Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 4 (2016) pp. 116-125

Interdependencies Modeling for the purpose of critical infrastructures protection

Lamiae CHTIOUI *, Amine BAINA and Mostafa BELLAFKIH

STRS Lab. National Institute of Posts and Telecommunications- INPT
Rabat – Morocco

{chtioui, baina, bellafkih}@inpt.ac.ma

***Abstract*: Critical Infrastructures are the systems that ensure the ease of use of crucial goods and services which are vital for the well-being of a society and the prosperity of an economy. However, the protection of these infrastructures is a difficult challenge, and must include many factors such as risks, resilience, control access, and interdependency. This paper focuses on Critical Infrastructures interdependencies. Thus networks are becoming tightly interdependent, especially for communication and computing systems that are firmly coupled with other networks, because of their current growing use and necessity for all sectors. Our objective is to study interdependencies in order to reduce their impacts and establish an effective procedure to prepare for future risks. This paper presents an overview on some modeling approaches and models used to analyze critical infrastructures interdependencies. Then it explains the idea of our future work that consists of conceiving application software to notify all nodes of interdependent networks of a failure emerging in a critical node, basing on the Implicative Interdependency Model.**

Key words: Critical infrastructures, Interdependencies modeling, Critical infrastructures protection, Implicative Interdependency Model, CRUTIAL System.

I. Introduction

The term critical infrastructure [1] refers to the set of systems that ensure the availability of crucial goods and services, which means systems that are fundamental for the functioning of a society or an economy such as energy, telecommunication and transport....etc, whose failure could have severe consequences on the economy and population and may undermine national or even international security [1]. Indeed, the protection of critical infrastructures is a requirement and must be improved to reduce risks due to different causes, in particular interdependencies[2], which must be taken into account during the implementation of protection procedures. Interdependencies stem from logical and functional relationships between components of distributed systems [3]. Therefore, a failure of a component in an infrastructure may impact the performance of another interconnected infrastructure.

Interdependency is defined by Rinaldi, Peerenboom, and Kelly[1] as a bidirectional relationship between two infrastructures through which the state of each one influences or is correlated to the state of the other. This implies that it is a mutual dependence. For example, the communication network controls the power grid, if the first one goes through an attack, the power network will not provide right operations or even fails; the same for the power network, a lack of energy will leads to the failure of communication system.

Interdependencies between critical infrastructures [2] can trigger further failures to propagate from one to another, through a cascading process, aggravating and prolonging the impact between networks. Modeling Interdependencies [4] allows extracting vulnerabilities related to them, once they are identified, a better analysis and vulnerability assessment of critical infrastructures is achieved.

To estimate the impact of a contingency affecting an infrastructure on the operation of another interconnected one, and to accomplish a better analysis and vulnerability assessment of critical infrastructures, considering interdependencies, relationships between components of concerned infrastructures must be modeled, which is difficult because of the dynamism, complexity and interconnections of critical infrastructures.

The questions we might ask are how to model interdependent critical infrastructures? and how to deal with these interdependencies? Modeling interdependent infrastructures is an important methodology which aims to define the basic structure of the system, and its ability to resist to failures when strong dependencies exist. It also provides a rigorous analysis of interdependency in order to create a complete decision basis to identify and reduce risks and ensure effective security. The problem that remains is about existing methodologies, are they applicable to real world situations? And what are their limitations?

To protect critical infrastructures, it is necessary to study the systems behavior and the processes of interaction among their components, when they are stressed or attacked. Therefore, a challenging issue consists in providing formalisms, methodologies, and tools to model the entire complex system composed of critical infrastructures and also human interaction. To understand the problem with interdependencies in critical infrastructures we:

- Have studied and compared between most used approaches in modeling critical infrastructures interdependencies.
- Have modeled them using UML language to understand and analyze their operation and functioning, in order to extract their strengths and try to include them in our

proposed approach.

- Conceive an approach that will deal with the phenomena of stress, attack or fail in general in the components of critical infrastructures.
- Give an overview on how to develop our platform on which the information system runs.

Our perspectives are to resolve problems related to interdependencies, and be capable to secure components to remedy the impacts of interdependencies, and to react to a node failure to prevent it to spread over networks and reach the other nodes that are interconnected with. We consider that infrastructures are a set of nodes related with links. Because of interdependencies impacts, and due to cascading effects, if one node is down, all nodes that are interconnected with it or receive services from, are going to be inoperable. Same thing for links, if a link is down, nodes that are connected through would not be able to communicate with each other. Our future work is to develop an application software to notify nodes of network of a failure emerged in most vulnerable nodes, or the nodes that need to be most protected. Identifying these elements will allow us to optimize resources needed for protection. In this paper we give an overview of the conception of our approach and explain how it should operate.

II. Related Work

In this section we present different methods that have been developed to model critical infrastructures interdependencies in the intention of their protection. Starting with Agent based Model (ABM) [5], often used to simulate interdependencies, is a bottom-up [6] approach that consists of modeling a complex system into a set of agents interconnected with one another, each agent represents a physical entity characterized by a location(geographic location), capacity (performances and behaviors), and memory (overuse, stress, aging…)[7]. Agents embeds the characteristics of a system entity into their behavior, making the system model representing real functioning of the modeled system [5].

An interesting method representing a top-down approach [6] was presented in [8], called System dynamics, is used to study and understand the behavior and the underlying structure of a complex system over time[9]. System Dynamics method deals with internal feedback processes (loops) and time delays that influence the whole system.

A methodology for the assessment of infrastructure interdependencies has been discussed by Eusgeld, Nan, and Dietz in [10]. This is High Level Architecture (HLA), it is a general architecture for modeling and simulating complex distributed systems. This technique breaks the entire system down into individually operating subsystems. Communications within them is managed by a run time infrastructure (RTI) [10] which ensures the distribution of information whenever it is updated [11].

Finally, hybrid system, this term is related to mathematical methodologies for the modeling and simulation of complex computational systems. The primary goal of hybrid system architecture is to facilitate the simulation of interdependent systems, and to benefit of each approach by integrating different types of modeling approaches in a single simulation platform[12].

