Universidad de Valladolid

Escuela de Doctorado Universidad de Valladolid

PROGRAMA DE DOCTORADO EN INGENIERÍA INDUSTRIAL

TESIS DOCTORAL:

A Framework to Study the Resilience of Organizations: A Case Study of a Nuclear Emergency Plan

Presentada por Cristina Ruiz Martín para optar al grado de
Doctor/a por la Universidad de Valladolid

Dirigida por:
Dr. Adolfo López Paredes
Dr. Gabriel A. Wainer

A Framework to Study the Resilience of Organizations: A Case Study of a Nuclear Emergency Plan

by

Cristina Ruiz Martín

Thesis Supervisors
Dr. Gabriel A. Wainer / Dr. Adolfo López Paredes

A thesis submitted to the Faculty of Graduate Studies and Research

In partial fulfillment of the requirements of the degree of
Doctor of Philosophy in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario
Canada

January 2018

Abstract

Nowadays, building resilience is a key topic in many research fields such as Management, Engineering, Psychology or Ecology. The frequency increase of natural and anthropogenic disasters and the consciousness about their effects are among the reasons why resilience has gained importance and Governments are investing money in boosting the resilience of organizations, infrastructure, cities, individuals, etc.

However, there is not much research on specific methodologies to design resilient organizations. A main goal of our research is to improve this aspect providing a framework to design resilient organizations. We explain how to design resilient organizations based on the Viable System Model principles. Then, we focus on an important aspect for being resilient: the communications. We use as a case study a Nuclear Emergency Plan from Spain to show the applicability of our framework.

Since the communications in an organization can be modeled as a diffusion process in multiplex networks, and we did not find any suitable architecture to study them in the context of our case study, the architecture we design in this thesis is generic and allows us to model and simulate any kind of diffusion process in a dynamic multiplex network.

Resumen

El desarrollo de la resiliencia es un campo de investigación importante en ámbitos como el Management, la Ingeniería, la Psicología o la Ecología. La importancia del estudio de la resiliencia se ha visto desarrollada por el aumento tanto de desastres naturales como antropogénicos, así como por el desarrollo de conciencia acerca de sus efectos. Estas razones de peso han influido en que los Gobiernos estén invirtiendo recursos en la mejora de la resiliencia de organizaciones, infraestructuras, ciudades, individuos, etc.

Sin embargo, a pesar de su importancia, el número de trabajos de investigación que se centran en el desarrollo de metodologías específicas para el diseño de organizaciones resilientes es reducido. El principal objetivo de esta investigación es mejorar este aspecto introduciendo un marco para el diseño de organizaciones resilientes. Para alcanzar este objetivo, se explica cómo emplear el Modelo de Sistemas Viables para el diseño de estas organizaciones. Nos hemos centrado en uno de los aspectos clave de la resiliencia: las comunicaciones. Para ello, se ha usado el caso de estudio del plan de emergencia de una central nuclear en España.

Las comunicaciones en una organización pueden modelarse como un proceso de difusión en redes multiplex. Buscamos arquitecturas aplicables a nuestro caso de estudio. Sin embargo, no se ha encontrado ninguna que cumpliera con los requisitos que se necesitaban. Este hecho, nos ha llevado a proponer una nueva arquitectura, que además de permitir estudiar la difusión de información en una organización, permite estudiar otros procesos de difusión en redes multiplex.

Acknowledgements

I want to express my gratitude to:

My thesis supervisors *Dr. Adolfo López Paredes* and *Dr. Gabriel Wainer* for being a guide during these years. Thanks for sharing your knowledge, expertise and give me the opportunity to do my PhD. in an international environment. Without your support and guidelines, this thesis would have not exit.

My colleagues in *INSISOC* and *ARS-Lab*. Thanks a lot for all your feedback and support. I also want to express my gratitude for making the work environment a wonderful place.

My *mum* and *family* for always being there.

I also love to thank every single person that has made me feel like at home during my research stays at Carleton.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

This thesis started as a research collaboration with the Civil Protection Agency in Castilla y León with the objective of analyzing emergency plans. The start of this collaboration came at the same time as the nuclear accident at TEPCO's Fukushima NPP. Since different studies pointed out that the emergency plan was not resilient (Langlois 2013), we decided to ask ourselves: How can we design resilient emergency plans?

To answer this question, since emergency plans can be considered a virtual organization, we started reviewing the concept of organizational resilience and how to measure it. We found a close relationship between resilient and viable organizations. Taking into account this relation, we proposed to apply the methodology to design viable organizations, introduced by Pérez Rios (2010), to the design of resilient organizations. This methodology, based on the Viable System Model (VSM), highlights the importance of the communications for the well-functioning of the organization.

Due to the importance of the communications, we have focused on their study. The communications inside organizations can be studied as a diffusion process in multiplex networks. In the Laboratoire de l'Intégration du Matériau au Système (IMS) at University of Bordeaux, Bouanan et al. (2016) developed an architecture to simulate information diffusion processes in social networks. Through a collaboration with IMS, we studied the applicability of their architecture to study resilience of communications inside organizations. We used the case study provided by the Civil Protection Agency: "Study a Nuclear Emergency Plan (NEP) from Spain". We found that their architecture was not suitable to include all the attributes specified in plan as detail in Chapter 4. As a result, we developed a new architecture that fulfills our requirements. We used the same modeling and simulation methodologies proposed in Bouanan et al. (2016): Agent Based Modeling (ABM), Network Theory and Discrete Event System Specification (DEVS).

We built the architecture using a bottom-up approach. First, based on the above-mentioned methodologies and their associated tools, we designed a model to study the communications inside emergency plans using the case study proposed by the Civil Protection Agency. Then, we extrapolated the development process and the specific model to provide a general architecture to study diffusion processes in multiplex dynamic networks, being the study of the communications inside organizations a particular case. Although we followed a bottom up approach to design the architecture, in the thesis, we first explain the general architecture and development process, and then how to apply it to the case study.

The application of the methodology introduced by Perez Rios (2010) and the architecture and the development process we propose in this thesis to study diffusion process in multiplex networks constitute a framework to study the resilience inside organizations. We understand as framework a standard set of practices and methodologies that allows us to study a specific problem and similar problems to the original one.

Through this study, we proposed improvements to the communications in the emergency plan proposed by the Civil Protection Agency. The framework developed in this thesis will also be the basis to study other emergency plans and other organizations. Therefore, although our initial goal was

to design resilient emergency plans, we came up with a framework to design resilient organizations using formal methods.

1.1. Research goals and Contributions

The final goal of this thesis is to provide a framework to design resilient organizations using formal methods.

To achieve this goal, we define the following objectives:

- Define resilient organizations, identify what are the characteristics that contribute to their development and understand how organizational resilience is measured.
- Identify a methodology to design resilient organizations
- Establish an architecture and a development process to study the resilience of communications inside organizations.

The contributions of this thesis are as follows:

- A conceptualization of organizational resilience and the identification of the main characteristics that contribute to their development.
- A four level maturity model for organizational resilience.
- The identification of two streams to measure organizational resilience: before and after the disruptive event occurs.
- The relationship between resilient and viable organizations that set the justification for the application of the VSM to the design of resilient organization.
- The identification of the communications inside organizations as a key element for their resilience.
- A general architecture and a development process to simulate diffusion processes in multiplex networks based on formal modeling and simulation methodologies.
- An instantiation of the architecture to simulate diffusion processes in multiplex networks using DEVS as formal modeling and simulation methodology.
- A customizable model to simulate information diffusion processes inside organizations taking into account the social aspect (i.e. the behavior of the people that carry the diffusion process) to study the resilience of the communications.
- The application of DEVS to provide rigor to study the resilience of communications inside organizations
- A framework to design resilient organizations based on the methodology introduced by Perez Rios (2010) and an architecture to simulate diffusion processes in multiplex networks.
- The application of the framework to study the resilience of a Spanish NEP

1.2. Research Environment

This thesis has been developed in a Cotutelle program between two research groups: INSISOC (INgeniería de los SIstemas SOCiales) and ARS-Lab (Advanced Real-Time Simulation Laboratory).

INSISOC is a research group (Excellence Research Group of Castilla y León) integrated by researchers and professors from *Universidad de Valladolid* and *Universidad de Burgos*. Nowadays, the group has fifteen researchers.

INSISOC was born in 1998 when Professor Cesáreo Hernández supervised the thesis of Professor Adolfo Lopez Paredes entitled "Analysis and Engineering the Economic Institutions. An Agent Based Methodology" ("Análisis e Ingeniería de las Instituciones Económicas. Una metodología basada en agentes"). The initial milestone of the group is the article entitle "The Social Dimension of Economics and Multiagent Systems" written by Professor López-Paredes and Professor del Olmo in 1998.

INSISOC is focused on the study of complex systems. We model and study the behavior of complex social systems defining the behavior of the components. Our aim is to explore and develop methodologies to study this type of systems and problems.

ARS-Lab was funded by Professor Gabriel Wainer once he joined Carleton University. Nowadays, the laboratory has eleven researchers, most of them Ph.D. students. Additionally, every year, we have visitor researchers and professor from different labs all over the world (France, Argentina, Brazil, etc.). The research in the laboratory is based on DEVS formalism. The aim is to augment previous work with new theory, methodology, and supporting development tools, including the integration of 3D visualization facilities.

In this thesis, we have taken advantage of the background and expertise in both groups: the study of complex systems using Agent Based Modeling (ABM) techniques and the expertise in formal methodologies (DEVS in the case of this thesis). Combining their expertise, we have been able to provide a framework to study the resilience of organizations.

1.3. Related Publications

We published some of the obtained results in different conference proceedings, journals and a book chapter. At the time of writing this thesis, we have two journal papers that we are waiting for them to be published.

The list of the related publication to this thesis is classified by type and listed based on the date of publication in a ascending order.

Journal articles

Cristina Ruiz-Martin, Mario Ramirez-Ferrero, José Luis González Álvarez, Adolfo Lopez-Paredes. *"Modeling of a Nuclear Emergency Plan: Communication Management".* Human and Ecological Risk Assessment: An International Journal, 21(5), 1152-1168. (2015). https://doi.org/10.1080/10807039.2014.955383

Summary: Using the case study provided by the Civil Protection Agency (the NEP), we modeled the communications in the organization using Network Theory. We used the model to study the properties of the Network such as the network diameter, the degree distribution, the average path length, etc. using Gephi. We used this analysis to suggest improvements to the NEP.

Cristina Ruiz-Martin, David J. Poza Garcia. *"Project Configuration by means of Network Theory"*. International Journal of Project Management. 33 - 8, pp. 1755 - 1767. 2015. https://doi.org/10.1016/j.ijproman.2015.07.010

Summary: We proposed to determine an appropriate sequence to develop the components of a Project Management Plan using Network Theory. Although our approach is compatible with any project management standard, we used the Project Management Body of Knowledge (PMBOK) to illustrate how to apply this methodology due to the complex interdependence among its processes. We built the Network Model of the PMBOK as we did with the NEP.

Manuel Morales Allende, **Cristina Ruiz-Martin**, Adolfo Lopez-Paredes, José Manuel Perez Ríos. *"Aligning Organizational Pathologies and Organizational Resilience Indicators"*. International Journal of Production Management and Engineering, 5(2), 107-116. (July 2017). https://doi.org/10.4995/raet.2017.7423

Summary: In this article, based on the discussion presented in the book chapter *"The Application of the Viable System Model to Enhance Organizational Resilience"* we proposed to identify the organizational pathologies defined applying the VSM using resilience indicators. We concluded that an organization with any organizational pathology is not likely to be resilient because it does not fulfill the requirements of viable organizations.

Cristina Ruiz-Martin, Adolfo Lopez-Paredes, Gabriel Wainer. *"What we Know and Do Not Know about Organizational Resilience"*. International Journal of Production Management and Engineering. (Accepted. In Press).

Summary: We presented a literature review about organizational resilience. The main contributions of this review are a conceptualization of organizational resilience, a four-level Maturity Model for Organizational Resilience (MMOR) based on the development of the abilities or capacities the organization has to deal with disruptive events and the identification of two streams to measure organizational resilience.

Cristina Ruiz-Martin, Gabriel Wainer, Adolfo Lopez-Paredes. *"Discrete-Event Simulation of Diffusion Processes in Dynamic Multiplex Networks"*. Simulation Modelling Practice and Theory. (Revisions Submitted).

Summary: We defined an architecture and a development process to study diffusion processes in multiplex dynamic networks based Network Theory, Agent-Based Modeling and formal M&S. We detailed the development process and the architecture using DEVS as the formal M&S methodology, and presented a case study based on the NEP.

Cristina Ruiz-Martin, Adolfo Lopez-Paredes, Gabriel Wainer. *"Assessment of Organizational Resilience through Network Theory"*. Dirección y Organización. (Submitted)

Summary: We applied Network Theory to do static analyses of the communication network established in the NEP. We studied how a failure in different communication systems affects the network connectivity and therefore the resilience information transmission process

Eduardo Agenjo, Natalia Martín-Cruz, **Cristina Ruiz-Martin**, Adolfo Lopez-Paredes. *"Does CMMI Implementation affect the Performance of the Firm? An Evaluation from a Dynamic Capabilities Approach".* International Journal of Production Management and Engineering (Submitted)

Summary: We studied the impact of the Capability Maturity Model Integration (CMMI) on firm performance both during and after its implementation. We used Spanish firms in the Information and Technology (IT) sector. Doing statistical analysis we found a negative relationship between the use of CMMI and profitability in the firms during the analyzed period and sector.

