FMEA) and Event Tree Analysis (ETA). These bottom-up approaches are inductive, from causes of failures to consequences. Conversely, the top-down methods are looking for the causes of an undesired event. These deductive approaches are well suited to the analysis of disasters and in this reasoning mode the main method is the Fault Tree. The deductive Fault Tree method [7], [8] is a technique which consists of building a tree where the root is an undesired event for the system and the leaves are primary events as basic events with no further development. The internal nodes are intermediate events, which represent faults can cause failures. Events are joined by logical gates and all elements are represented by specific symbols, giving a fault tree of the considered system.

B. Languages and Tools

To optimize errors during the generation of dependability models, languages associated with tools have been developed. In the languages dedicated to dependability, FIGARO [9] is widely used in dependability studies of industrial systems. Tools associated to this language are among the most powerful tools for safety analysis. This language is born from the unification need between various formalisms leading to conventional models. Indeed, to automatically deduce the necessary data for generate classical models (Fault trees, Petri nets, Markov chains, etc.) the FIGARO language has been developed to unify the entries of these models [9]. FIGARO uses the object oriented approach and provides a graphical model without a priori on the modeled system. The language follows specific syntax and semantics and has two equivalent

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:13, No:10, 2019

orders [9]. The order 1 allows the declaration of concepts, made up the knowledge base types and the order 0 is obtained by instantiation and heritage mechanisms on the first order rules.

To conduct fast and quality dependability studies, EDF (Electricite De France) developed the KB3 program [5] using knowledge base written in FIGARO. The simulation of the system behavior and the generation of classical models (as fault tree) in dependability are obtained through the set of tools grouped in the KB3 plat-form.

III. Proposed Approach

A. Methodology Framework

Multi-agent systems [10] provide a framework for modeling and simulation of many systems. The aim is to represent their elements, behaviors and interactions in the form of computing entities with their own autonomy. These entities or agents [11] can be physical or virtual, acting and perceiving environment and exchanging communications. Various problems used Multi-Agent Systems (MAS) and it turns out to be a good solution for distributed control [11]. Among the wide variety of applications [12] of MAS, telecommunications and network management are important issues. In this context, the proposed methodology (Fig. 1) to analyze the computer failures is supported by a MAS. Steps are carried out according to the two levels of modeling and simulation.

By integrating the KB3 tools, the whole process provides a dependability plat-form to a user for investigate failures of computer networks. Thereby, this will show that languages and tools usually applied in dependability studies of industrial systems can also suit in dependability of computer networks.

The proposed MAS is formed by six agents grouped in three meta-agents:

- The conceptual-agent is dedicated to the building of the knowledge base from the system specifications written in the FIGARO language. The conceptual-agent contains a Knowledge Base (KB) agent to define elements and a parameter agent to assign graphical and textual parameters for each element. The final KB with affected parameters is obtained through interactions between these two agents.
- The generating-agent allows producing automatically both the complete model of the system by the topology agent and a dependability model of the system by the interface agent. The generating-agent communicates with the conceptual-agent to validate the obtained models.
- The simulation-agent deals with interactive simulation through two agents: the failure agent affects a failure to one element of the system and the analysis agent give results of this assignment. Impact on models can be view following communication between the generating and simulation agents.

The approach is applied to a computer network, localized in an industrial company and formed of PCs and servers interconnected. The network architecture follows the star topology [13] where all the elements are connected via a switch, which is the active central node [14].