Each one of the modeling methods already mentioned has its pros and cons, but the common problem faced with all of them is the unavailability of data related to critical infrastructures which are sensitive and not publically available. To overcome these limitations, a holistic, dynamic and quantitative approach is used [13]. It is based on experiences and the incidents database. This data can be the periods when the critical infrastructure was not operational, or the amount of money spent to acquire a service of another infrastructure.

Whether it's a modeling and simulation approach or a dynamic approach [13] the goal is to identify all the interdependencies. Then analyze cascading effects in order to reduce the impact of interdependencies, and establish an effective policy to prepare for future risks.

Recognizing the need for a deeper understanding of the interdependency in interdependent networks, significant efforts have been made in the research community in the last few years to achieve this goal. Accordingly a number of models have been proposed and analyzed.

A. Network Flow Model

Network-flow models [14] are designed to analyze infrastructure networks and simulate the performance of systems with infrastructural interdependencies. Such models only ensure flow continuity at nodes, while physical laws governing the flow of supplies within infrastructure systems are not fully satisfied. In [15] a new network flow model is introduced which has at its heart a general node model. Any number of different infrastructures can be conceived of as a single graph model that represents all systems composing the infrastructure, and incorporates infrastructural interdependencies. The model description contains graph-theoretic notation. Edges (links) represent interactions between vertices (nodes). For example, edges can be power lines, cables, etc., whereas vertices can be power plants, stations, etc.

Instead of supply and demand vertices, which are typical in the standard approach, the network flow model developed in[15] introduces additional processes of production, consumption and storage to the node. Moreover, a single node represents all these functions. The model developed is intended to simulate the operation of interdependent systems; it allows optimization of infrastructure performance by minimizing the total operational cost associated with production, storage and commodity flow [15].

B. Graph Model

To model interdependent networks, Buldyrev, Parshani, Paul, Stanley, and Havlin in [16] consider two networks, A and B, with the same number of nodes, N. The functioning of node Ai, i in {1, 2…N} in network A, depends on the ability of node Bi, in network B, to provide a critical resource, and vice versa. If node Ai(or Bi) stops functioning, owing to a failure, node Bi (or Ai) will stop functioning. Such dependence is denoted by a bidirectional link, Ai↔Bi that defines a one-to-one correspondence between nodes of network A and nodes of network B. The proposed graph model considers that the number of nodes is assumed to be the same in the interconnected networks, and there exists a one-to-one dependency between these nodes. However, in a

follow up paper [17] same authors opine that this assumption is unrealistic, and a component of an infrastructure may depend on many others of interconnected networks, for this reason the model fails to capture complex interdependencies existing between the entities.

At this stage it is important to clarify the difference between disjunctive and conjunctive dependency. A disjunctive dependency of a node in the A network (ai) on more than one node in the B network (bj and bk), implying that ai may be "alive" as long as either bj or bk is alive, it accounts for conjunctive dependency when both bj and bk has to be alive in order for ai to be alive. In a real network the dependency is likely to be even more complex involving both disjunctive and conjunctive components.

The graph based interdependency models proposed in the literature [16], [17], [18] are simplistic in nature and cannot capture such complex interdependency involving both conjunctive and disjunctive interdependencies. In order to capture them and overcome the limitations of graph based approach, an Implicative Interdependency Model (IIM) using Boolean logic was proposed.

C. Implicative Interdependency Model

Implicative Interdependency Model [19] is an entity based Model that is able to capture complex dependency relationships existing between the entities of interdependent network systems. It uses Boolean Logic to model the interdependencies between networks entities, these interdependent relationships are termed as Implicative Interdependency Relations (IDRs) [19]. Interdependent network setting is represented as I(A,B,F(A,B)), where sets A={a1, a2, .., an}and B={b1, b2,.. bm} are the concerned networks entities, and F(A, B) is the set of dependency relations, or IDRs. Entities can either be in one of two states, operational or failed. The IDR formulation is carried out as follows: the entity ai is operational if:

- Entities bj , bk, bl are operational.
- Or bm, bn are operational.
- Or bp is operational.

we express it in terms of an IDR of the form ai ← bjbkbl + bmbn + bp. Table I represents a sample interdependent network I(A,B,F(A,B)), where A={a1,a2, a3, a4}, B={b1, b2, b3} and F(A, B) is the set of IDRs between the entities of A and B. In this example, the IDR b1 ← a1a3 + a2 implies that entity b1 is operational if both entities a1 and a3 are operational, or entity a2 is operational [19].

TABLE I. Implicative Interdependency Relations of networks A and B

Network A	**Network B**
a1 ← b1b2	b1 ← a1a3 + a2
a2 ← b1 + b2	b2 ← a1a2a3
a3 ← b1 + b2 + b3	b3 ← a1 + a2 + a3

Failure cascade can be derived from the dependency relationships outlined in the IDR set. For example, for the interdependent network outlined in Table I, Table II shows the failure propagation when entities {a2, b3} fail at the initial time step (t = 0). It may be noted that the model assumes that dependent entities fail immediately in the next time step, for example, when {a2, b3} fail at t = 0, b2 fails at t = 1 as b2 is dependent on a2 for its survival. The system reaches a steady state when the failure propagation process stops.

In this example, when {a2, b3} fail at t = 0, the steady state is reached at time step t = 4.

The IIM models the cascading failure process by representing the interdependent networks as a closed loop control system, a failure in one entity in A may lead to the failure of another entity in A indirectly through some B entities.

TABLE II. Failure cascade propagation when entities {a2, b3} fail at time step t = 0. A value of 1 denotes entity failure.

Entity	**Time Steps (t)**						
	0	1	2	3	4	5	6
a1	0	0	1	1	1	1	1
a2	1	1	1	1	1	1	1
a3	0	0	0	0	1	1	1
a4	0	0	0	0	1	1	1
b1	0	0	0	1	1	1	1
b2	0	1	1	1	1	1	1
b3	1	1	1	1	1	1	1

The main challenge for using this model is to formulate the IDRs so as to accurately represent the interdependent system. IDRs can be formed by either careful analysis of the underlying systems, or by consultation with subject matter experts of these systems [19].