Book Chapters

Cristina Ruiz-Martin, Jose Manuel Pérez Rios, Gabriel Wainer, Javier Pajares, Cesareo Hernandez, Adolfo Lopez-Paredes. *"The Application of the Viable System Model to Enhance Organizational Resilience"*. In Advances in Management Engineering. Springer 2017. ISBN: 978-3-319-55888-2. https://doi.org/10.1007/978-3-319-55889-9_5

Summary: In this book chapter, we identified a relationship between viable and resilient organizations. We argued that the application of the principles of the Viable System Model (VSM) improves organizational resilience. We also argued that the VSM constitutes a valid framework to design resilient organizations.

Conferences

Cristina Ruiz-Martin, Adolfo Lopez-Paredes, Gabriel Wainer. *"Applying Complex Network Theory to the Assessment of Organizational Resilience".* INCOM 2015. IFAC-PapersOnLine 48-3 p. 1224–1229. Ottawa, Canada. May 2015. (Best Paper Award) https://doi.org/10.1016/j.ifacol.2015.06.251.

Cristina Ruiz-Martin, Gabriel Wainer, Adolfo Lopez-Paredes. *"Modeling the Communications in an Emergency Plan with P – DEVS"* Winter Simulation Conference (WSC). Ph.D. Colloquium. IEEE Press, p. 3086-3087 Huntington Beach, US. December 2015. https://doi.org/10.1109/WSC.2015.7408412

Cristina Ruiz-Martin, Gabriel Wainer, José Manuel Pérez Ríos, Javier Pajares, Cesáreo Hernández, Adolfo Lopez-Paredes. *"Organizational Resilience in Practice: the Viable System Model"* International Joint Conference (IJC2016). San Sebastián, País Vasco, España. July 2016. (Poster)

Gabriel Wainer, **Cristina Ruiz-Martin,** Adolfo Lopez-Paredes. *"Cellular Models for Emerging Traffic Behaviour"* Second International Symposium on Cellular Automata Modeling for Urban and Spatial Systems (CAMUSS 2016) Quebec City, Canada. September 2016.

Cristina Ruiz-Martin, Youssef Bouanan, Gabriel Wainer, Gregory Zacharewicz, Adolfo Lopez-Paredes. *"A Hybrid Approach to Study Communication in Emergency* Plans" Winter Simulation Conference (WSC). IEEE Press, p. 1376-1387 Arlington, US. December 2016. https://doi.org/10.1109/WSC.2016.7822191

Eduardo Agenjo, Natalia Martín-Cruz, **Cristina Ruiz-Martin**, Adolfo Lopez-Paredes. *"The Impact of CMMI Implementation on the Firm Performance. An Evaluation from a Dynamic Capabilities Approach".* International Joint Conference - ICIEOM-ADINGOR-IISE-AIM-ASEM (IJC 2017) Valencia, Spain, July 2017. (Best Paper Award)

Cristina Ruiz-Martin, Felix Villafañez, Adolfo Lopez-Paredes, Gabriel Wainer. *"Impact of Business Intelligence in Organizational Resilience"* 2017 INFORMS Annual Meeting, Houston. USA. October 2017 (Abstract)

Cristina Ruiz-Martin, Gabriel Wainer, Adolfo Lopez-Paredes. *"An Architecture to Simulate Diffusion Processes in Multiplex Dynamic Networks"*. Winter Simulation Conference (WSC). Ph.D. Colloquium. Las Vegas, US. December 2017.

Cristina Ruiz-Martin, Gabriel Wainer, Adolfo Lopez-Paredes. *"Formal Abstract Modeling of Dynamic Multiplex Networks"*. SIGSIM-PADS18. Rome, Italy. May 2018 (Submitted)

1.4. Structure of the Thesis

The rest of this thesis is organized as follows:

In Chapter 2, we present the background related to this thesis. First, we present the problem we address in this thesis. We discuss recent works in emergency management and we review the concept of organizational resilience. We also review the concept of viable organizations and the VSM.

In Chapter 3, we explain the three methodologies we use to develop our architecture to simulate diffusion processes in multiplex networks: Network Theory, ABM and DEVS.

In Chapter 4, we propose a definition of organizational resilience that integrates the ones presented in the review provided in Chapter 2 and discuss the measurement of organizational resilience. We also relate viable and resilient organizations and we defend the application of the VSM to design resilient organizations.

In Chapter 5, we present the preliminary work for the architecture proposed in this thesis to simulate diffusion processes in multiplex networks.

In Chapter 6, we detail the proposed architecture and development process to simulate diffusion processes in multiplex networks using formal modeling and simulation methodologies. We also explain how to instantiate the architecture when we use DEVS as formal modeling and simulation methodology.

In Chapter 7, we detail how to use the Data Experiment Collection component of the architecture and the Step 1 of the development process using as a case study a NEP from Spain. We also explain how to obtain the Network model and Agent-Based model components of our architecture using the Step 2 of the development process. This case study is also used in the rest of the chapters to explain the other components of the architecture and the other steps of the development process.

In Chapter 8, we focus on the definition and implementation of the Diffusion Abstract model for the NEP using DEVS and Steps 3&4 of the development process

In Chapter 9, we present some results of analyzing the Network model.

In Chapter 10, we explain how to analyze the simulation results obtained using the NEP Diffusion Abstract model and provide relevant information for decision-makers.

In Chapter 11, we state the conclusion of this work and present future research lines.

Chapter 2. Background

In this chapter, we first describe the problem we aim to address in this thesis: the design of resilient organizations. Since we aim to design resilient organizations, we will provide a review about organizational resilience (how it is understood and measure) and the concept of viable organizations and the VSM.

After the review of organizational resilience, in Chapter 4, we define resilience at the organizational level, *as the measurable combination of characteristics, abilities, capacities or capabilities that allows an organization to withstand known and unknown disturbances and still survive*. Since resilience is mentioned several times along this chapter, we introduce the definition here.

2.1. Description of the Problem

Nowadays, building resilience is a key topic in many research fields such as Management, Engineering, Psychology or Ecology. Governments are investing resources to develop resilient institutions, communities, organizations, and individuals.

Since 2010, the US Department of Homeland Security has evolved from discussing what resilience means to set three principles to develop resilience: (1) adaptability, (2) withstanding and (3) rapidly recovering. The European Commision (2017) has also identified resilience among the top five European Union's priorities. The European Union Action Plan for Resilience (European Commission 2016) outlines three priorities in the area of resilience: (1) support the development and implementation of national resilience capacity, (2) promote innovation and learning capacities to advocate resilience and (3) develop tools and methodologies to improve and measure resilience.

The Government of Canada is also focused on resilience, especially on climate resilience since climate change is strongly affecting Canada (Goverment of Canada 2014). They propose to build climate change resilience based on the following actions: (1) translating scientific information and traditional knowledge into action, (2) building resilient infrastructure, (3) protecting and improving human health and well-being, (4) supporting vulnerable regions and (5) reducing the hazards related to climate change and risk of disaster.

Although there are differences about how to build on resilience, there is no doubt that the frequency increase of natural and anthropogenic disasters and the consciousness about their effects are among the reasons why this topic has gained importance and Governments are investing money in improving the resilience of their country, including organizations, infrastructure, cities, individuals, etc.

Moreover, recent disasters have shown evidence of the catastrophic consequences and have revealed that not all of hazards can be prevented (Hosseini et al. 2016; Lalonde 2007). We do not need to look far to find some recent disaster. A few months ago (August/September 2017), several hurricanes in the Atlantic Ocean (Hurricane Harvey, Irma, and Maria) devastated several Caribbean islands and had several consequences is the US such as Texas or Florida. During this same period, Mexico suffered a devastating earthquake. Other examples of disasters in recent history include the

Earthquake and Tsunami in Japan in 2011 (MacKenzie et al. 2012), the Darfield Earthquake in New Zealand in 2010 (Whitman et al. 2014; Kachali et al. 2012), and Hurricane Katrina in 2005 (Garnett & Kouzmin 2007). Likewise, anthropogenic disaster has occurred such as the accident at TEPCO's Fukushima Nuclear Power Plant (NPP) in 2011 (Omoto 2013; Langlois 2013) or the World Trade Centre attack in September 2001 (Kendra & Wachtendorf 2003; Mendonça & Wallace 2015).

To overcome the above-mentioned situations, Emergency Plans are designated. However, the traditional approaches to design them are based on top-down perspectives that aim for the compliance of a set of laws, regulations, and directives. These traditional approaches usually focus on a hierarchical structure similar to a military command chain giving a small margin to adaptation to unforeseen circumstances not identified in the plan.

Emergency plans can be considered a virtual organization, where members from different organizations get organized according to the definitions in the emergency plan with a specific purpose: solve the emergency.

We have chosen emergency plans as the organizations to test our framework because they are complex organizations where the communication between the people involved is a key element for the coordination and well function. This is important in any type of organization, but especially, in Emergency Plans, where having up to date information and data is critical. Moreover, the literature remarks the importance of improving emergency plans and we have access to data of a real NEP provided by the Civil Protection Agency.

2.1.1. Examples of recent disasters and their consequences

After the accident at TEPCO's Fukushima NPP, several problems in the emergency plan and crisis management were identified, including the loss of functionality at the off-site emergency management center. One of the causes was the lack of availability of the communication systems. The emergency plan was not well implemented in terms of warning the population, evacuation, distribution of iodine tablets, etc. The responsibilities were not well defined. There were a poor communication and information management. The analysis of the above issues suggested, among other actions, the review of the communication and management systems in the emergency plans (Omoto 2013). Doing a similar analysis, the International Atomic Energy Agency (IAEA) suggested the need for improvements. These improvements included strengthening management systems, response arrangements, transparency, and effectiveness of communications mechanisms (Langlois 2013).

Following the Darfield earthquake on September 2010, in (Whitman et al. 2014), the authors carried out a survey among New Zealand's organizations. They used this survey to find the most helpful factors in mitigating the disruption in the operations after the earthquake. These factors are well-designed and well-built buildings, the relationship with staff and the capability to restore critical services quickly or not to get them interrupted.

After Hurricane Katrina in New Orleans in 2005, the most important industries in the area such as the fishing, the cotton, the rice or the sugarcane industries were destroyed. In (Chewning et al. 2012), the authors studied how Information and Communication Technology (ICT) were used by organizations to aid in their recovery after the Hurricane. Through their empirical study, they showed that the organizations that have the ability to use ICT in these situations are more resilient. These

organizations used ICTS to improve their connection, coordination and share the evolution of the emergency with both external and internal stakeholders.

After the World Trade Centre attack in September 2001, the New York City departments lost their primary emergency operation center. In (Kendra & Wachtendorf 2003), the authors examined how organizational resilience was exhibited in the recovering activities. They concluded that anticipation is a key dimension of resilience. This anticipation lays in the design of the organization, training, and preparation. However, they pointed out that creative thinking, flexibility, and ability to improvise in new emergent situations are also important. The analysis of these crises showed that organizations have to improve their capacity to adapt and reorganize when unforeseen events occur. They also pointed out the importance of the communication systems.

2.1.2. Previous research on resilience and emergency management

Following these examples provided in the previous section, we consider that a resilient communication system contributes to improving the emergency management, which lays on emergency plans.

The importance of the communication mechanisms in emergency plans is also remarked by the principles of resilient systems and resilient organizations (Longstaff & Yang 2008) since emergency plans can be considered a virtual organization.

Although the previous examples remark the importance of improving the resilience of the communications inside emergency plans (a type of organization), there are not many tools or methodologies that allow us to test and improve the communications in organizations. This number is fewer if we include the social aspect (i.e. the behavior of the people that handle these communications).

Previous research in this area focused on identifying factors that help to improve organizational resilience. For example, Folke et al. (2005) reviewed the main features to deal with crisis, changes and to build resilience. Through this review, they found four important factors that contribute to improving the organizational resilience. These factors include building knowledge and understanding the resource and ecosystem dynamics; feed ecological knowledge into adaptive management practices; support flexible institutions and multilevel governance systems; and deal with external perturbations, uncertainty, and surprise. Carroll (1998) described how self-analysis of operating problems in organizations can improve their resilience. However, he also illustrated how the logics underlying these activities depend on the socio-cultural context such as hierarchy or occupational groups. This context dependence can cause conflicts and communications problems inside the organization. Crichton et al. (2009) studied incidents in different sectors, and found common aspects among them. They concluded that the application of cross-sector lessons learned during crisis management can improve organizational resilience. McManus et al. (2008) proposed a process to improve the organizational resilience. This process includes six factors: building awareness (i.e. identifying and understanding the elements that contribute to organizational resilience), selection of essential organizational components, self-assessment of vulnerability, identification and prioritization of key vulnerabilities, and increasing adaptive capacity (the ability of the organization to make appropriate decisions in time, both daily and in crises).

However, these authors do not propose any tool to evaluate how resilient is an organization. Lee et al. (2013) worked in this direction introducing a survey tool to measure the resilience of organizations. They measured organizational resilience based on two factors: adaptive capacity and planning. To analyze how the organization performs based on these factors, they defined several indicators (eight for adaptive capacity and five for planning). For each of them, they identified their strengths and weakness and this is used as a measure of the resilience. However, this does not provide a systematic method to analyze the resilience of the organization in terms of communications and information management.

Several works have specifically focused on improving the resilience of emergency plans from different perspectives. Some of them focused on identifying the factors that improve the resilience. For example, Zhou et al. (2011) proposed five critical success factors for emergency management. These top factors include the organizational structure, a clear definition of responsibilities and the effectiveness of the information system to ensure the transference of information.

Others aimed to improve the organizational resilience through the development of processes. For instance, in (Turoff et al. 2004), the authors proposed eight design principles to build a flexible, robust and dynamic information management system for emergency response. These principles highlight the importance of up-to-date data, well-defined roles, information sharing across the organization, etc.