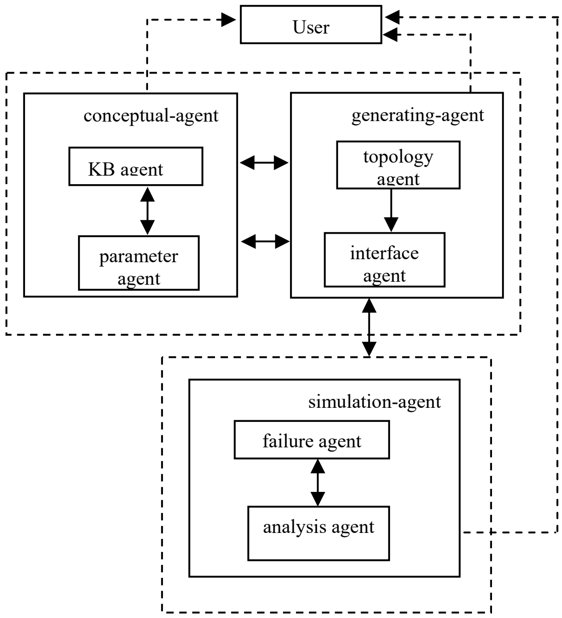

Fig. 1 Methodology Approach

There are four servers with their own service: a domain server which is responsible of the network, a security server for the management of antivirus and new applications, a database server for the data update and a communication server to control connections with the outside. In this distributed network, there is a set of PCs for the users (clients) of the production units and a set of cables as the transmission mean.

B. Modeling Phase of the Network

1) The KB

The adopted methodology is realized through the modeling plat-form KB3 using the open-source Visual Figaro [15]. This tool is the support of the conceptual agent to build the KB, through the static (declarative) and the dynamic part [5].

The static part defines the two types 'node' and 'link' which allow the specification of elements belonging to the KB. The elements server and client are defined as node as well as the switch and the element bi_dir edge is defined as a link for connection of elements in the network:

```
TYPE node;
TYPE server KIND_OF node;
TYPE client  KIND_OF node;
TYPE switch KIND_OF node;
TYPE link;
TYPE bi_dir edge KIND_OF link;
```

Then, one must introduce the interface concept for linking elements of the system. For example, the interface for the element bi_dir edge allows the connection between two nodes:

```
INTERFACE
end KIND node CARDINAL 2;
```

In the dynamic part, rules describe the object behavior through occurrence and interaction rules [5]. In the dependability context, one occurrence rule is declared for all elements, showing the change state, from the state "WORKING" to the state "FAILURE". The associate probability distribution is the exponential distribution, defined by the single parameter, the failure rate λ [16]. The occurrence rule is:

MAY_OCCUR FAILURE fail DISTR EXP (lambda);

For each element and to propagate the event effects, interaction rules with the operating conditions are declared. For example, the interaction rule for the node switch is:

IF WORKING AND function = 'switch' THEN link;

2) Generation of models

The parameter agent steps in the assignment of graphical and textual parameters for each elements of the previously specified computer network in the KB. These assignments are:

- a yellow circle for the four servers: server of domain (ser_sd), server of security (ser_sec), server of database (ser_BDD) and server of communication (ser_com)
- a blue circle for the switch (sw)
- a specific shape for the client (clt). There are two clients in the network, clt1 and clt2
- a simple line for the link (l)

The implementation of the generating agent consists in activation of the topology agent, giving automatically the structural model (Fig. 2) of the computer network.

The generating agent activates the second agent, the interface agent, allowing an automatic generation of dependability model like the fault tree. So, from both the physical model and the KB, the order 0 FIGARO model is done automatically to obtain an interface required for the fault tree generation and for the interactive simulation.

The instantiation on the first model obtained previously gives the following objects:

OBJECT ser_sd IS_A server;
OBJECT ser_sec IS_A server;
OBJECT ser_BDD IS_A server;
OBJECT ser_com IS_A server;
OBJECT sw IS_A switch;
OBJECT clt1 IS_A client;
OBJECT clt2 IS_A client;
OBJECT l_sd IS_A bi_dir edge;
OBJECT l_sec IS_A bi_dir edge;
OBJECT l_BDD IS_A bi_dir edge;
OBJECT l_com IS_A bi_dir edge;
OBJECT l_cl1 IS_A bi_dir edge;
OBJECT l_cl2 IS_A bi_dir edge;

The object definition is completed by characteristics derived from the specification of order 1. For example, the interface of the object l_cl1 becomes:

INTERFACE end = clt1 switch;

The obtained complete specification is used to generate the fault tree (Fig. 3) of the network. The steps consist of the definition of the events leading to failures, and all events are joined by logical gates.