Utilizing this comprehensive model, techniques are provided to identify the K most "vulnerable" nodes of interdependent systems. Hence, IIM is utilized to model interdependencies between two networks and analyze the entity hardening problem [20] in order to optimize resources used for protection.

To explain the entity hardening problem, we profit by the help of the previous example. Consider the IDR set shown in Table I. supposing that K= 2, the most vulnerable entities of this system are {a2, b3}. If the network operator doesn't harden any one of the entities a2 or b3, then in this example all the network entities eventually fail, as seen from the fault propagation in Table II.

TABLE III. Entity a2 is hardened

Entity	**Time Steps (t)**				
	0	1	2	3	4
a1	0	0	0	0	0
a2	*	*	*	*	*
a3	0	0	0	0	0
a4	0	0	0	0	0
b1	0	0	0	0	0
b2	0	0	0	0	0
b3	1	1	1	1	1

TABLE IV. Entity b3 is hardened

Entity	**Time Steps (t)**				
	0	1	2	3	4
a1	0	0	1	1	1
a2	1	1	1	1	1
a3	0	0	0	0	0
a4	0	0	0	0	0
b1	0	0	0	1	1
b2	0	1	1	1	1
b3	*	*	*	*	*

When the network operator chooses to harden both a2 and b3 then none of the entities in the network fails. If the network operator has resources to harden only one entity and he chooses to harden a2, the destruction of b3 by a failure will

eventually lead to the failure of no other entity of the network, as shown in Table III. If on the other hand, the network operator chooses to harden b3, destruction of a2 will eventually lead to the failure of the entities {a2, b2, a1, b1} as shown in Table IV. In this case the operator should harden a2 instead of b3.

III. Comparison and discussion

To compare between interdependencies modeling methods cited above, we need to define criteria to be based on, these criteria have been taken from literature:

A. Modeling focus

There are many different ways to classify infrastructure models. We may distinguish between two concepts:

1) *Modeling individual infrastructure systems [15]:*

These approaches tend to identify hidden interdependencies in individual infrastructure systems containing nodes and links.

2) Modeling infrastructures interdependencies [15]:

These techniques are used for identifying critical infrastructures and for analyzing the characteristics and dimensions of their interdependencies.

B. Model strategy:

To proceed with a modeling approach, one of two approaches is used:

1) Bottom-up approach [6]:

The system is described starting with its individual parts. The approach can be built on components characterized by location, capacity, and memory. This approach is considered to be more intuitive and less error-prone than top-down approach.

2) *Top-down approach [6]:*

An overview on the functioning of system must be available, it focuses on the global properties of the system. It is less appropriate than the bottom-up approach to determine the characteristics of the low level.

C. Types of Interdependencies:

In [1] four categories of interdependency are defined:

1) Physical [1]:

Physical interdependencies exist between two infrastructures when their states depend on the material flow transited between them.

2) Cyber [1]:

An infrastructure has cyber interdependency if its state depends on information transmitted through the information infrastructure, it consists of informational links.

3) Geographic [1]:

Infrastructures are geographically interdependent if a local environmental event can create state changes in all of them. Geographic interdependencies occur when elements of infrastructures are in close spatial proximity.

4) Logical [1]:

Two infrastructures are logically interdependent when an agent in one infrastructure is linked to an agent in the other one without any direct connection.

D. Data needs:

This criterion indicates general information about the quantity and quality of data needed by the respective methodical approach. Necessary data include information about the topology, commodity flows, functioning, system description…etc. The use of modeling and simulation approaches depends on data availability and their sufficient quality. Two scales are proposed:

1) High:

The approach strongly depends on a high quantity and quality of data to provide reasonable modeling.

2) Low:

The methodology needs to have a minimum quality or quantity of information.

E. Cascade path:

It refers to the ability of the modeling approach to determine a cascading path in case of failure.

In Table V, we show a comparison between the modeling methods and infrastructure models mentioned in this paper according to criteria cited. To realize this table we were based on previous studies about each model.

TABLE V. Comparison between interdependency models, we mean by P: Physical, C: Cyber, G: Geographic, L: Logical.

Approach	**Modeling focus**		**Model strategy**		**Types of Interdependencies**				**Data needs**		**Cascade path**
	Modeling individual systems	***Modeling infrastructure interdependencies***	***Bottom-up***	***Top-down***	***P***	***L***	***G***	***C***	***High***	***Low***	
ABM	*		*		*	*	*	*	*		*
System Dynamics	*			*	*			*	*		
HLA	*		*	*	*	*	*	*		*	
Hybrid system	*	*	-	-	*	*	*	*	*	*	
Network Flow Model	*		*		*	*	*	*		*	
Graph Model	*	*		*	*	*	*	*		*	
IIM	*	*	*		*	*		*	*		*

In our case, the model we need to adopt must meet the requirements we want to ensure, which are basically:

- Modeling focus: the ability to extract interdependencies within and between infrastructures, which means extracting interdependencies between many systems and between components of an individual one.

- Types of Interdependencies: the ability to determine both physical and logical interdependencies.
- Data needs: the approach we need must depend on a high quantity and quality of data to provide reasonable and precise modeling.
- Cascade path: our approach must be able to find out a cascade path in case of failure.

According to this table, IIM seems to be the most suitable for our needs. We will be based on its properties and include them in our approach.

IV. UML for interdependencies modeling approaches

For a deeper analysis, in this section we model the activity of some approaches using UML sequence diagram to understand how each one operates, so we can analyze and compare them in order to extract their properties and integrate them in our approach. Our purpose is to model and understand and be able to conceive a new approach that meets our requirements.

Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct and document the artifacts of a software system. UML can effectively capture information about the static structure and dynamic behavior of a system [21]. It is the most popular method which could make software product to be reusable, portable, and interoperable. Modeling approaches we will analyze through UML are: ABM, HLA, SD and IIM. For all these methods we consider an infrastructure composed of 4 networks (organizations) to model interdependencies between.