There is also research focused on improving the resilience in emergencies through the evacuation performance. For example, in (Lv et al. 2013), the authors introduced a new method for evacuation management support. They used interval-parameter programming within joint-probabilistic constrained programming and integer linear programming (optimization technique that deals with uncertainty) to calculate the optimal evacuation route. They applied the model to calculate the evacuation route in different scenarios. Applying this model, they also evaluated the robustness of the system analyzing what is the influence when the constraints for the evacuation problem are modified. Simonovic & Ahmad (2005) developed a simulation model based on system dynamics for understanding human behavior during flood evacuations. Their aim was to simulate the effect of different evacuation policies. They applied this model to emergency planning in the Red River Basin. Chen et al. (2006) applied agent-based simulation to identify the time it takes to evacuate the Florida Keys in case of a hurricane. They also studied what is the effect of a landfall in the evacuation route. They identified the most congested roads and the bottlenecks. They also got the average speed of the vehicles. In (Hammond & Bier 2015) the authors aimed to identify alternative evacuation strategies for nuclear emergencies. They studied different strategies based on the predicted radiation plumes. They compared them in terms of the size of the evacuation area and the adequacy of the protection measures and chose three. They compared these three to the existing ones and they conclude that there are methods that perform better than the ones currently applied.

Studies on how to improve resilience by decreasing the uncertainty level or the failures in the plan have also been developed. For example, Bañuls et al. (2013) developed collaborative scenarios using Cross-Impact Analysis, a methodology to find relationships between events and to reduce uncertainty. To analyze the resultant graph they used Interpretive Structural Modeling, a methodology to identify and summarize the relationship between the variables. The authors used these scenarios to assist in developing emergency plans and as a training tool. Karagiannis et al. (2010) proposed a generic model for internal industrial emergency plans that can be instantiated for specific industries. The authors used this generic model to perform an iterative risk analysis to identify possible failures in the

implementation of the plan. Their objective was to improve the robustness of the emergency plans. Gomes et al. (2014) analyzed a nuclear emergency plan exercise. They studied a real simulacrum with real people to analyze the messages passed and to find sources of resilience and weakness. Their aim was to identify sources of improvement for future simulacrums to increase the resilience of the emergency plan.

Other research works focused on improving resilience improving the decision support systems. For example, in (De Maio et al. 2011), the authors proposed a knowledge-based system that includes the information from social, software and web technologies to support emergency management. The decision support system uses Fuzzy Cognitive Maps to deal with the complexity of this information (a Fuzzy Cognitive Map is a type of representation in which the relations between the elements can be used to calculate how important the impact of each element is). With this tool, they assisted the emergency manager during the crisis. In (Espinosa-Paredes et al. 2008), the authors applied Fuzzy Cognitive Maps to represent the decision-making process during abnormal situations in a NPP. Their case study considers the loss of coolant in a boiling water reactor, and they simulated different scenarios to test their approach. They presented a way to predict the effects and causes in a complex system such as an NPP. They also provided a tool that helps in the decision-making during emergencies in nuclear power plants.

Mendonça et al. (2006) applied gaming simulation to evaluate decision support systems for emergency response and to train the people involved using virtual environments. These virtual environments provide data (e.g. activity record, log of the communications, data recorded by observers, etc.) that can be used to evaluate the system.

Other authors focused on improving resilience through anticipation. For example, Chang et al. (2006) analyzed where to reallocate off-site monitoring installations using both simulation and optimization techniques. Their goal was to reduce the current monitoring network without affecting its monitoring capacity. In (Park & Jung 2007), the authors proposed a measure to quantify the complexity of tasks in emergency procedures in an NPP. This measure is calculated using five factors: the amount of information to be managed, the logic in the sequence of actions to be performed, the number of actions to the person has to do, the amount of knowledge needed to recognize what to do, and the need of resources needed to establish a decision criteria. They also found that this measure is correlated with the time to develop the tasks. Akbar et al. (2013) presented a simulation tool to forecast hurricanes and water surges in the Gulf of Mexico. Their aim was to aid in decision-making and first response preparation. To validate their tool, they used the Hurricane Katrina as a case study.

Although these are useful methods and tools, none of them tackles how to improve the organizational resilience from the communications and structure point of view. We focus on filling this gap suggesting the use of a methodology to design resilient organizations (the VSM) and providing an architecture and development process to study the resilience of communications using as a case study a real NEP from Spain.

2.2. Organizational Resilience

During the last years, the study of resilience has become more important because people are more aware of the consequences of natural and anthropogenic disasters (Tukamuhabwa et al. 2015). However, some authors (see e.g. (Horne III 1997)) think that the study of resilience is gaining

importance due to the speed of changes in the economy, society, and technology. Because of the speed of changes, survival is now considered a critical aspect; being resilient is important for survival in such a changing environment.

Although there is increasing interest in the topic, there is no agreement about where it was first introduced. Some authors say that it was in Psychology (see e.g. Coutu, 2002). Others say that the concept was popularized after Holling (1973), "Resilience and Stability of Ecological Systems" (see e.g. Henry & Ramirez-Marquez, 2010; Annarelli & Nonino, 2016).

Today, research on resilience is important in many different fields such as Management, Ecology, Psychology, Disaster Management, Organization Management, Sociology, and Engineering. As research in resilience has been attacked in many areas, there is no a widely accepted definition, even in the same area (Bergström et al. 2015).

At first sight, one may think that there is no relation among the different research areas on resilience. For instance, one could believe that resilience against disasters is not related to build resilient systems, organizations or individuals. However, several authors have already identified relationships between the different fields. For example, we need resilient individuals to build resilient organizations (Mallak 1997; Biggs et al. 2012; Doe 1994). Resilient organizations also need resilient supply chains (Sheffi 2007) or resilient infrastructure (Bell 2002; Erol et al. 2009). Resilient organizations contribute to create resilient communities (Kendra & Wachtendorf 2003; Lee et al. 2013) or societies (Beermann 2011) and to develop resilient territories (Gilly et al. 2014). Resilience engineering principles affects the resilience of organizations (Righi et al. 2015).

Figure 1 represents the relationship among these areas centered on their relation to organizational resilience. Organizational Resilience influences the resilience research areas painted in grey color, and it is influenced by research areas depicted in white color. Organizational resilience is influenced by resilient individuals, resilience engineering, infrastructure resilience, cyber resilience, system resilience, supply chain resilience and business resilience. Organizational resilience influences community resilience, societal resilience, economic resilience, city or urban resilience, territory resilience and socio-ecological resilience.

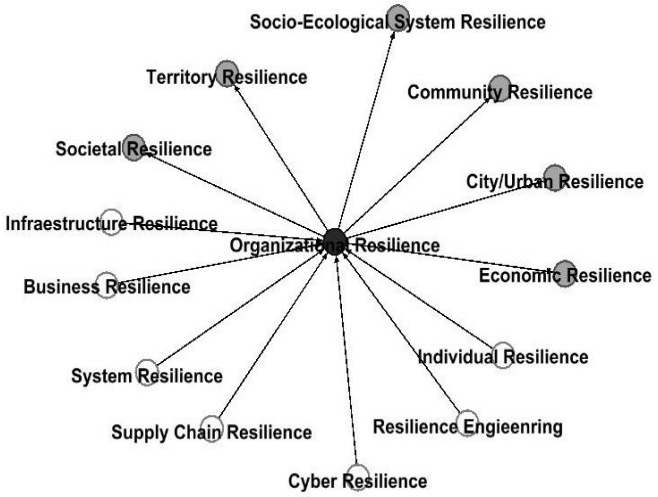


Figure 1. Relations between resilience concepts and organizational resilience. The arrows indicate the direction of the relations.

In this review of the concept of resilience, we focus on resilience at the organizational level since we are interested in the study of organizations in general and emergency plans in particular. Remember, that as we have stated in Section 2.1, emergency plans are a virtual organization designated to solve the emergency. More specifically, we analyze how organizational resilience is conceptualized and how it is measured.

At the organizational level, resilience has simultaneously emerged from different fields such us Enterprise Risk Management, Business Continuity Management, Emergency Management, Crisis Management, Physical Security, and Cyber-Security (Braes & Brooks 2011; Braes & Brooks 2010; Gibson & Tarrant 2010). In these fields, researchers and practitioners have studied how to protect the organizations against disruptive events.

Louisot (2015) considers resilience as a main issue of Risk Management, and Jackson, Firtko and Edenborough (2007) view resilience as a new way of thinking about risk. As systems and organizations cannot be designed to anticipate all possible risks (Fiksel 2003), we need resilient organizations to deal with high consequence and low probability risks events (Dalziell & Mcmanus 2004; Ambulkar et al. 2015) or when policies, procedures, practices, and tools fail during an emergency response (Kendra & Wachtendorf 2003).

Although there are several reviews about resilience at the organizational level (Bhamra et al. 2011; Annarelli & Nonino 2016; Linnenluecke 2017; Bhamra et al. 2015) the research questions we discussed earlier are not yet answered. Bhamra et al. (2011) introduced a general review about resilience based on 74 papers published before 2011. They identified five perspectives for resilience studies (ecological, individual, socio-ecological or community, organizational and supply chain). They focused on the conceptualization of resilience based on these perspectives. However, only three of the definitions they presented were in the context of resilience of organizations. Bhamra et al. (2015) presented an updated version of Bhamra et al. (2011) including 100 articles and five definitions valid for organizations. Annarelli & Nonino (2016) investigated the research domains of organizational resilience based on a literature review and co-citation analysis. They aimed to understand the actual state of development of organizational resilience and the future research directions in this area. They also reviewed several definitions of resilience and organizational resilience, and proposed a new one. However, they did not analyze what are the differences in the conceptualization of organizational resilience. Linnenluecke (2017) focused on the evolution of organizational resilience theory. She acknowledged that there is no unified theory and proposed several future research questions such as "What capacities bring about resilience really?" or "How resilience can/should be operationalized?"

Although these reviews have dealt with many important issues, there are still many open questions regarding the conceptualization and elements that contribute to resilience and how it is assessed.

In the rest of the section, we summarize the review of the conceptualization and assessment of organizational resilience. This review and the discussion about organization resilience presented in Chapter 4 has been published in (Ruiz-Martin et al. 2018).

2.2.1. How is resilience conceptualized at the organizational level?

We have reviewed over 50 definitions about resilience at the organizational level, therefore called organizational resilience. This review indicates that, although there seems to be a common core

understanding about what organizational resilience means, there are relevant issues to be discussed as we explain below.

There are three main streams in the conceptualization of resilience: (1) resilience as a feature of an organization (i.e., something that an organization has), (2) resilience as an outcome of the organization's activities (i.e., something that an organization does); and (3) resilience as a measure of the disturbances that an organization can tolerate.

We found that all of them have the same basic meaning: they have an emphasis either on the organization survival, or in dealing with jolts, risks or changes. However, there is no consensus about the following issues: (1) if the risks are only related to threats or also to opportunities, (2) what survival means, (3) if the risks are already known by the organization or not, and (4) if resilience is always a desirable property.

Resilience as a characteristic or set of characteristics from an organization

As we show in Table 1, most authors understand organizational resilience as an ability to deal with internal and external changes, risks or jolts. Others define it as a capacity to deal with them. Finally, some others define it as a capability to deal with these issues. Ability, capacity, and capability have different connotations; however, the authors of these papers do not clarify why they choose one term or the other. Likewise, they do not define ability, capability or capacity. These three words are sometimes used as synonyms to refer to the power to perform an action or a task. Therefore, we will assume that these terms are interchangeable in the definitions.

Table 1. Organizational resilience conceptualization as capacity, ability or capability

Approach	Authors
Resilience as an ability to deal with internal and external changes, risks or jolts	(Horne III 1997; Mallak 1997; Mallak 1998; Hamel & Valikangas 2003; Freeman et al. 2003; Starr et al. 2003; Sheffi & Rice Jr. 2005; Jackson 2007; Grøtan & Asbjørnslett 2007; Bhamidipaty et al. 2007; Milanzi & Weeks 2014; Mafabi et al. 2015; Alblas & Jayaram 2015; Chand & Loosemore 2016; Hu et al. 2008; Tillement et al. 2009; Danes et al. 2009; Hollnagel 2010; Ates & Bititci 2011; Lengnick-Hall et al. 2011; Demmer et al. 2011; Acquaah et al. 2011; Bauernhansl et al. 2012; Tse et al. 2012; Winston 2014; Jaaron & Backhouse 2014; Lengnick-Hall & Beck 2005)
Resilience as a capacity to deal with internal and external changes, risks or jolts	(Tierney 2003; Fiksel 2006; Manyena 2006; Stewart & O'Donnell 2007; Powley 2009; Dewald & Bowen 2010; Proper & Pienaar 2011; Linnenluecke et al. 2012; Gilly et al. 2014; Ortiz-de-Mandojana & Bansal 2015; Alexiou 2014)
Resilience as a capability to deal with internal and external changes, risks or jolts	(Bell 2002; Reinmoeller & Van Baardwijk 2005; Zhang & Van Luttervelt 2011; Kamalahmadi & Parast 2016; Annarelli & Nonino 2016; Robb 2000)

Without specifying why they use ability, capacity or capability, some authors combine these terms with a specific adjective to define resilience. For instance, Manyena (2006) and Hollnagel (2010) consider resilience as something intrinsic to the organization. Powley (2009) defines resilience as a

latent capacity. Gilly et al. (2014) state that the resilience of an organization is both an active and a reactive capacity. Resilience can be also considered something dynamic (Alexiou 2014; Kamalahmadi & Parast 2016) or incremental (Ortiz-de-Mandojana & Bansal 2015).