The undesired event is the 'breakdown of the system' (EI), produced by events on one of the system components. These events are connected by a gate OR. For the switch, the basic event is the deficiency ('def') and there is no decomposition. However, the fault-tree development is made for the two others elements. Deficiencies on servers are the basic events, connected by a gate AND with the event ES "breakdown of the servers". In the same, the gate AND connects basic events on links, the cut-links ('cut'), with the event "breakdown on the links" (EL).

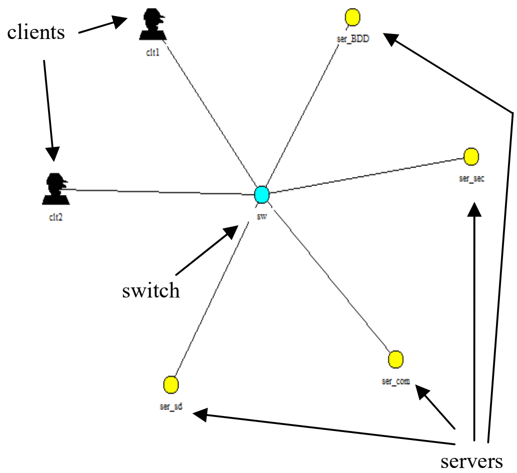

Fig. 2 Structural model of the network

IV. Analysis of Network Failures

A. Analysis by Simulation

The meta-agent simulation is implemented following two agents: the failure agent drives the failure effect of an element on the computer network, through failures on servers, switch and links. The displaying is obtained by the activation of a selected transition among the type of failures (transitions) relating to the thirteen objects. These transitions are stocked in a transition base (Table I) and imported by the simulation process for analysis of failures.

TABLE I
DATABASE OF TRANSITIONS

Object	Transition	Object	Transition
clt1	def	l_sec	cut
clt2	def	ser_BDD	def
l_ clt1	cut	ser_com	def
l_ clt2	cut	ser_sd	def
l_bdd	cut	ser_sec	def
l_com	cut	sw	def
l_sd	cut		

The second agent, the analysis agent, allows the description of failure effects on the network. The simulation process (red icons) provides two cases from the effects on the network through the analysis agent, following impact or no impact of failures.

1) Failures with no Impact

The first case considers no effect on the system. It appears if the failure is on one element (server, link) and only the concerned element is disconnected. For example, Fig. 4

visualizes the failure on the security server (object ser_sec), after the failure activation ('def') on this server.