A. Agent-based model (ABM)

Agent-based model consists of modeling a complex system into a set of agents interconnected with one another.

Agents autonomously elaborate information and resources to define their outputs, which become inputs for other agents. Agents are autonomous and computational entities, capable of embedding the characteristics of a system component into their behavior, making the system model representing real functioning of the modeled system [22].

We model ABM using UML as modeling language, for this we consider a critical infrastructure that delivers a specific critical good/service, and composed of 4 networks A B C and D which collaborate to provide required needs. Each one of these networks supplies an output that will be used as an input for other ones; it could be power, communication connection, transaction or data…etc.

Networks that compose the infrastructure are modeled as agents. We distinguish among actors, infrastructures, and the environment. An actor is the entity (human or system) that seeks a service, directly or indirectly from the infrastructure. The infrastructure is a system that provides some service to the actors. The environment is the place in which actors and infrastructures operate. The environment can be a source of perturbations that influence both actors and networks (e.g., a natural disaster, an overload condition).

To produce a sequence diagram of our sample infrastructure and its interdependencies we suppose that each network can be associated to a single agent, and we consider a use case that consists on requesting a service from the infrastructure by an actor.

In the diagram presented in Fig.1 the messages RequestService(X) and ProvideService(X) where X is network A B C or D, represent the output (service) of X which is necessary for the processing continuity in order to respond the actor request. What we can extract from ABM is the functioning way of agents, knowing inputs and outputs of subsystems allows us to establish and define relations between them.

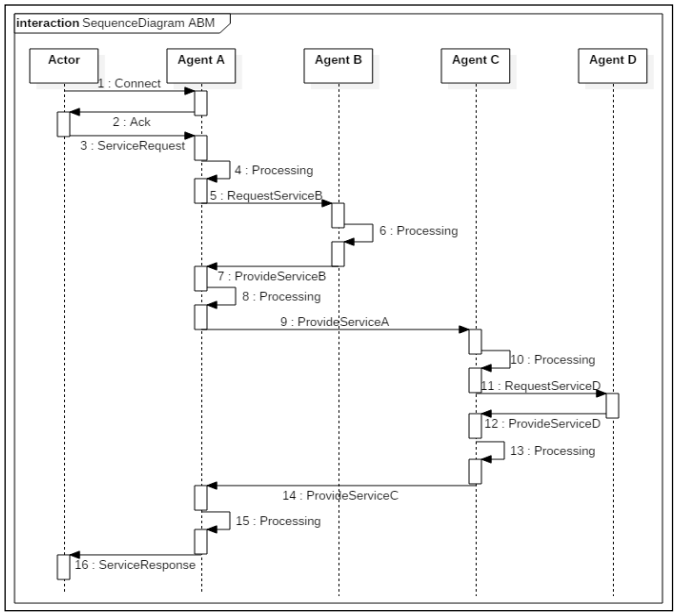

Figure 1. UML sequence diagram for ABM.

Knowing inputs/outputs is necessary to study the influence of traffic change between systems on their operation. An existing model based on inputs/outputs used in economic sector has been applied on CI to study their interdependencies [23]. Leontief input/output model serves to describe interdependencies among CIs; it is constructed from an observed set of data for an area, a country for example. Activities in the area can be categorized into a number of sectors such utilities and transportation. The necessary data are the flows of products from each of the sectors to each of the sectors. In [23], input/output is used to model interdependencies, it has been used to model the interactions among sectors and forecast the impacts of the changes in one on the others.

B. System dynamics (SD)

System dynamics, is used to study and understand the behavior and the underlying structure of a complex system over time [9]. For doing this two diagrams are used:

- Causal-loop diagram [8] the first step is to construct this diagram and to capture the strengths of the basic interactions between the system components. This diagram is composed of:

Stocks: the accumulation of resources in a system component.

Flows: the rates of change that alter those resources.

Information: about the value influences based on changes in the regarded stocks.

The strengths of the interactions are represented by a “+” or

"-"sign. A "+" sign denotes that the causal link is positive, A "-"sign denotes a negative causal link. In Fig.2 the link from stock1 to stock2 is marked "+", changes in the two stocks are in the same direction, i.e. if stock1 increase stock2 will also increase, and vice versa. While the link from stock2 to stock1 shows that if stock2 increase stock1 will decrease.

- Stock-and-flow diagram [8] this diagram conducts the differential equations that causes the evolution of the system. Changes in stocks and flows are described with differential equations.

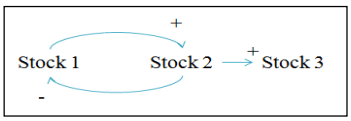

Figure 2. Causal loop diagram.

For modeling SD in UML, we consider the same sample infrastructure used in the previous part. Networks that compose the system are seen as a set of components exchanging stocks.

Sequencing Diagram presents the synchronic sequence or process of the entire system. During the process from stock flow diagram to sequencing; we found some problems about transformation. First, stock flow diagram is synchronous, but sequencing diagram is asynchronous. It is difficult to decide which attributes or operations should be list first in sequencing based on stock flow diagram. Second, sequencing diagram didn't have the concept of time delay, but stock flow did. It's difficult to describe time delay in sequencing diagram [24]. We use always our illustration system; the user here is the human or system person that demands a service from the infrastructure. The UML sequence diagram is presented as shown below:

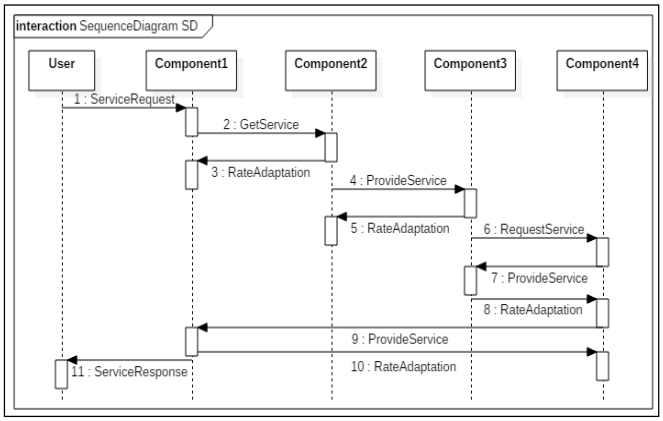

Figure 3. UML sequence diagram for SD.