A small group of authors (Hilton et al. 2012; Burnard & Bhamra 2011) considers resilience as an emergent property the organization exhibits when it encounters setbacks. Others consider resilience as a process to recover from a disruption (van Breda 2016). Horne III & Orr (1998) understand resilience as a quality to respond to significant change. Other researchers (McManus et al. 2008; Erol et al. 2009; Gunasekaran et al. 2011) define organizational resilience as a function of specific capabilities or abilities. For instance, (McManus et al. 2008) define resilience as a function of three abilities or capabilities: adaptive capacity, situation awareness, and management of keystone vulnerabilities. Erol et al. (2009) include enterprise flexibility, adaptability, agility, and efficiency as attributes for enterprise resilience. Gunasekaran et al. (2011) include adaptability, responsiveness, sustainability, and competitiveness. The essence behind these capabilities is the same: dealing with change, environmental jolts or risks. Defining resilience as a function of characteristics indicates that resilience is a complex concept.

Resilience as an outcome of an organization

Other authors, instead of defining resilience with the focus on what a resilient organization has, define resilience with the focus on what a resilient organization does. For instance, resilience is defined as *"the maintenance of positive adjustment under challenging conditions such that the organization emerges from those conditions strengthened and more resourceful"* (Sutcliffe & Vogus 2003; Vogus & Sutcliffe 2007). A resilient organization can return to its performance level after a disruption (Sheffi 2007). It is able to achieve its objectives and realized opportunities in face of predicted or unpredicted disruptive events (Whitehorn 2010; Hilton et al. 2012; Wright et al. 2012).

Resilience as a measure of the disturbance that an organization can tolerate

A small group of authors defines resilience as a magnitude. Under this view, resilience is the amount of disturbance an organization can tolerate and still survive (Linnenluecke & Griffiths 2010; Mamouni Limnios et al. 2014)

2.2.2. Open issues in resilience conceptualization

Many authors consider resilience a property related to events that may have a negative impact in the organizations. For example, resilience is related to surviving or adapting to disruptions (Horne III & Orr 1998; Bell 2002; Sheffi & Rice Jr. 2005; Hu et al. 2008; Lengnick-Hall et al. 2011), disasters or catastrophic events (Tierney 2003); (Alblas & Jayaram 2015), challenging conditions (Sutcliffe & Vogus 2003; Vogus & Sutcliffe 2007), disturbances (Tillement et al. 2009; Linnenluecke & Griffiths 2010; Hollnagel 2010; Mamouni Limnios et al. 2014), threats (Bhamidipaty et al. 2007; Dewald & Bowen 2010) or changes (Fiksel 2006; Grøtan & Asbjørnslett 2007; Stewart & O'Donnell 2007; Milanzi & Weeks 2014; Mafabi et al. 2015). However, a small group of authors considers that these changes or disturbances can also be opportunities (i.e. positive risk such as an increase of the demand) (Bhamidipaty et al. 2007; Dewald & Bowen 2010; Ates & Bititci 2011), and resilient organizations take advantage of these opportunities.

Regarding the discussion about what is the meaning of "surviving" in the context of resilience, some conceptualizations state that an organization is resilient if it bounces back to a prior point of stability (Freeman et al. 2003; Sheffi 2007). Others acknowledge that an organization is resilient if it returns to the same point or if it achieves another state of stability (i.e., it changes, while minimizing the effects due to changes and hazards) (Burnard & Bhamra 2011; Acquaah et al. 2011; Demmer et al. 2011). Some authors consider that a resilient organization can also bounce forward, grow or become stronger (Bell 2002; Fiksel 2006; Vogus & Sutcliffe 2007). Woods (2015) identifies four meanings of resilience that bring four interpretations of "surviving". These four streams are using resilience as rebound (i.e. returning to previous or normal activities after a disruption), robustness (i.e. absorbing disturbances), graceful extensibility (i.e. how to extend adaptive capacity in the face of disruptions) and sustaining adaptability (i.e. the ability to adapt to future disruptions as the conditions change and evolve). These four meanings can be understood as different forms of survival.

Many authors do not define the type of disruptions that resilient organizations are prepared to deal with. Others state that the disruption or change is turbulent (i.e. it happens very quickly compared to the normal adaptation time) (Fiksel 2006; Ates & Bititci 2011; Burnard & Bhamra 2011; Bauernhansl et al. 2012). Others consider that resilience refers to both expected and unexpected events (Hollnagel 2010; Hilton et al. 2012; Wright et al. 2012).

In most of the research works, resilience is considered as a desirable ability or capability for the organizations. Although this is not specifically stated in the definitions, it can be inferred. However, a few of them consider that resilience is not always desirable, depending on the state of the system or organization (Mamouni Limnios 2011; Mamouni Limnios et al. 2014). For example, in a Cournot duopoly (an economic model where the companies compete on the amount they produce and the production decision is made independently of each other and at the same time), after an increase in the production cost for both firms in the same amount, companies are not willing to exhibit resilience (understood as bouncing back to the previous state of cost) (Lambertini & Marattin 2016). The reason is that the new equilibrium in the market may satisfy both companies and they will not be willing to invest money to return to the previous level of costs.

2.2.3. Resilience and related concepts: fragile, robust and antifragile

To clarify the divergences we presented in section 2.2.2, we need to analyze the concepts related to resilience. Resilience is related to fragility, robustness, and antifragility. The concept fragility is related to how a system is broken or damaged in case of variability (Taleb 2012; Taleb & Douady 2013). Robustness is the capacity of a system to absorb disturbances (Woods 2015). Antifragility is a new concept introduced by Taleb (2012), which is defined as the property of a system that, when facing challenges such as failures or volatility, it improves. He differentiates fragile, robust/resilient and antifragile entities, although he uses indistinctly the words resilient and robust.

Woods (2015) pays attention to the difference between robustness and resilience. Being different, using them indistinctly creates confusion when studying resilience. A robust organization absorbs disturbances, but it does not necessarily recover in case of disruptions. Read (2005) provides an illustrative example comparing trees. In case of wind, both a palm and a sycamore tree moves from their equilibrium position. When both trees are exposed to the same wind intensity, the sycamore tree movements are much smaller than the palm tree. Therefore, it is more robust. However, the palm tree

is more resilient as it is able to recover easier from bigger disturbances (i.e., the sycamore tree will probably break).

By focusing on the type of disruption that the resilient organizations are prepared to face, these organizations should be able to survive to both known and unknown disturbances. A robust organization is designed to cope and absorb a set of known disturbances. Therefore, a resilient organization is more prepared to survive than a robust one. Following this view of resilience, we consider it as a desirable property in any organization although in section 2.2.2., we show an example (Cournot duopoly) where an organization is not willing to exhibit resilience.

Being resilient is not only related to bouncing back to the same previous point of stability; being resilient is also achieving another desirable point of stability. If this new point is better than the previous one, and the organization is stronger, we consider that this organization is not only resilient but also *antifragile*. The distinction between *resilient* and *antifragile* organizations clarifies the open question about if resilient organization responds just to threats or also to opportunities. If the organization is able to recover or survive to threats, it is resilient. If this same organization takes advantage of the threats and opportunities to become stronger, it is resilient and antifragile.

2.2.4. Organizational resilience and its attributes

As discussed in section.2.2.1, resilience is a complex and dynamic concept. Complex concepts are characterized by different elements or attributes (Suddaby 2010). To identify these elements, we analyzed over 110 works that tackle the different models and frameworks proposed to build or improve organizational resilience. This review revealed that there is a great variety regarding to the factors and mechanisms that contribute to resilience. Sometimes, the authors refer to the same concept with different words. For instance, improvisation (Coutu 2002; Kendra & Wachtendorf 2002), creativity and innovation (Dervitsiotis 2004) are used to refer to bricolage skills; face down reality (Coutu 2002) is used to refer to situation awareness. Despite the different terminology, we also found some common and repeated characteristics or factors that contribute to enhance resilience.

The most cited attributes or elements of a resilient organization are presented Table 2 identifying the authors that defend them. It is necessary to remark that other proposed elements may also be important attributes for organizational resilience. A resilient organization includes a mix of several capabilities and actions to be performed. It is this mix what makes an organization resilient (Gibson & Tarrant 2010).

Table 2. Most cited attributes that contribute to resilience

Attribute	Authors
Building situation awareness	(Coutu 2002; McManus et al. 2008; Afgan 2010; Braes & Brooks 2010)
Managing organization's vulnerabilities	(McManus et al. 2008; Whitehorn 2010; Erol, Sauser, et al. 2010)
Having resources	(Orchiston et al. 2016; Brewton et al. 2010; Crichton et al. 2009; Kendra & Wachtendorf 2002; Mallak 1998; Ates & Bititci 2011; Aleksic et al. 2013)
Improvisation capacity	(Kendra & Wachtendorf 2002; Rerup 2001; Weick 1993; Grøtan

	et al. 2008; Coutu 2002; Mallak 1997)
Ability to anticipate events	(Berman 2009; Rerup 2001; Hardy 2014; Apneseth et al. 2013; Wright et al. 2012)
Agility	(Ismail et al. 2011; Gibson & Tarrant 2010; Starr et al. 2003; Thomas et al. 2016; Megele 2014)
Learning capacity	(Hilton et al. 2012; Zhang & Van Luttervelt 2011; Burnard & Bhamra 2011; Aguirre et al. 2005; Robb 2000)
Collaboration	(Boza & Poler 2013; Winston 2014; Proper & Pienaar 2011; Alonso & Bressan 2015)
Resilient individuals	(Doe 1994; Mallak 1997; Riolli & Savicki 2003)
Flexibility	(Berman 2009; Kendra & Wachtendorf 2002; Proper & Pienaar 2011; Megele 2014; Pal et al. 2014)
Robustness	(Jackson 2007; Pal et al. 2014; Heinicke 2014; Kendra & Wachtendorf 2002; Tierney 2003; Tompkins 2007)
Redundancy	(Chopra & Khanna 2014; Johnsen & Veen 2012; Powley 2009; Tierney 2003; Winston 2014; Hu et al. 2008)

2.2.5. How is resilience assessed in practice at the organizational level?

In this section, we focus on the assessment of organizational resilience, and we study how it can be measured in practice. With this purpose, we reviewed over 30 works that propose tools or methods to assess organizational resilience. The number of articles reviewed is fewer than those reviewed for organizational resilience conceptualization because there are fewer works in the literature in this area. The review of these works indicates a lack of consensus about how to measure organizational resilience.

We can classify these works in the same three streams discussed in section 2.2.1: those assessed using the features of the organization, those assessed on the organizational outcomes, and those based on how the organization recovers from failure.

A) Measurement based on the organizational characteristics

To study how organizational resilience is measured based on the organizational characteristics, we classify the works based on how the problem is assessed: using indicators or other techniques such as Fuzzy Cognitive Maps (FCMs) or assessment of organizational processes.

A.1) Measurement of organizational resilience based on indicators

McManus et al. (2007) and Seville (2009) suggested 23 indicators followed by a description to evaluate four factors (situation awareness, management of keystone vulnerabilities, resilience ethos and adaptive capacity) that contribute to enhance resilience. Whitehorn (2010) defined a subset of 15 indicators among the previous ones. The indicators proposed by Lee et al. (2013) are a subset of the ones proposed by McManus et al. (2007) and Seville (2009). They propose to evaluate each factor using several items (see Lee et al. 2013 for the list of items). They tested the model proposed by McManus et al. (2007). They found that using their sample and scale, the three factors model was not

supported. They proposed a new version with four factors and they propose to evaluate the factors through 73 items. However, their sample data and scale did not support this new model. They finally suggested a model with two factors (adaptive capacity and planning), 13 indicators and 53 items to be evaluated. Whitman et al. (2013) proposed a shorter version of Lee et al (2013) assessment tool. They proposed to use just 13 items, one per indicator. They justified this short version based on two reasons. The first one was the low rate response they got while measuring resilience with the long questionnaire. The second one is the correlation in the results between the two assessment tools. The indicators proposed by Lee et al. (2013) include some of the characteristics for resilient organizations presented in section 2.2.4, such as innovation and creativity (it matches with improvisation capacity), collaboration (it matches with partnerships) or situation monitoring and reporting (it matches with situation awareness and ability to anticipate events).

A different approach comes from Starr et al. (2003), who proposed to assess resilience based on eight points: (1) the organization transparency, (2) the understanding of risk interdependencies, (3) the development of viability studies in the organization, (4) the alignment between the strategy in the organization and the objectives, (5)the organizational knowledge about the efforts being spent on resilience, (6) situation awareness, (7) how the organization uses situation awareness to react in a timely manner and (8) the existence of measures to evaluate resilience and the progress of the organization. However, they did not propose a scale for these eight points. Tompkins (2007) proposed using Robustness, Responsiveness, Resourcefulness, Rapidity and Redundancy (the *Five R's*) to evaluate resilience. However, the items to be evaluated in each category were not discussed. Sanchis & Poler (2013) proposed to measure resilience based on the vulnerability of the organization, its adaptive capacity and recovery ability. Kohno et al. (2012) proposed to evaluate resilience taking into account the areas where the organization's facilities are located, the infrastructure the organization needs, the organization facilities and the supply chains. Apneseth et al. (2013) proposed to assess organizational resilience based on how good the organization is at monitoring, responding, anticipation and learning.