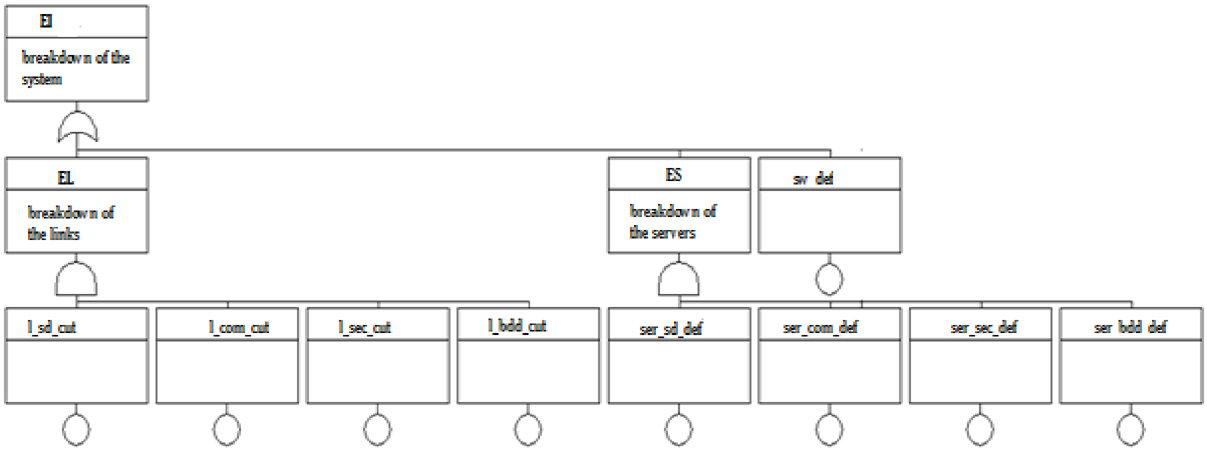

Fig. 3 Fault tree of the network

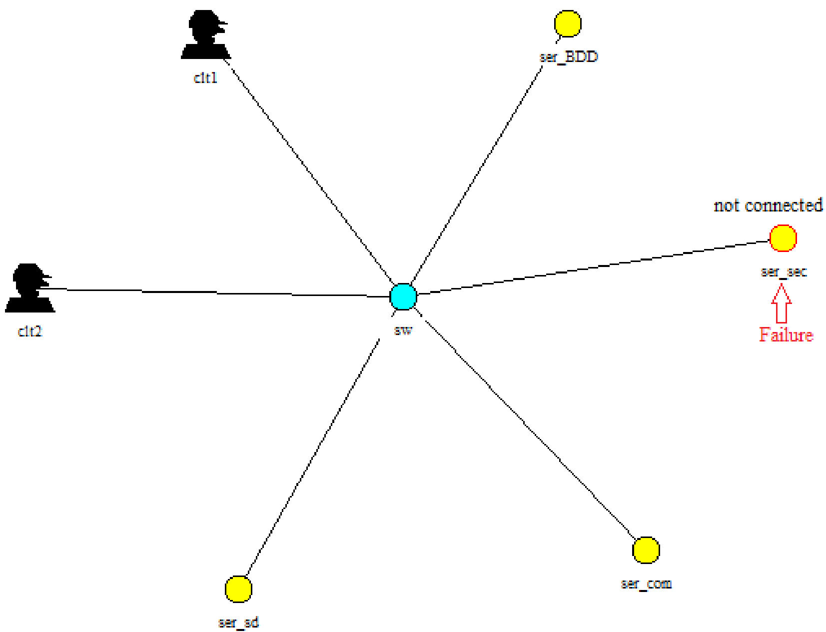

Fig. 4 Failures with no impact

2) Failures with Impact

The second case considers an effect on the system. It appears after the transition activation ('def') on the switch (object sw). In this case, all others elements are disconnected (Fig. 5) and the computer network is failed.

B. Validation of Results by Fault Tree Analysis

The obtained results are validated by the analysis of the fault tree: The first case is illustrated by the semantic of the logical connector AND: all basic failures on servers (respectively on links) are linked by a gate AND, so a failure on one server (link) has no impact to the network functioning. The breakdown of the servers ES (links EL) is obtained if all servers (links) are failed. The second case gives the undesired event (breakdowns of the system) induced by the failure on the switch; this situation is visualized in the fault tree as the top event is connected by a gate OR with the basic event on the switch. To explicit these results and to highlight the dependencies between elements of the network, an extended version of fault tree is used: the Boolean Logic Driven Markov Processes (BDMP) [17]. This formalism is well suited for dependability studies like for example, reliability evaluation in multicast applications [18] and modeling safety and security [19].

A BDMP is a fault-tree augmented by a set of driven Markov processes which is associated with its basic events (leaves) evaluated in two states (on and off) and a set of "triggers" to model all kinds of dependencies between the system components.

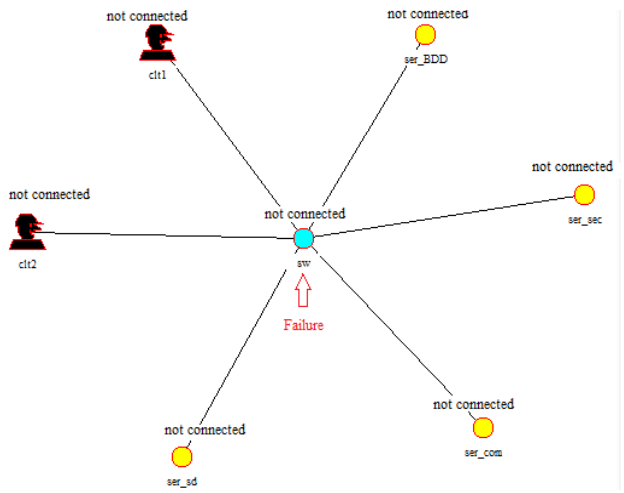

Fig. 5 Impact of failures

The automatically generated BDMP (Fig. 6) related to the fault tree is formed of the nine leaves associated to the four servers with their links and the switch. In the fault tree, links and servers are grouped separately, dealing with failures on servers or links exclusively. In fact, there is an inter dependency between a server and its link, as failure on one link or one server gives no communication. This relation is modeled in the BDMP with a gate (OR) and all servers with their respectively links are joined by a gate (AND).