Stocks of components change after receiving or delivering a service, depending on flows equations. RateAdaptation means that the stock of interacting components is adapted to the changes. SD views the services exchange as stocks and flows altered by differential equations.

System dynamics allows understanding the behavior of an infrastructure's components over time, and deals with internal feedback processes that influence the whole system.

C. High Level Architecture (HLA)

HLA is a general architecture for modeling and simulating complex distributed systems. This technique breaks the entire system down into individually operating subsystems, communication within them is managed by a run time infrastructure (RTI) [11]. This standard can be used to study the dynamic behavior of interdependent infrastructures. In the HLA simulation standard a simulation is called 'federate', all federates are connected via RTI, the set of federates connected via a single RTI is called federation. RTI keeps federates informed about ongoing changes and the state of objects, it ensures the distribution of information to all federates whenever it is updated. Many other major interactions between the federate and the federation are also conducted through the services provided by RTI [11].

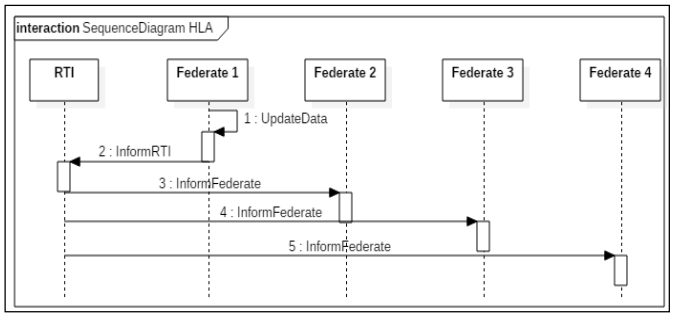

Figure 4. UML sequence diagram for HLA

Whenever information is updated the RTI informs other federates. RTI is considered as a centralized system that keeps all other components of the infrastructure informed with ongoing modifications and updates.

D. Implicative Interdependency Model (IIM)

Implicative Interdependency Model [19] is an entity based Model that is able to capture complex dependency relationships existing between the entities of interdependent network systems. IIM can extract relationships between the components of an infrastructure, so that we can use it to identify weakest elements which trigger the maximum number of the remaining nodes to fail. Defining them is necessary in order to optimize resources used for protection.

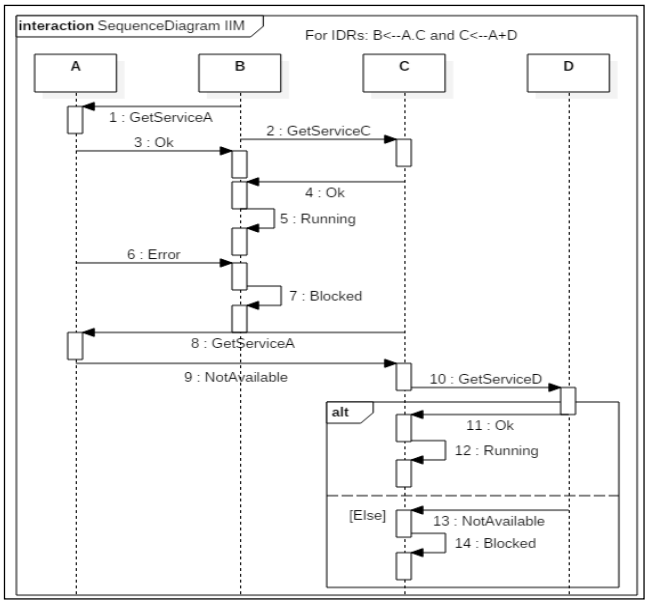

Figure 5. UML sequence diagram for IIM

Figure 1 shows an UML presentation of IIM. We can learn from modeling IIM with UML to inform each network about only related ones basing on interdependency relationships.
The aim of our approach is to not have a "Blocked" state; if a service is not available the system will look for another entity that delivers same service preventing this blocking state.
As we explain before, our future work will be based on IIM to deal with the problems of Critical Infrastructures Protection that are related to interdependencies.

V. Applying IIM on a real architecture

To prove the relevancy of IIM we will apply it on a real world infrastructure. In this section we present the results of applying IIM model on a real critical infrastructure. To study and analyze different aspect of IIM, we apply it on Electric Power Systems (EPS) described in CRUTIAL [26], the European project that addresses new networked systems based on Information and Communication Technology for the management of the electric power grid, we use the EPS to view deeply interdependencies between Electric and information networks. EPS are composed by two cooperating infrastructures: the Electric Infrastructure (EI) [26] for the electricity generation and transportation, and its Information-Technology based Control System (ITCS) [26] that is in charge of controlling and regulating the EI. The architecture of EPS will allow us, at a first scale, to extract interdependencies in each infrastructure separately then determine interdependencies between them.

A. Interdependencies in the EI:

In the Fig.6 we can see the main elements that constitute the overall electric infrastructure[25]: generators (a components), substations (b components), loads(c components) and power lines (d1,i between generators and substations, d2,i between substations and loads). The energy produced by the generators is adapted by transformers (included in substations), to be conveyed to end users (loads), through different power grids. The power lines are components that physically connect the substations with the power plants and the final users, and the substations are structured components in which the electric power is transformed and split over several lines.