Somers (2009) proposed to measure organizational resilience potential based on the six organizational resilience attributes proposed by Mallak (1998a). These factors can take three levels and the overall resilience of the organization is evaluated from 1 (low resilience) to 7 (high resilience). Hollnagel (2010) proposed to assess resilience based on the ability of the organization to respond, monitor, anticipate and learn. Van Trijp et al. (2012a); Van Trijp et al. (2012b) proposed to evaluate resilience as a function of four factors: situation awareness, management of keystone vulnerabilities, adaptive capacity and quality. To evaluate these factors, they defined a performance measure based on the attributes they depend on. For example, to measure situation awareness, they evaluate the level of awareness about expectations, obligations and limitations, the ability to look forward opportunities and potential crisis, the level of awareness about resource availability, the ability to identify the crisis and their consequences, the understating of the trigging factors for a crisis and the understating of minimum operating requirement for recovery. Rigaud et al. (2013) propose to evaluate resilience based on the organization capacity to (1) respond, (2) monitor short-term developments and threats, (3) anticipate long-term threats and opportunities and (4) learn from past events. They proposed several indicators for each one. However, they did not describe the indicators proposed.

Other research in resilience measurement is focused on specific sectors. For instance, Danes et al. (2009) determined resilience in family firms evaluating the following seven items: (1) Role clarity, (2) Who has the decision authority, (3) Ownership equality, (4) Fairness of compensation, (5) Failure to

resolve firm conflicts, (6) Unfair workloads and (7) Competition for resources between the family and firm. Wicker et al. (2013) developed an organizational resilience scale to measure resilience in sport clubs. They develop items (ranged from 1 to 5) to evaluate each factor of resilience defined by Bruneau et al. (2003): robustness, redundancy, resourcefulness and rapidity. For example, to measure rapidity, they evaluate the capability of the organization to achieve goals in a timely manner, adapt quickly to changing circumstances, meet priorities in a timely manner, restore services quickly during unexpected events and respond quickly to disruptive events.

A.2) Other techniques to measurement organizational resilience

In the same line of analyzing organizational characteristics, other authors use FCMs and Fuzzy sets to analyze these characteristics. For example, Grande & Trucco (2008) proposed to analyze the resilience of an organization using FCMs to capture the relations between the variables that contribute to resilience. To study a Civil Defense System, they proposed to evaluate 17 variables and their relations. Asgary et al. (2009) developed a Fuzzy-JESS Expert System based on 17 variables and a set of rules that takes into account these variables to determine the level of resilience in the business. The variables include existence of a strategic plan, existence of a business continuity committee or number of potential hazards among others.

Aleksić et al. (2013) proposed to assess organization resilience potential of SME using fuzzy sets and evaluating the contributing factors for each business process. The importance of each factor in the process is weighted to calculate the resilience of the process. Then, the importance of each process in the organization is also weighted to measure the overall organizational resilience. They proposed to evaluate internal factors (planning strategies, capability and capacity of internal resources, internal situation monitoring and reporting, human factors and quality), external factors (external situation monitoring and reporting and capability and capacity of external resources) and enabling resilience factors (design of the organization, detection potential, emergency response and safety management system).

Tadić et al. (2014); Macuzić et al. (2016) proposed to evaluate resilience using a fuzzy approach. They propose the following steps: (1) Create an organizational reference model and to identify the factors that contribute to resilience, (2) Weight the importance of these factors and processes using a fuzzy approach, (3) Determine linguistic expressions to evaluate these factors, (4) Calculate the resilience factors' values using a fuzzy approach and (5) Rank the organizational resilience factors.

Hu et al. (2008); Hu et al. (2009); Hu et al. (2010) proposed to solve an optimization problem in a network model of the enterprise to determine the effect of a disruption and the resilience of the enterprise. The objective is to understand the balance between operational redundancy and inventory redundancy to achieve resilience. They do not provide items to be analyzed to evaluate resilience. Caralli et al. (2010) proposed the CERT® Resilience Management Model to assess resilience. It defines 26 process areas with specific goals and practices. These areas include asset definition and management, resilience requirement development, risk management, people management or monitoring. The position of the organization in these processes can be used as benchmark for identifying organizational capability for managing operational resilience.

B) Measurement based on the organizational outcomes

This stream is less popular as fewer authors use this approach. For example, Watanabe et al. (2004) proposed to use the Operating Income to Sales to measure resilience. Dalziell & Mcmanus (2004) proposed to measure resilience based on KPIs defined taken into account the organization's objectives. However, these authors did not state the items, attributes, components or KPIs to be measured. Afgan (2010) proposed an index to measure resilience based on the change of company profit, the change of total company income, the change of product cost and the change of manpower (i.e. human resources availability). Markman & Venzin (2014) proposed to measure resilience based on the Return on Equity (ROE) and volatility. Jackson (2007) proposed to measure resilience potential based on statistical correlation between minor and major incidents. He found that minor accidents are positively correlated to major accidents.

C) Measurement based on the organizational recovery

In the third stream, the authors measure resilience based on how the organization recovers from failure. The drawback of this approach is that the organization needs to suffer failures to assess its resilience. Therefore, this way to measure resilience is only valid after the organization has suffered some shocks. There are two main ways to measure resilience following this approach. Henry & Ramirez-Marquez (2010) propose to measure resilience quantitatively using recovery and loss as follows:

$$Resilience = 100 \cdot \frac{Recovery}{Loss}$$

Where loss is the deterioration from the original state after the disruption and recovery is the amount it bounces back from the disruptive state to the recovered state. The authors acknowledge that the limitation is to not to consider the money and time to recover. They do not consider what we should evaluate to measure loss and recovery. Erol, Henry & Sauser (2010); Erol, Henry, Sauser, et al. (2010) proposed to measure resilience based on recovery time, level of recovery, initial vulnerability and potential loss averted. However, they do not indicate how to assess these items

2.3. Viable organizations and the Viable System Model

The challenge that leaders and managers in organizations face in the current turbulent environment is formidable. The complex environment in which they act demands that managers have access to decision-making tools commensurate with the complexity which they must face (Schwaninger & Pérez Ríos 2008). In relation to this issue of the capacity for handling complexity, it has been pointed out that the quality of decisions made by managers is limited by the quality of the models they use for the systems they try to govern. If we are concerned with the viability of an organization (understood as system – see Beer (1989)), meaning with viability the capacity of a system to maintain a separate existence, (i.e. to survive regardless of changes in its environment), then we can apply an organizational cybernetic approach, in particular the Beer´s Viable System Model (VSM). According to the VSM, a viable organization must have the capacities of self-regulation, learning, adaptation, and evolution.

In his VSM, Beer (1981, 1985) establishes the necessary and sufficient conditions for the viability of an organization. These are related to the existence of a set of functional systems (Beer identified them as System 1, 2, 3/3*, 4 and 5) in an organization and a set of relationships among these functional systems and the environment. These systems and the relations among them are represented in Figure 2.

According to Beer, all viable systems contain viable systems and are themselves contained in viable systems. The most important aspect of this recursive conception of viable systems is that, no matter which place they occupy within the chain of systems, they must always contain the five functional systems that determine viability, in order to be viable.

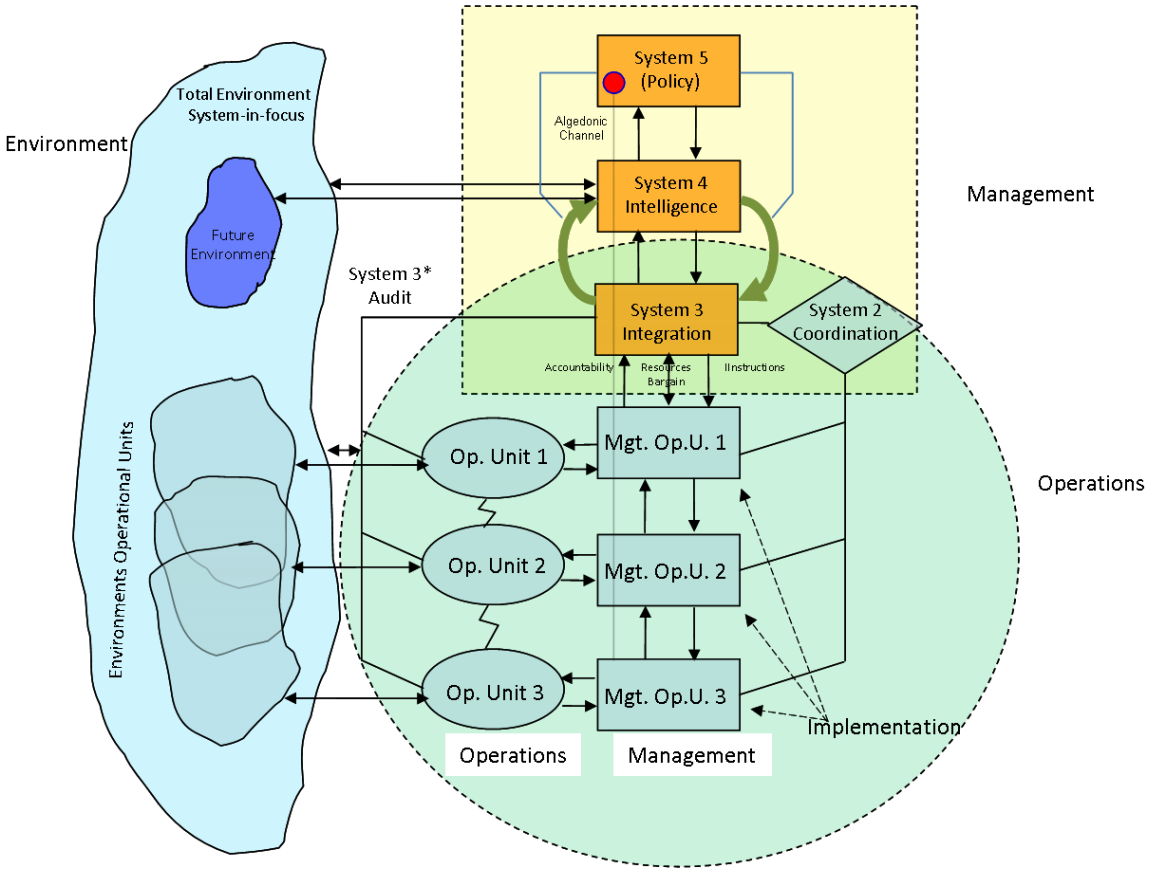


Figure 2. Viable System Model, adapted from Beer, 1981 (Pérez Ríos 2012). Used with permission of the author.

System 1 is responsible for producing and delivering the goods or services which the organization produce. In the example shown in Figure 2, System 1 is made up of three elemental operational units (Op. Unit 1, 2 and 3) which can be divisions of a company, sub organizations, etc. The main role of *System 2* is to guarantee a harmonic functioning of the organizational units, which compose system 1. *System 3* is responsible for optimizing the functioning of the whole set of system 1, made up of the different operational units. We can say that it is responsible for the "here and now" of the organization. The main responsibility of *System 4* is to monitor the environment of the organization. It takes care of the "outside and then" of the organization, with the aim of being prepared for changes. *System 5* takes care of the normative decisions and is responsible for defining the ethos, the vision and the identity of the organization.

2.3.1. Organizational Pathologies according to the VSM

According to Pérez Ríos (2012), any shortage on the systems proposed by Beer (see Figure 2) or in their communication mechanisms is translated into different organizational pathologies. Any organizational pathology may cause the disappearance of the organization, at least as an independent entity.

Pérez Ríos (2012) classifies the organizational pathologies into three main groups: structural pathologies, functional pathologies and information pathologies.

Structural Pathologies

Structural pathologies are related to how the organization is designed and how it copes with environmental variability. There are four structural pathologies: non-existence of vertical unfolding, lack of first recursion levels, lack of middle recursion levels and entangled vertical unfolding with interrelated memberships.

Functional Pathologies

Functional pathologies are related to the adequacy of the organization's systems to the prescriptions made by the VSM. Functional pathologies are classified based on the system they affect and those ones that affect the whole organization.

Functional pathologies related to system 5 are: ill-defined identity, institutional schizophrenia, lack of metasystem (i.e. collapse of system 5 in system 3) and inadequate representation in higher levels. Functional pathologies related to system 4 are headless chicken (i.e. the organization does not monitor the environment and is not able to adapt to changes) and dissociation between system 4 and 3. Functional pathologies related to system 3 are: inadequate management style, schizophrenic system 3, week connection between system 3 and 1 and hypertrophy of system 3. The functional pathology related to system 3* is the lack or insufficient development this system. System 2 can present two pathologies: disjointed behavior within system 1 and authoritarian system 2. The pathology related to system 1 are "autopoietic beasts" (i.e. organizations that only focus on individual goals and do not take into account the whole) and dominance of system 1. The pathologies related to the whole organization are organizational autopoietic beasts and lack of metasystem.

Information System and Communication Channel Pathologies

Information system and communication channel pathologies are related to the malfunctioning of the communication and information system. Information pathologies are the lack of information systems, the fragmentation of information systems and insufficient or lack off key communication and algedonic channels.

Chapter 3. Methodologies Used for the Architecture Design

In this section, we present the three methodologies used in this thesis to build the architecture we use to model and analyze diffusion processes in multiplex networks, being the study of diffusion of information inside an organization a particular case:

- We use ABM to identify the agents involved in the diffusion process and their behavior.
- We use Network Theory to establish the relationships between the agents and to do static analysis.
- We use DEVS to formalize and implement the model in order to study dynamic scenarios that include the behavior of the agents.

In this chapter, we introduce the basic concepts and tools of the methodologies used in the thesis for those readers who are not familiar with them, so they can follow the explanations present in subsequent chapters.

3.1. Network Theory

In Network Theory, a system is modeled as a set of nodes connected by links. The nodes and the links can have different meanings. For example, nodes can be cities and the links roads between them. The nodes can also be people and the links the social relation between them, etc. In Figure 3, we show different types of networks.