The simulation agent implementation allows an interactive simulation of the BDMP, following the previous two cases.

We consider a failure simulation on one element: this case gives an impact only on the concerned failure. For example, a failure on one server gives a breakdown on this element; other servers are running. The trigger is oriented from components to the switch. So, there is no failure on the switch and the system is operational.

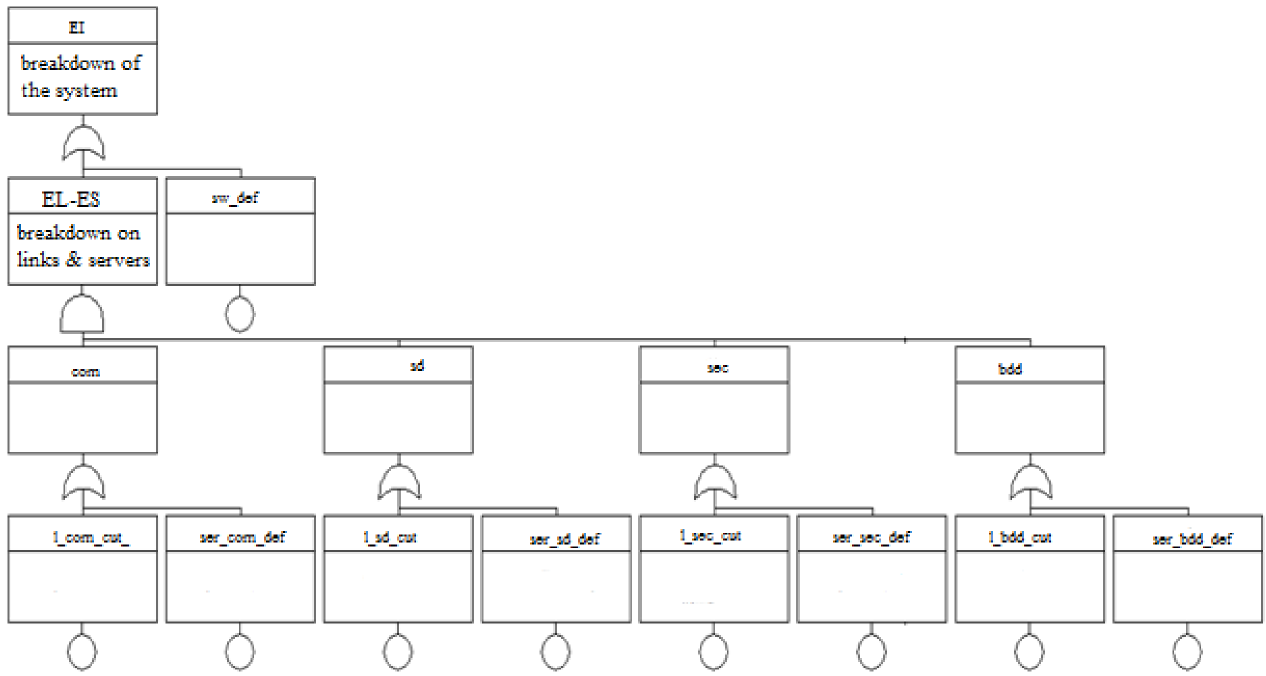

Fig. 6 BDMP of the network

Now consider an assigned failure to the switch. Fig. 7 visualizes the simulation of the obtained BDMP in the plat-form KB3. The trigger (dotted arrow) is oriented from this element to the gate joined all other components. Following the logic connector OR between this gate and the switch, the system is failed due the generation of the undesired event EI 'breakdown of the system'.