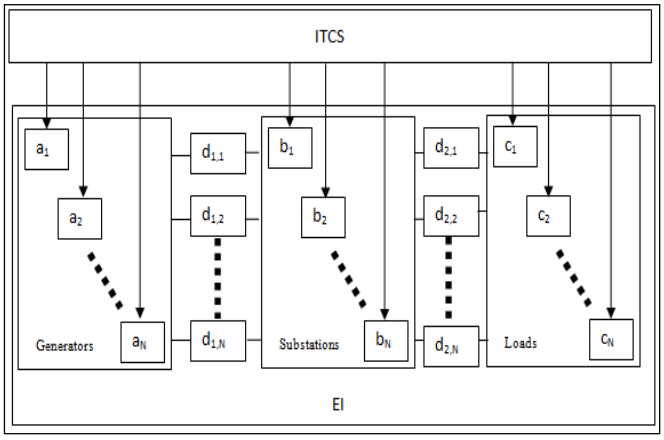

Figure 6. Simplified architecture of Electric Infrastructure

Now we outline dependencies between these elements:
Generators (ai, 0< i <total number of generators) do not depend on any other component of EI. Substations (bi,0<i<total number of substations), that are structured components in which the electric power is transformed and distributed to loads. Each one depends on the generator and power line associated with, the IDR can be formed as: bi← aid1,i .
Loads: end users must be provided by needed energy. Each load depends on the substation with associated line and the generator. The IDR is: ci← bid2,iaid1,i.
Power lines do not depend on other components of EI.
We can deduce from IRDs that the most vulnerable nodes are generators and power lines linking generators with substations.
The fundamental architecture of CRUTIAL is viewed as WAN-of-LANs. The WAN switches packets through facility gateways (representative gateways of each LAN), as example LANs: the administrative LAN; the operational LAN; the engineering LAN; the Internet access LANs …etc [25].
An example of WAN-of-LANs we use to represent a small part of a distribution power grid is shown in Fig.7. It presents a Distribution System Operator (DSO) centre. This centre includes several networks and is connected to the substations through the substation control network. This network is connected to the substations through the WAN. The DSO centre includes the corporate network, the public service network, the operation network, and the data historian network, as shown in fig. 7. All these networks are modeled as LANs and are connected by CRUTIAL Information Switch (CIS) that protect them from one another and from the Internet.

B. Interdependencies in the ITCS:

ITCS is composed of Distribution System Operator (DSO) and Transmission System Operator (TSO) networks, for sake of simplicity we consider that DSO and TSO are of the same architecture. Figure 7 [26] shows the architecture of the DSO network.

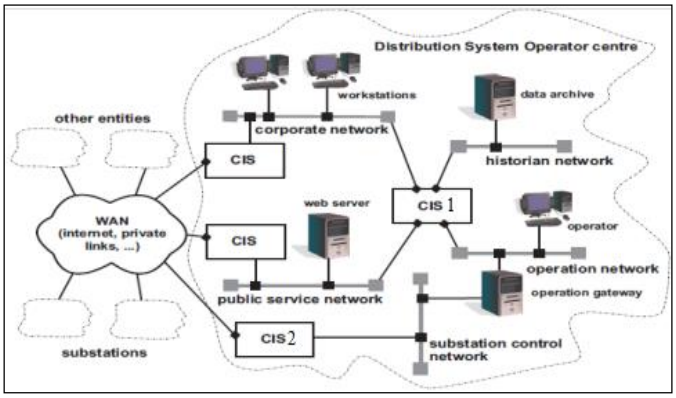

Figure 7. Distribution System Operator LAN.

To study interdependencies among DSO network we refer to Fig.8 which simplifies the DSO infrastructure, we neglect CIS between DSO and other networks (in our case we are interested only in CIS1 and CIS2), the DSO is composed of 5 networks/LANs (n1,n2,n3,n4,n5) connected to the CIS1

(sw1),which insure secure channels for them to communicate, by appropriate linkage (l1,l2,l3,l4,l5,l6), LAN4 and LAN5 communicate through an operation gateway(g).
Substation control network (n5) is connected to substations through the WAN by CIS2 (sw2).
For all these components we establish the corresponding IRDs: LAN1 (corporate network) needs data stored in LAN2, and needs to extract data from LAN4 by means of CIS1. The IDR is: n1 ← l1sw1l2n2+l1sw1l4n4. LAN2 (historian network) where historical information about the infrastructure is stored, the IDR is: n2 ← l2sw1l4n4. LAN3 (public service network) where services like web servers are placed, does not depend on other LANs to be operational. LAN4 (operation network) where operators monitor and control the power generation infrastructure, the IDR is: n4← l5gl6n5.
LAN5 (substation control network) which control substations, receive orders from LAN4, the IDR can be formed as: n5← l5gl6n4.
CIS1: in charge of transferring information between LANs, does not depend on them to operate.
Gateway: insure communication between LAN4 and LAN5.
Links: do not depend on other components.

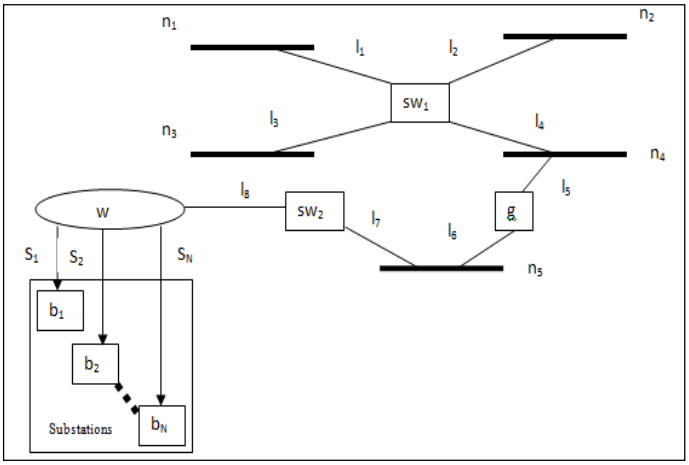

Figure 8. Simplified architecture of Distribution System Operator.

C. Interdependencies between the EI and the ITCS:

Distribution System Operator (DSO) is the part of ITCS that controls substations, and it is connected to them through the substation control network. The communication between DSO and substations is assured by the WAN.
For each substation (bi) to be controlled, cables linking them to substation control network (n5) must be operational. In addition to the WAN and CIS2. The IDR can be formed as: bi←siwl8sw2l7n5.
IIM can extract relationships between the components of an infrastructure, so that we can use it to identify weakest elements which trigger the maximum number of the remaining nodes to fail. By applying IIM to the CRUTIAL framework, we can conclude that it can be applied on any type of infrastructure, either electric or information infrastructure.