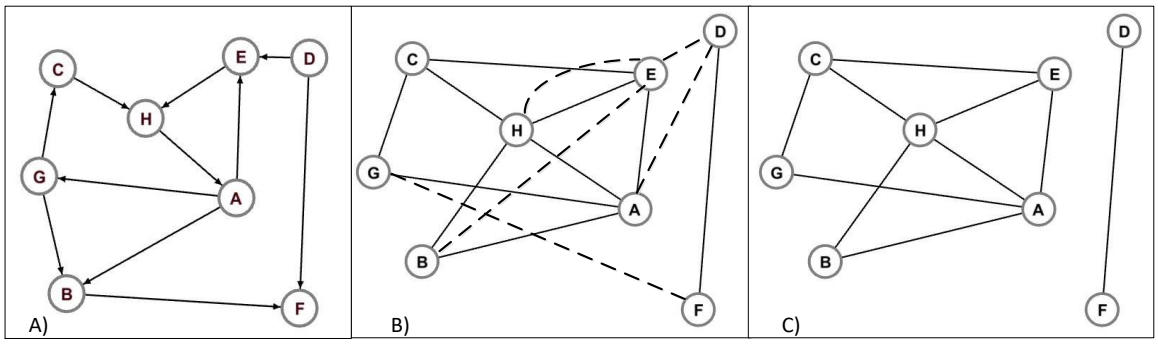


Figure 3 Examples of networks: a) Simplex directed network with eight nodes. b) Multiplex bidirectional network with one component. The different types of lines represent the different layers. c) This network represents the resultant network when the dot-line layer in figure b fails or disappears. After the failure, we get a simplex bidirectional network with two components.

Networks can be classified following different criteria, as follows (these classifications are nonexclusive).

Simplex vs. Multiplex Networks

If we look at the meaning of the links, networks can be simplex (Figure *3*a, c) or multiplex (Figure *3*b). In simplex networks, all the links have the same meaning. For instance, a simplex

network could be the metro system. In this case, the nodes would be the metro stations and the links, the train lines connecting the stations.

In multiplex networks, the links have different meanings. Each type of link is a different layer of the network. For instance, a multiplex network could be the transportation system. In this case, the nodes would be all the stations (metro station, train station, bus stop, airport, etc.). These stations would be connected by different transport means (links): bus, train or plane. Each type of link belongs to a different layer.

Although multiplex networks can include specific properties in each layer, how to define the interconnections between layers and how to define global network metrics is not yet developed (Gómez et al. 2013). However, there have been advances in this area. For example, centrality measures to study multiplex networks have been proposed (Sole-Ribalta et al. 2014). In addition, there is software to visualize multiplex networks (De Domenico et al. 2014). There are also new methods to identify communities in multiplex networks (Kao & Porter 2017). In (Battiston et al. 2017), the authors summarize the advances on the definition of node and edge metrics and mesoscale network properties. They conclude that there are still lot of open problems and many questions to be asked.

Directed vs. Bidirectional vs. Mixed Networks

If we look at the type of the links, they can be Directed (Figure 3a), Bidirectional (Figure 3b, c) or Mixed. In directed networks, all the links are directed. That means that all nodes have a source node and a target node. The opposite connection represents a different relation. It means that the connections are not reciprocal. An example of directed network could be the supervisor-supervisee relationship. In this type of network, as shown in (Figure 3a), an arrow represents the direction of the link. In bidirectional networks the source and target node are interchangeable. There is no specific direction for the link. It means that the relations are reciprocal. An example of this type of network could be the coworker relationship. A Mixed network combines direct and bidirectional links. An example can be the relations in a team inside an organization: we have supervisors and supervisees (direct link) and coworker relations (bidirectional link).

Weighted vs. Non-weighted networks

If we consider the weight of the links, networks can be weighted or non-weighted. The links in non-weighted networks represent the existence or absence of the relation. For example, links in a non-weighted network in the bus system represent if there is connection or not between two stations. Instead, in weighted networks, the links have a weight associated. This weight is a measure of how strong is the relation between the entities. Following the same example, the weights of links in a weighted network in the bus system may represent how many bus lines connect the two stations.

3.1.1. Network Metrics

There are various metrics to characterize and extract properties from a network model. These metrics include the network density, its diameter, the degree distribution, its clustering, modularity or number of connected components among other.

The network density is the number of links in the network divided by the total possible links. The network diameter is the maximum distance over all nodes of the network. The average path length is

the mean of the distances between each pair of nodes along the shortest path. Modularity class indicates how good a partition of the network in communities is. The average clustering coefficient is the mean of all nodes' clustering coefficients. It measures how the nodes tend to cluster. The page rank measures the importance of a node based on its connections (high value means connections to important nodes). HITS algorithm provides two measures about a node: Hub and Authority (recursively based on the out-degree and in-degree of adjacent nodes). Influence domain of a node is the number of nodes that can reach it (directly or indirectly); without loss of generality it can be defined as the number of nodes it can reach. Proximity prestige of a node is calculated dividing influence domain by the average distance from all nodes in its influence domain. In-degree of a node is the number of nodes pointing to it. Out-degree is the number of nodes it points to. Betweenness centrality is a measure about the number of times a node acts as bridge in the shortest path between two other nodes. Assortativity is a measure of correlation between nodes (a positive value indicates correlation between nodes with similar degree, while a negative one indicates correlation between nodes of different degree). A network component consists of all the connected nodes of the network. A network can have a single component (Figure 3b) or multiple ones (Figure 3c).

There is different software available to analyze properties of the network model: Gephi (Bastian et al. 2009), Pajek (Nooy et al. 2005), MuxViz (De Domenico et al. 2014), R (Ihaka & Gentleman 1996), which includes a package for network analysis called igraph (Csardi & Nepusz 2006), etc.

In this thesis, we use Gephi as a supporting tool for the analysis process and to elaborate the figures of the network. Gephi is an open source tool to visualize and calculate different properties of the network such as number of connected components, network density, and network diameter or degree distribution, among others. Gephi also displays different visualizations of the same network based on different algorithms such as communities or node's labels positions. Moreover, Gephi also allows customizing your own view. It also provides filtering features: we can filter in the network based on the nodes, links and global network properties (Bastian et al. 2009).

3.1.2. Applications of Network Theory

Network theory has proven to be a useful methodology to model and study the relations between entities (Newman 2003; Newman et al. 2006). Network Theory provides a set of techniques to study the networks that represent relations between discrete objects. It has been widely applied in different fields where one needs to model systems that have strong interdependence within entities (Strogatz 2001). Some examples in Biology include the modeling of food chains (Dunn & Wilkinson 2015), metabolic networks (Gallos et al. 2007) or brain structure and functions (Bullmore & Sporns 2009). In Medicine, it has been used, for instance, to model the spreading of disease (Chami et al. 2013), drug and vaccine administration (Poland et al. 2013; Ibrahim et al. 2013). It has been used in different technological systems, including power grids (Negeri et al. 2015), transport networks (Zhu & Luo 2016; Deng et al. 2015), communication infrastructure (Peng et al. 2012), and social networks (Liu et al. 2016). Finally, it has been used for Project Management (Fang et al. 2012; Ruiz-Martin & Poza 2015), Supply Chain Networks (Hearnshaw & Wilson 2013), Socio-ecological Systems (Janssen et al. 2006) and Social Economy (Poza et al. 2011; Santos et al. 2012).

In the area of resilience, Network Theory has been applied to study the structure, redundancy and robustness of a water distribution network in order to improve its resilience (Yazdani et al. 2011). It has also been used to find the most critical element in industrial symbiotic networks (Chopra &

Khanna 2014), to identify which communities are more vulnerable in network systems and be able to prioritize resources to protect the critical elements (Rocco S. & Ramirez-Marquez 2011) and to study critical infrastructure connection and topology (Eusgeld et al. 2009; Ouyang 2014). Likewise, it has been applied in analyzing vulnerabilities in process plants in particular cascading effects (Khakzad & Reniers 2015), to improve resilience in communication networks infrastructure (Brinkmeier et al. 2009), and to improve resilience in air traffic management (Cook et al. 2015; Dunn & Wilkinson 2015).

3.2. Agent Based Modeling

Agent Based Modeling (ABM) can be defined as a "computational method that enables a researcher to create, analyze and experiment with models composed of agents that interact within an environment" (Gilbert 2007). An agent is "computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives" (Jennings et al. 1998).

One of the main advantages of ABM is the possibility to establish a direct correspondence between the entities and its interactions in the real system and the agents and its interactions in the models (Edmonds 2001).

One of the disadvantages of ABM is the need of computer simulation to analyze the models. We cannot deal mathematically with most of them (Galán, Izquierdo, et al. 2009). Studying the models computationally has advantages and disadvantages. The advantage is that it is easily observable the emergent behavior of the system. The disadvantage is that when translating the model to a computer program, it is easy to introduce errors in code and artifacts (based on the model assumptions).

3.2.1. Applications of Agent Based Modeling

The segregation model presented by Schelling (1971) can be considered one of the first agent-based models, even though it was not implemented computationally. In this model, the agents are assigned a behavior based on the people who live nearby and the emergent behavior of the system is displayed. Just defining the micro-behavior of the agent, the macro-behavior of the system emerges. Since then, it has been applied in different fields such as Sociology (Lopez-Paredes 2001; Lopez-Paredes et al. 2002; Pavón et al. 2008), Political Sciences (Poza et al. 2011), Economy (Posada & Lopez-Paredes 2008), Anthropology (Angourakis et al. 2015) or Resource Management (Galán, Lopez-Paredes, et al. 2009; Araúzo et al. 2010; Lopez-Paredes & Hernández 2008).

3.3. Discrete Event System Specification (DEVS)

The DEVS formalism (*Discrete Event System Specification*) is a formal discrete-event M&S methodology (Zeigler et al. 2000). It is derived from Systems Theory, and it allows one to define hierarchical modular models that can be easily reused. In DEVS, an atomic model defines the behavior of a component. It is specified as a black box with a state and a duration for that state. When state duration time elapses, an output event is sent, and an internal transition takes place to change the model state. A state can also change when an external event is received. Then, a DEVS model is defined by describing the set of states the model goes through, the internal and external transition functions, the output function and the state duration function. DEVS models can be put together by

linking the outputs of a model to inputs of other models to form coupled models. Models made out of more than one component are called coupled models. We can also link coupled models.

Atomic models define the behavior of the system. The formal definition of an atomic model is as follows:

$$AM = < X, Y, S, ta, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda >$$

Where:

- X is the set of input events.
- Y is the set of output events.
- S is the set of sequential states.
- $ta: S \rightarrow \mathbb{R}_0^+ \cup \infty$ is the time advance function that determines the time until the next internal transition.
- $\delta_{ext}: QxX^b \rightarrow S$ is the external transition function that determines the next state when external events arrive, where $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$ }, e is the elapsed time since the last state transition and X^b is a set of bags over elements in X.
- $\delta_{int}: S \rightarrow S$ is the internal transition function that determines the state transition of the model when the state duration is over and no external event has arrived.
- $\delta_{con}: QxX^b \rightarrow S$ is the confluence transition function that determines the next state when and external events arrive at the same time than an internal transition is triggered.
- $\lambda: S \rightarrow Y^b \cup \emptyset$ is the output function that determines the output of the model based on its current state. Y^b is a set of bags over elements in Y and $\emptyset$ is the empty set.

Coupled models are defined connecting multiple DEVS models (coupled or atomic) linking the models' inputs and outputs. A coupled model is defined as the next 7-tuple:

$$CM = < X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC >$$

Where:

- X: Is the set of input events.
- Y: Is the set of output events.
- D: Is the set of the names of the sub-components.
- $\{M_d\}$: Is the set of sub-components where $d \in D$. Each M_d is a DEVS model (either atomic or coupled)
- EIC: is the set of external input couplings
- EOC: is the set of external output couplings
- IC: is the set of internal couplings

3.3.1. DEVS Simulators

DEVS models can be implemented using any DEVS simulator. There are different DEVS simulators such as CD++ (Wainer 2002), DEVSJava (Sarjoughian & Zeigler 1998), VLE (Quesnel et al. 2009), CDBoost (Vicino 2015; Vicino et al. 2015) , etc. In (Wainer 2009), the author provides a comprehensive list of software available to implement DEVS models.

In this thesis, we use CDBoost (Vicino et al. 2015; Vicino 2015), a fast DEVS simulator based on the CD++ toolkit (Wainer 2002). CDBoost is implemented in C++11 and it is cross-platform. CDBoost provides simple interfaces to the modeler, who can easily transform a DEVS model to a DEVS simulation. At the user level, it has two main classes, one for defining the atomic models and one for defining the coupled models. The class for defining the atomic models provides a constructor where the model parameters can be instantiated and the five DEVS functions: internal, external, confluence, time-advance, and output. The class for defining coupled models takes four parameters: the list of coupled model components, the list of external input couplings, the list of external output couplings and the list of internal couplings.

Figure 4 and Figure 5 show the CDBoost simulator definition to implement DEVS models. Figure 4 shows a template to implement an atomic model, and Figure 5 a template to define coupled models.

```
1 struct AtomicName_defs{ //Input&Output Port declaration
2    struct input_port1 : public in_port< MSGi1> {};
3    struct input_portn : public in_port< MSGin> {};
4    struct output_port1 : public out_port< MSGo1> {};
5    struct output_portn : public out_port< MSGon> {};   };
6
7 template<typename TIME>
8 class AtomicName{
9  using defs=AtomicName_defs;//port definition in context
10  public:
11  struct state_type{ //Define your state variables here   };
12  state_type state;
13  AtomicName() noexcept {//parameters/initial state values}
14
15 //DEVS functions
16 void internal_transition() {
17  //Define internal transition function        }
18 void external_transition(TIME e, typename make_message_bags
19              <input_ports>::type mbs) {
20  //Define your external function here                    }
21 void confluence_transition(TIME e, typename
22          make_message_bags <input_ports>::type mbs) {
23  // confluence function here            }
24 typename make_message_bags<output_ports>::type output() const {
25      // Output function
26    typename make_message_bags<output_ports>::type bags;
27    //Define your output function here. Fill bags
28    return bags;                          }
29 TIME time_advance() const {
30      //Define the time advance function                 }
31 };
```

Figure 4. DEVS atomic model implementation using CDBoost.