According to the BDMP model, a breakdown of the computer network can be caused by failures of all elements showing on the figure through the non-pertinent elements ('non pert'): all failing servers or all failing links provide also a breakdown of the system.

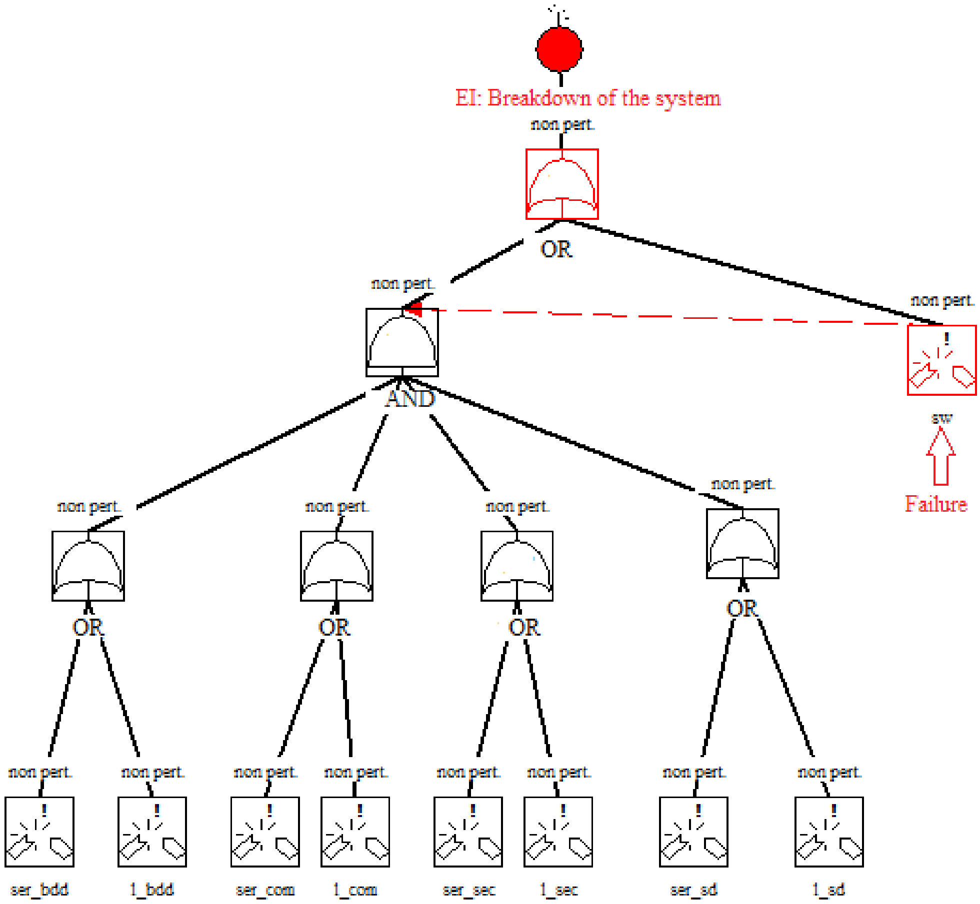

Fig. 7 Simulation of the system breakdown

V. Conclusion

In this article, we presented an analysis of failures for a computer network. The methodology is in the dependability context and the adopted approach is formed of modeling and simulation levels.

A MAS supports the methodology through these two levels with agents grouped in three meta agents. The first level gives models of the system by the implementation both of the conceptual agent and the generating agent. The second level proposes an interactive simulation for an analysis of failures through their effects on the system; this process is obtained by the simulation agent implementation. The assistance of the multi-agent approach provides a generic framework for modeling and simulation and giving a dependability platform for investigating failures of computer networks. The FIGARO formal language is used to define the KB with specific tools of the platform KB3.