VI. Our proposed approach

Our perspectives are to resolve problems related to interdependencies, and be capable to respond to these questions: How to secure a node to remedy the impacts of interdependencies? How to react to a failure of a node, to prevent it to spread over networks and extend the nodes that are interconnected with? Based on the graph model, we consider that infrastructures are a set of nodes related with links. Because of interdependencies impacts, and due to cascading effects, if one node is down, all nodes that are interconnected with it or receive services from, are going to be inoperable. Same thing for links, if a link is down, nodes that are connected through would not be able to communicate with each other.
Our future work is to develop an information system that runs on a service platform and provides information about the interdependencies in an infrastructure. The principle of the application is to notify the components of the system of a failure emerged in most vulnerable nodes who are identified by IIM, or the nodes that need to be protected most. Identifying these elements will allow us to optimize traffic, notification messages, data storage...etc. in order to optimize resources needed for protection. The process of the application will be handled in four steps:

- Define the set of most important elements to protect, using The Implicative Interdependency Model. Interdependency studies intervene at this step. The application must be able to determine these elements.
- Control traffic between critical elements and other components. Then store the control information to keep a trace of traffic data in a database. The control will focus on interdependency relationships, which means elements that depend on each other, in order to maintain their collaboration and functioning.
- Observe if there is any suspicious behavior, basically between interdependent components, by creating a notification programs and triggers to inform the administrator about ongoing changes. And this is the core of the system.
- When a suspicious event emerges in a specific node, the system will inform the administrator and alert other interconnected nodes to react in case if the problem can be resolved automatically. The reaction depends on the suspicious behavior's nature. Probable events could be: add/delete node(s), malfunctioning… etc. and possible reactions to be executed can be switching to a backup node, or reducing cascading effects…etc.

Knowing interdependencies by using IIM will help constructing this application software, because this model allows determination of most vulnerable elements in order to optimize resources used for protection, so that it is possible to secure these components and reduce impacts of interdependencies, and also anticipate probable disruptions.
Since we deal with critical infrastructures, we need a model that allows capturing the interdependencies in the system, because they may be stressed by an unexpected critical event. For example, the impact of a natural disaster on the communication network, as well the impact of a service

failure on other interdependent services.

We adopt UML as modeling language, thus proposing a new way to represent the interdependencies that occur in a complex system composed by different critical infrastructures. Here is the sequence diagram of our system:

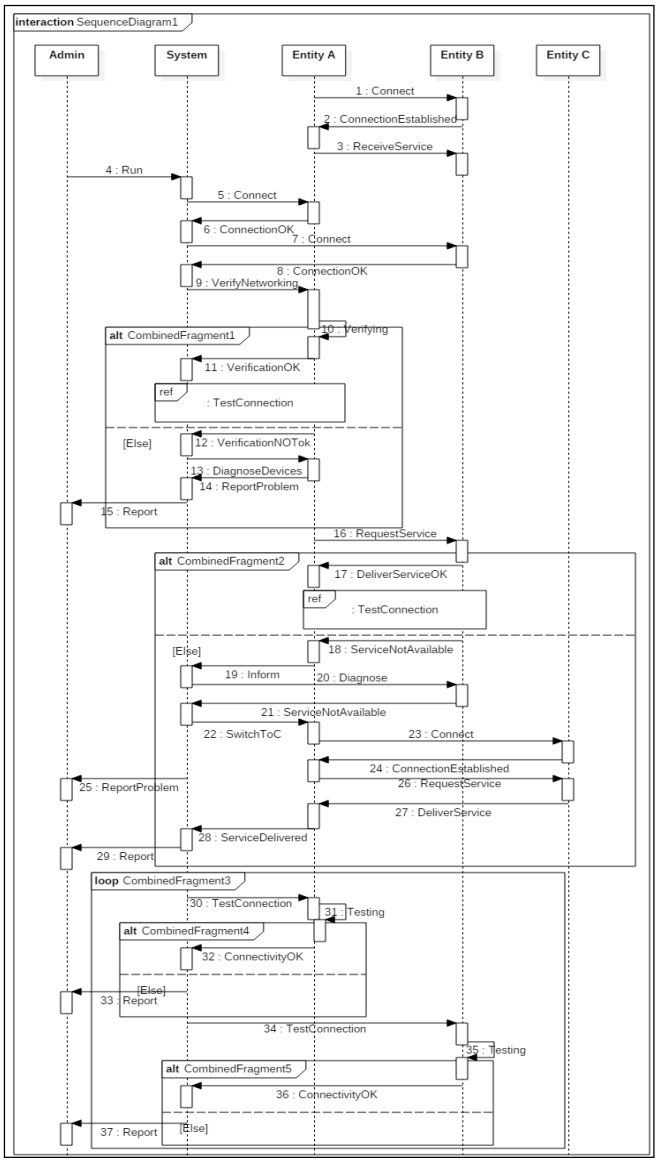

Figure 9. UML sequence diagram for our proposed approach

The centralized system contains information about each node. The parameters of each node are: Node (connected nodes, inputs, outputs, state) connected nodes can be recognized by IRDs, inputs/outputs consists on delivered and needed services, the state may be “operational” or “failed”. Now we show what would be the system behavior when adding or deleting a node:

- Delete node: if a node is deleted, stressed, or just failed the system will try to switch to another node that delivers same service, as shown in the message 18 of fig. 9.
- Add node: if a new node is added o the system it will identify it, determine its inputs/outputs and add it to the database.

VII. Conclusion

In this paper we highlighted the importance of interdependencies study for the protection of critical infrastructures, and we mentioned most used modeling methods to analyze interdependencies. Then according to criteria used in interdependencies modeling, it seems that IIM is the most suitable in our case. And to make certain of this assumption we applied it on a real electric/information infrastructure. Finally, we showed the advantages of IIM and the reason why we choose to use it in our further work, and we gave an overview of the conception of our future platform that will run an information system for handling interdependencies in critical infrastructures.