As seen in the figure, we first, we declare the model ports as a structure (lines 1-5) and the atomic model as a class (lines 7-31). Each atomic model class has the set of state variables grouped together in a structure (lines 11). It also has a model constructor to instantiate the model parameters and initial values (line 13). We implement all the DEVS functions (internal, external, confluence, output and time advance, in lines 15-31) in C++. The code in bold cannot be modified (it is part of the simulator).

```
1 //*****INSTANTIATE ATOMICS *******//
2 template<typename TIME>
3 class iestream_int : public iestream_input<int,TIME> {
4 public:
5 iestream_int(): iestream_input<int,TIME>
6 ("inputs/test_filterNetworks.txt") {};   };
7 //*****DEFINE COUPLED *******//
8 struct inp_in_1 : public in_port<int>{};
9 struct outp_out_2 : public out_port<double>{};
10 using iports_C1 = std::tuple< inp_in_1 >;
11 using oports_C1 = std::tuple< outp_out_2 >;
12 using submodels_C1=models_tuple<filterNet, iestream_int> ;
13 using eics_C1=tuple<EIC
14      <inp_in_1,iestream_int, iestream_defs::in> >;
15 using eocs_C1 =tuple< EOC
16     < filterNet, filterNet _defs::out, outp_out_2> >;
17 using ics_C1=tuple<IC
18     <iestream_int,iestream_defs::out,
19           filterNet, fiterNet _defs::in> >;
20
21 using C1=coupled_model <TIME,iports_C1,oports_C1,
22           submodels_C1,eics_C1,eocs_C1,ics_C1>;
23
24 int main(){ //Call the simulator
25   runner<NameOfTimeClass, NameOfTopModel, logger_top> r{0};
26   r.runUntil(300000); }
```

Figure 5. DEVS coupled and top model implementation using CDBoost.

The coupled models are implemented using the template provided in Figure 5. We instantiate all the atomic models with their parameters (lines 1-6) and then we define the coupled models (including the top model).

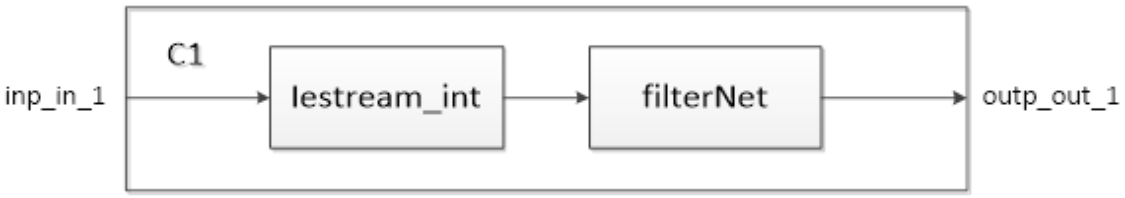


Figure 6. Example of DEVS coupled model defined in figure 2

Figure 5 is an implementation of the coupled model shown in Figure 6. We first declare the coupled model ports (lines 8-9). We then define the top model components: input ports (line 10), output ports (line 11), submodels (line 12), external input couplings (line 13-14), external output couplings (line 15-16) and internal couplings (line 17-19). The coupled model (line 21-22) is defined as a tuple of all these components. The last step is to call the simulator (lines 24-26). We set the name of the time class and the top model name (line 25), and simulation running time (line 26).

3.3.2. Advantages of DEVS

The use of DEVS provides several advantages in the field of modeling and simulation. It is a methodology to develop hierarchical models in a modular way. This modularity allows model reuse and thus, reducing development time and testing. The model definition, implementation, and simulation are separated. The same model can be implemented on different platforms facilitating the

reliability of models and results. Moreover, the simulation algorithm for DEVS models is already verified and validated.

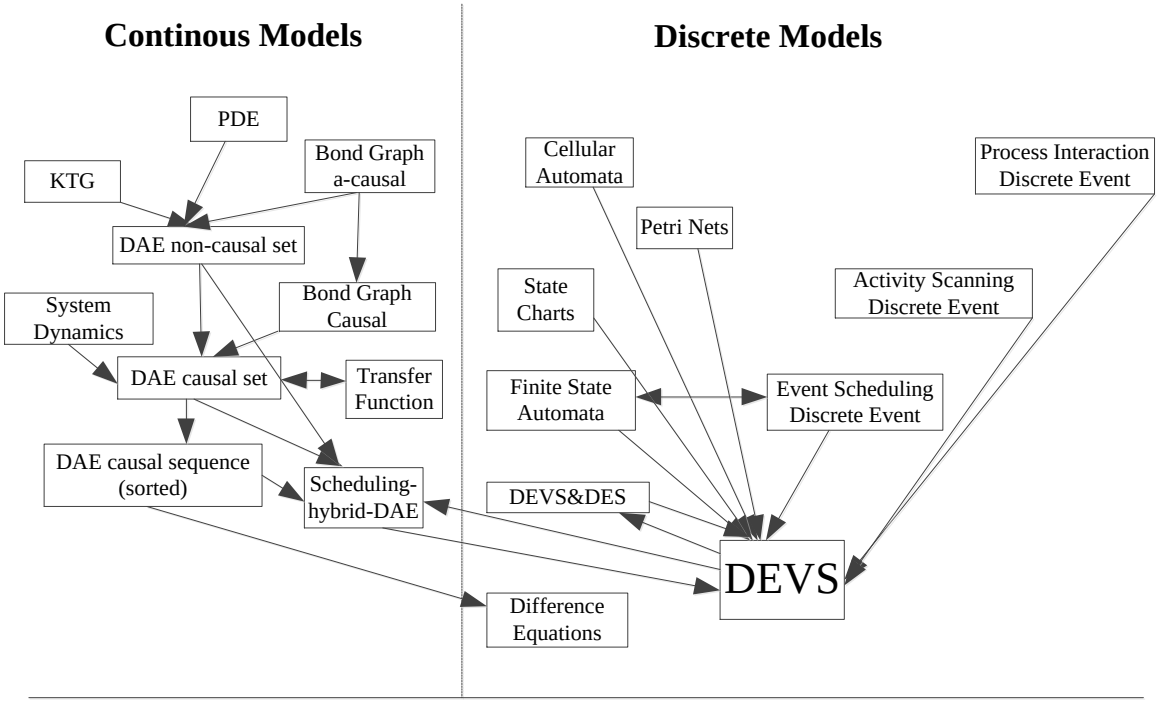


Figure 7 Formalism Transformation Graph. Adapted from (Vangheluwe 2000)

Although there are multiple formalisms for modeling and simulation, we have chosen DEVS for the work in this thesis because it is a common denominator for many other formalism (Vangheluwe 2000).

In Figure 7, we show a Formalism Transformation Graph where the formalisms are represented as nodes and the links denotes existing formalism transformation (i.e. mapping of source formalisms into destination formalisms without modifying the behavior). As we can see in the figure, many formalisms from both continuous and discrete worlds can be mapped into DEVS.

Chapter 4. Viable and Resilient Organizations. The Application of the Viable System Model

In this chapter, we discuss the review about organizational resilience presented in Chapter 2. Based on the literature review, we provide a conceptualization of resilience that integrates the three streams previously discussed and proposed a maturity model for organizational resilience. We also discuss that resilience measurement is still an open research field, and we discuss two streams to measure it: one before and after the disaster occurs.

Then, we relate viable and resilient organizations. Based on this relation, we propose the VSM to design resilient organizations.

4.1. Discussion about Organizational Resilience

In this section, we discuss the review about organizational resilience conceptualization and assessment. We propose a conceptualization of resilience following the indications provided by Suddaby (2010) to construct clarity in Theories of Management and Organization. We also present a Maturity Model for Organizational Resilience (MMOR). Finally, we present the basic dimensions to measure organizational resilience.

4.1.1. Discussion about organizational resilience conceptualization

After the review of organizational resilience conceptualization and following the indications for developing a clear conceptualization defined in Suddaby (2010), we can see that there is not a clear conceptualization of organizational resilience. A clear conceptualization should have a good definition, scope conditions or contextual circumstances, semantic relations with other concepts, and coherence and logical consistency.

We propose a conceptualization that integrates the three views presented in section 2.2.1: ***Resilience, at the organizational level, is the measurable combination of characteristics, abilities, capacities or capabilities that allows an organization to withstand known and unknown disturbances and still survive.***

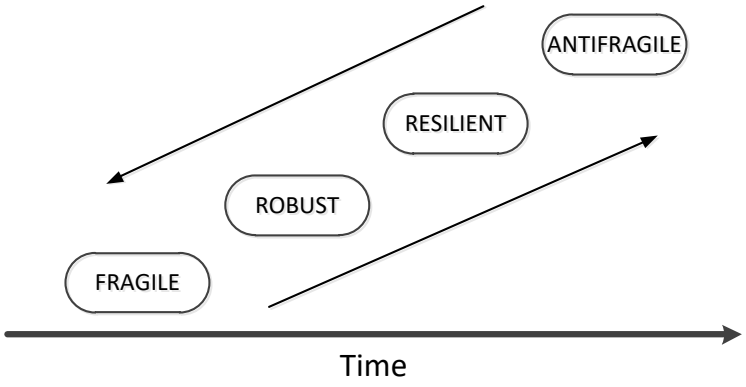


Figure 8 Four-level Maturity Model for Organizational Resilience (MMOR)

Resilience is not a static concept. The degree of resilience an organization has evolves over time. An organization evolves from fragile to antifragile, resilience is a middle estate in this evolution (Taleb 2012). Focusing on how well the organization has developed its abilities to survive to changing or turbulent environments, we propose a four-level Maturity Model for Organizational Resilience –MMOR- (Figure 8). The organization can be at any of the following levels: fragile, robust, resilient and antifragile.

The organization evolves from one level to another over time based on the improvement of its abilities, characteristics or capabilities to deal with disturbances. A fragile organization is not able to withstand with changing environments: it collapses. A robust organization is able to survive to some set of changes in the environment. However, if these changes are outside the designed parameters, the organization will probably collapse. A resilient organization is not only robust, but it is also able to survive to unforeseen events. An antifragile organization is able to not only to survive, but also to prosper or thrive from turbulent environments.

The reverse process (i.e. fall backward from antifragile to fragile) is also possible. For example, an organization can be in the resilient level and changes either in the environment (e.g. new risks that the organization cannot deal with) or inside the organization (e.g. changes in the organizational structure or staff) can cause the organization fall backward.

Regarding to the attributes, elements or characteristics for resilience, we propose the ones presented in Table 2 as an initial combination: building situation awareness, managing organization's vulnerabilities, having resources, improvisation capacity, ability to anticipate events, agility, learning capacity, collaboration, resilient individuals, flexibility, robustness and redundancy.

4.1.2. Discussion about organizational resilience assessment

Through our review, we identified that the works that focus on the measurement of organizational resilience can be classified in the same three main streams as the definitions: (1) assessment of the organizational characteristics, (2) assessment of the organizational outcomes and (3) measure the failure recovery. We have found that there is not a consensus in the way the authors assess organizational resilience, neither inside of these three streams. Moreover, measuring organizational resilience based on how the organization recovers from failure has a drawback: we need the organization to fail to measure the level of resilience.

We consider that two main dimensions to evaluate organizational resilience should coexist. The first one should aim to provide an estimate of organizational resilience potential (i.e. evaluate resilience before a disruptive event occurs). The second one should aim to evaluate the level of resilience an organization has exhibited after a disruptive event has occurred.

To provide an estimate of organizational resilience potential, we propose to follow the first stream: assesses organizational resilience based on the organizational characteristics. The organizational attributes or characteristics to be evaluated should include, at least, the ones presented in Table 2

To measure the level of resilience an organization has exhibit after a disruptive event, we recommend following the third stream: assess organizational resilience based on how the organization recovers from failure. We propose to evaluate a recovery ratio that measures the organizational loses

against the recovery and the recovery time. The recovery ratio should include both organizational capabilities and organizational performance. Measuring resilience after a disruptive event has occurred will help to provide better estimates of the resilience potential studying the correction between the two measures.

Although there are different works that aim to measure resilience, there is not still a quantitative measure that allows us to compare two organizations and conclude which one is more resilient. The measurement of resilience is still an open research field. We want to clarify that we are not focused on providing the measurement scale, just set the basis to develop it.

4.1.3. Further discussion about organizational resilience

As discussed in Chapter 2, the study of resilience covers different related areas. Future research directions should aim to identify these relations. Some questions to be answered are: (1) What is the lowest level of resilience? Is it having resilient individuals? (2) What kinds of resilience (i.e. infrastructure resilience, resilient individual, etc.) affect organizational resilience and how? (3) What kind of resilience (i.e. community resilience, city resilience, and so on) are influenced by organizational resilience? and (4) How all these areas of resilience are integrated to develop a more resilient world?

In section 4.1.1, we introduced the MMOR model. Future research direction should also aim to investigate if this concept and the four-level MMOR to develop antifragile organizations (i.e. from fragile to antifragile organizations) are also applicable to the other concepts such as infrastructures, individuals, communities or territories.