The approach has been applied to a computer network, giving the KB of the system associated to structural and dependable models. By a dynamic analysis of these models, effects of failures are studied. The obtained results lead to a hierarchy of failures from the catastrophic failure towards the minor failure. The failure on the switch has the highest severity because all other elements of the network are disconnected. So, the star network requires a greater protection of its central node. Failure on link or on server implies an effect only on the concerned element. A numeric evaluation of the severity criteria is underway to accurately classify failures on the network components. On the other hand, the obtained results showed that the KB3 platform can also suit in dependability of computer networks and in this work, the adopted network is based on a star architecture. Knowing that the methodology is generic, as a future work, an application of the approach on a more sophisticated computer system is envisaged.

References

[1] G. Zwingelstein, "*Sûreté de fonctionnement- Principaux concepts*," Techniques de l'ingénieur, S8250 v3, 2019.

[2] A.Villemeur, *Reliability, Availability, Maintainability and Safety Assessment*. Wiley-Blackwell, 1991, vol. 1.

[3] A. Avižienis, J.C. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, 1(1), 2004, pp. 1–23

[4] R. K. Kaur, B. Pandey and L.K. Singh, "Dependability analysis of safety critical systems: Issues and challenges," *Annals of Nuclear Energy*, 120, 2018, pp. 127–154.

[5] M. Bouissou, S. Humbert, S. Muffat and N. Villate, "KB3 tool: feedback on knowledge bases," in *Proc. of European Safety & Reliability Conference,* Lyon, France, (18-21 March 2002), pp. 114–119.

[6] S. Bernardi, J. Merseguer and D.C. Petriu, M*odel-Driven Dependability*

Assessment of Software Systems. Springer, 2013.
[7] W.E. Veeley, F.F Goldberg, N.H. Roberts and D.F Haasl, *Fault Tree Handbook*, NUREG 0492, US Nuclear Regulatory Commission, Washington, USA, 1981.
[8] N. Limnios, *Fault Trees*. Wiley & Sons, 2013.
[9] M. Bouissou, H. Bouhadana, M. Bannelier and N. Villatte, "Knowledge modelling and reliability processing: presentation of the FIGARO language and associated tools," in *Proc. of Safety of Computer control system IFAC Symposium*, Trondheim, Norway, (30 October-1 November 1991), pp. 69–75.
[10] J. Ferber, *Multi-agent systems: An introduction to distributed artificial intelligence*. Addison-Wesley Professional, 1999.
[11] J. Xie and C.C. Liu, "Multi-agent systems and their applications," *Journal of International Council on Electrical Engineering*, 7(1), 2017, pp. 188–197.
[12] S.S. Manvi and P. Venkataram, "Applications of agent technology in communications: a review," *Computer Communications*, 27, 2004, pp. 1493–1508,
[13] O. Bonaventure, *Computer Networking: Principles, Protocols and Practice*, Saylor Foundation, 2011.
[14] N. Bisht and S. Singh, "Analytical study of different network topologies," *International Research Journal of Engineering and Technology*, 02 (01), 2015, pp. 88–90.
[15] G. Torrente and M. Bouissou, "Building knowledge bases in the dependability field with the open-source environment "VISUAL FIGARO"," in *Proc. 16th Congress Lambda-Mu16,* Avignon, France, (7-9 October 2008).
[16] J. Marshall, *An Introduction to Reliability and Life Distributions*, Product Excellence using 6 Sigma Module, Reliability and Life distributions, University of Warwick, 2012.
[17] M. Bouissou and J.L. Bon,"A new formalism that combines advantages of fault-trees and Markov models: Boolean Logic Driven Markov Processes," *Reliability Engineering and System Safety*, 82 (2), 2003, pp. 149–163.
[18] M. Noureddine and S. Benhabib, "Performance et fiabilité des systèmes de communication multipoints", in *Proc. 10ème Congrès International de Génie Industriel,* La Rochelle, France, (12-14 Juin 2013).
[19] S. Kriaa, M. Bouissou, F. Colin, Y. Halgand and L. Pietre-Cambacedes, "Safety and Security Interactions Modeling Using the BDMP Formalism: Case Study of a Pipeline," in *Proc. 33rd International Conference on Computer Safety, Reliability, and Security,* Florence, Italy, (10-12 September 2014), LNCS 8666 Springer, pp. 326–341.