Our main intend is to protect critical infrastructures regarding interdependencies. Whenever a failure emerges, the plan is to take the system back to normal operation rapidly without disrupting the global functioning of the infrastructure.

References

[1] Rinaldi, Steven M., James P. Peerenboom, and Terrence K. Kelly. "Identifying, understanding, and analyzing critical infrastructure interdependencies." Control Systems, IEEE 21.6 (2001): 11-25.

[2] Zimmerman, Rae, and Carlos E. Restrepo. "Analyzing cascading effects within infrastructure sectors for consequence reduction." Technologies for Homeland Security, 2009. HST'09. IEEE Conference on. IEEE, 2009.

[3] Zio, Enrico, and Giovanni Sansavini. "Modeling interdependent network systems for identifying cascade-safe operating margins." Reliability, IEEE Transactions on 60.1 (2011): 94-101.

[4] Rinaldi, Steven M. "Modeling and simulating critical infrastructures and their interdependencies." System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on. IEEE, 2004.

[5] Casalicchio, Emiliano, Emanuele Galli, and Salvatore Tucci. "Federated agent-based modeling and simulation approach to study interdependencies in IT critical infrastructures." Distributed Simulation and Real-Time Applications, 2007. DS-RT 2007. 11th IEEE International Symposium. IEEE, 2007.

[6] «Software Estimation, Enterprise-Wide » june 2007. available on: www.ibm.com/developerworks/rational/library/jun07/temnenco/.

[7] Barton, Dianne C., and Kevin L. Stamber. An agent-based microsimulation of critical infrastructure systems. No. SAND2000-0808C. Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.

[8] Sterman, John D, «Systems dynamics modeling: tools for learning in a complex world », IEEE Eng. Manag. Rev., vol. 30, no 1, p. 42-42, 2002.

[9] Min, Hyeung-Sik J., et al. "Toward modeling and simulation of critical national infrastructure interdependencies." Iie Transactions 39.1 (2007): 57-71.

[10] Eusgeld, Irene, Cen Nan, and Sven Dietz. "“System-of-systems” approach for interdependent critical

infrastructures." Reliability Engineering & System Safety96.6 (2011): 679-686.
[11] Nan, Cen, and Irene Eusgeld. "Adopting HLA standard for interdependency study." Reliability Engineering & System Safety 96.1 (2011): 149-159.
[12] Lemmon, Michael D., Kevin X. He, and Ivan Markovsky. "Supervisory hybrid systems." Control Systems, IEEE 19.4 (1999): 42-55.
[13] Laugé, Ana, Josune Hernantes, and Jose M. Sarriegi. "Critical infrastructure dependencies: A holistic, dynamic and quantitative approach." International Journal of Critical Infrastructure Protection 8 (2015): 16-23..
[14] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network Flows: Theory, Algorithms, and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
[15] Holden, Richard, et al. "A network flow model for interdependent infrastructures at the local scale." Safety Science 53 (2013): 51-60.
[16] Buldyrev, Sergey V., et al. "Catastrophic cascade of failures in interdependent networks." Nature 464.7291 (2010): 1025-1028.
[17] Gao, Jianxi, et al. "Networks formed from interdependent networks." Nature physics 8.1 (2012): 40-48.
[18] J. Shao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, «Cascade of failures in coupled network systems with multiple support-dependence relations », Phys. Rev. E, vol. 83, no 3, p. 036116, 2011.
[19] A. Sen, A. Mazumder, J. Banerjee, A. Das, and R. Compton, «Identification of K most vulnerable nodes in multi-layered network using a new model of interdependency », in 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014, p. 831-836.
[20] J. Banerjee, A. Das, C. Zhou, A. Mazumder, and A. Sen, «On the Entity Hardening Problem in Multi-layered Interdependent Networks », ArXiv14126686 Cs, dec. 2014.
[21] Zhu H, Li G, Zheng L (2008) A UML profile for HLA-based simulation system modeling. 6th IEEE Int Conf Ind Inform 2008 INDIN 2008, pp 1602–1607.
[22] Cardellini, V., Casalicchio, E., & Galli, E. (2007, March). Agent-based modeling of interdependencies in critical infrastructures through UML. In Proceedings of the 2007 spring simulation multiconference-Volume 2 (pp. 119-126). Society for Computer Simulation International.
[23] Lin, J., Tai, K., Tiong, R. L., & Sim, M. S. (2016). A General Framework for Critical Infrastructure Interdependencies Modeling Using Economic Input-Output Model and Network Analysis. In Complex Systems Design & Management Asia (pp. 59-74). Springer International Publishing.
[24] Chang, L. C., & Tu, Y. M. (2005, July). Attempt to Integrate System Dynamics and UML in Business Process Modeling. In Proceedings of the 23rd International Conference of the System Dynamics Society.
[25] S. Chiaradonna, P. Lollini, et F. Di Giandomenico, «On a Modeling Framework for the Analysis of Interdependencies in Electric Power Systems », in 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007. DSN '07, 2007, p. 185-195.
[26] WP4-D18-final.pdf available on http://crutial.rse-web.it/.

Author Biographies

Chtioui Lamiae was born 1990. She received her engineering degree in computer networks and system in 2014 from National Institute of Posts and Telecommunications Rabat, Morocco. She is currently a Phd student working on information system security for the protection of Critical Infrastructures regarding interdependencies.

Baina Amine is an Assistant Professor at National Institute of Posts and Telecommunications Rabat, Morocco. He had his PhD in Computer Science in "Access Control for critical infrastructure" from the Laboratory of Systems Analysis and Architecture in Toulouse. He had his Computer Engineer's degree from the National Engineering School of Bourges, France.

Mostafa BELLAFKIH is a Professor of Higher Education at National Institute of Posts and Telecommunications Rabat, Morocco with a doctorate in Applied Sciences (Networks and computer systems) from the Mohammadia School of Engineers, University Mohamed V Rabat in 2001. He achieved his doctoral thesis in computer sciences in 1994 from the University Pierre-et-Marie-Curie.