4.2. Relationship between Viable and Resilient Organizations

A review of several definitions of resilience have pointed out that, among other characteristics, resilient organizations have to recover from challenges or disruptive events (i.e. survive) (Sheffi & Rice Jr. 2005; Fiksel 2006; Manyena 2006; Stewart & O'Donnell 2007; Hollnagel 2010; Annarelli & Nonino 2016). Viability is the capacity of an organism to maintain its separate existence (i.e. ability to survive despite changes in the environment). Therefore, a resilient organization has to be a viable one.

As we have already mentioned, the VSM establishes the necessary and sufficient conditions for the viability of an organization (Beer 1979; Beer 1981; Beer 1985; Beer 1989). According to the VSM a viable organization must have the capacities of self-regulation, learning, adaptation, and evolution.

These capacities, among others, are within the set of factors that contribute to enhance organizational resilience or, within the set of characteristics and properties a resilient organization should have. For example, McManus et al. (2008); Van Trijp et al. (2012) or Jackson (2007) consider adaptability as an attribute that a resilient organization should have.

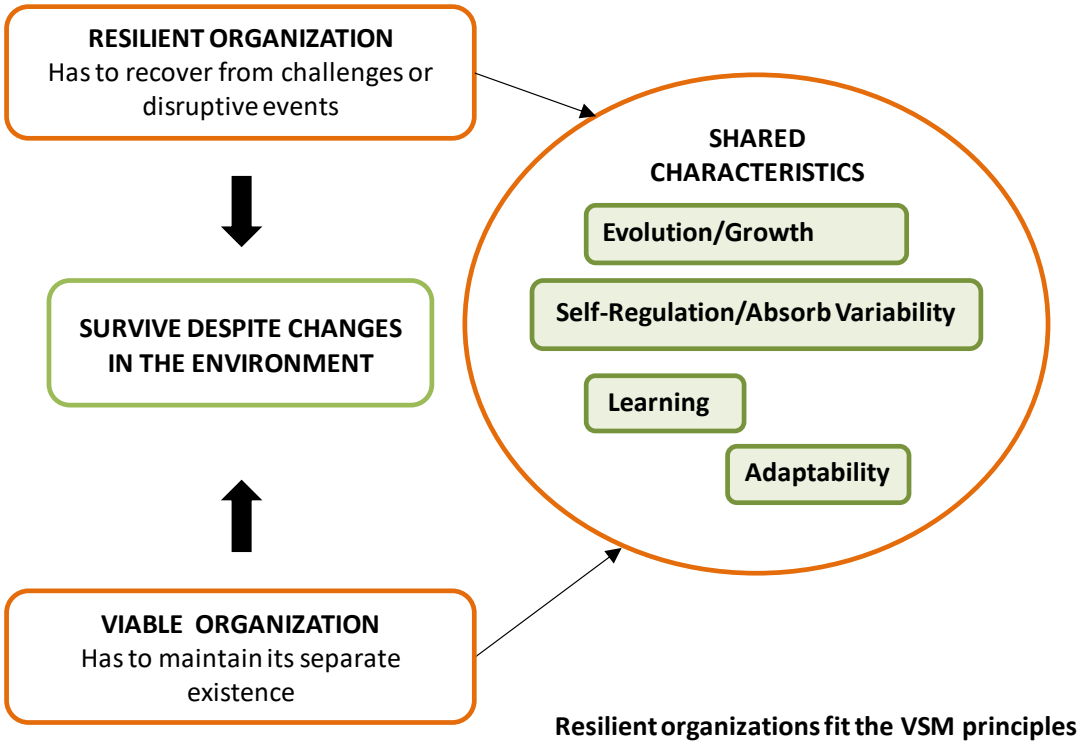


Figure 9. Shared characteristics between resilient and viable organizations

Learning (Stewart & O'Donnell 2007; Robb 2000; Zhang & Van Luttervelt 2011; Hilton et al. 2012; Alexiou 2014) and evolution (Demmer et al. 2011) are also included among the factors and characteristics of resilient organizations. Other authors (Fiksel 2006; Proper & Pienaar 2011) do not explicitly talk about evolution, but they include growth (which can be understood as evolution) among the characteristics of resilient organizations. Self-regulation, understood as absorbing environment variability (i.e. auto adaptation to changes in the environment), is also included among the characteristics of resilient organizations (Linnenluecke & Griffiths 2010; Jaaron & Backhouse 2014). This explanation is summarized in Figure 9.

Following this analysis and based on the shared characteristics of resilient and viable organizations, we conclude that resilient organizations fit the VSM principles. Therefore, the systemic methodology introduced by Pérez Ríos (2010) is valid and appropriate to design a resilient organization. We explain the application of this methodology in the next section.

4.3. A Methodology to Design Resilient Organizations

Based on Organizational Cybernetics (OC) and, in particular, the VSM's conceptual elements, Pérez Ríos, (2010) introduced a systemic methodology to help design or diagnose systems in view of their viability. Based on the commonalities of resilient and viable organizations and taking into account that we have justify that resilient organization has to also be viable, we propose to use this guide to design resilient organizations.

The process to apply it is structured in four main steps as we show in Figure 10.

The first step is to identify the identity and the purpose of the organization. In this process, we will try to assess what the organization is (and what the organization is not) and what it is, or should be, its purpose.

In a second step, we see how the organization faces the total environment complexity (variety) by means of creating a vertical structure made up of sub-organizations where each of them will be in charge of the different sub-environments in which the total environment is also divided. For this purpose, we use the VSM.

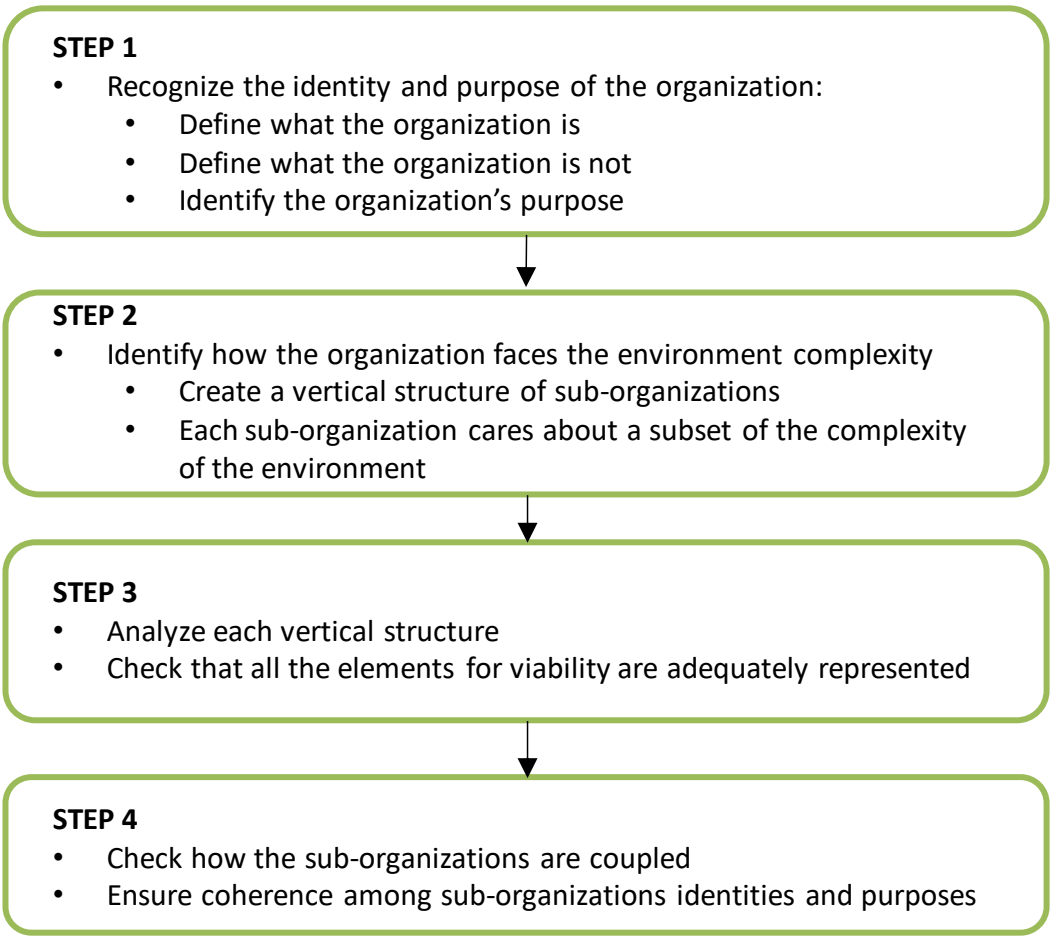


Figure 10 Process to design and diagnose systems in view of their viability

In a third step, we should go through each of those vertical levels and get into them to check that all the necessary and sufficient elements for viability, which OC and the VSM identifies. We need to check that they are adequately represented in all the organizations, sub-organizations, sub-sub-organizations, etc. in which we have unfolded the initial organization.

The fourth and last step would be to check the degree of coupling of all organizations, sub-organizations etc. at all recursion levels, from the point of view of the coherence among their respective identities and purposes.

We want to highlight that any shortage in the five systems presented in Figure 2 or in its functions due to absence, to malfunction or to deficient design of the communication channels that connect them

carries pathologies in the organization. These pathologies, as explained in section 2.3.1 cause that the organization does not work properly or even disappear, at least as an independent entity. Identifying this pathologies and tackling them will make the organization viable and therefore improve its resilience.

Based on this approach we consider that an organization is more luckily to be resilient if it does not present any pathology.

One of the pathologies presented in section 2.3.1 is related to the information system and communication channels inside the organization. In rest of this thesis, we will focus on providing a solution to study the communications inside an organization taking into account the behavior of the people involved in them. As we have already mentioned, the solution we provide is a simulation model designed based on an architecture to simulate diffusion processes in multiplex networks. Using this simulation model we can study the communication inside the organization taking into account different factors such as the behavior of the people or the reliability of the communication channels.

Chapter 5. An Architecture to Study Diffusion Processes in Multiplex Dynamic Networks: Preliminaries

The aim of this thesis is to provide a Framework to study the resilience of organizations using formal methods. For this purpose, we will use as a case study a Spanish NEP (see details in Chapter 7)

The structure of an organization and the communications among its participants can be modeled as a problem of information diffusion in multiplex networks (represented as directed graphs). The idea is to build a network in which the nodes in the graph represent the individuals in the organization, and the links between them represent the communication relations.

Initially, we collected information from experts in the Spanish NEP. We studied the emergency plan using ABM. We identified the agents involved in the plan, its organizational structure, their communication mechanisms (distinguishing the different technologies used by the agents to communicate with each other), and the messages and actions that the agents take before (preventive), during (control and mitigation) and after (recovery) the emergency (Ruiz-Martin 2013). In this process, we realized that these agents were related with each other creating a network, therefore we decided to use Network Theory (Newman 2003) to create a network model and we implemented our case study using Gephi (Bastian et al. 2009). We use the network model to study the characteristics and properties of the communication and the command chain network inside the NEP (Ruiz-Martin, Ramírez Ferrero, et al. 2015). We also used the model, as we detail in Chapter 9, to study how a downfall in the different technologies affects the robustness of the communication structure (Ruiz-Martin, Lopez-Paredes, et al. 2015). These analyses provided important results to understand how the emergency plan works and to propose improvements in terms of its communication structure.

However, using this methodological approach has some limitations. We needed to assume that all communication technologies are equivalent, and we needed to study our network as a simplex one. We made this assumption because although multiplex networks can include specific properties in each layer, characterizing the interconnections between layers and the appropriate global network description measures is still challenging (Gómez et al. 2013). Moreover, in order to study our network, we needed to consider it as a static model, not being able to change the scenario dynamically (we constructed the network for each scenario, and studied it as a simplex static one).

5.1. The Study of Diffusion Processes in Multiplex Networks

In recent years, different approaches have been used to study diffusion processes in multiplex networks. Much of the research work is based on the definition of algorithms to simulate different diffusion processes. For example, in medicine, different methods have been created to study the relationship between information dissemination and disease spreading. Wang et al. (2016) proposed an algorithm to study how the diffusion of preventive measures to protect the population against a disease affects the disease dissemination. Granell et al. (2013) worked on the same topic using Microscopic

Markov Chains. Other authors focused on algorithms to study the propagation of specific diseases such as dementia (Raj et al. 2012).

Some research focuses on the application of algorithms and models originally designed to study the spread of diseases for studying other problems. For example, Khelil et al. (2002) applied an epidemiological diffusion model to study diffusion processes in mobile ad hoc networks. Estrada & Gómez-Gardeñes (2014) propose a communicability function to analyze the flow of information in multiplex networks. Some of the works that studied information dissemination processes focus on social networks. Several diffusion models for contagious processes such as opinion adoption, social movements or behavior modification were proposed (Yağan & Gligor 2012; Cozzo et al. 2013).

Some of these diffusion algorithms are parameterized. A correct estimation of the parameters is key for driving conclusion of the studied problem. Saito et al. (2009) focused on the estimation of the parameters for a "continuous time delay independent cascade model" to study information diffusion. The data for the estimation is obtained from observations of diffusion information data.

Another line of research in the study of diffusion processes in multiplex networks is based on the application of ABM. Jiang & Jiang (2015) matched the elements considered on diffusion processes in social networks with the concepts of ABM. They proposed that ABM could be used to study the diffusion problem in social networks in two ways. The first one is an alternative method to the theoretical perspective to get empirical results. The second one is a complementary method to connect theoretical research to the empirical one. Following this line, Xiong et al. (2015) studied the effect of the diffusion of innovation in social networks using numerical simulation. However, they do not clarify the implementation platform.

Bouanan et al. (2016) introduced an architecture to simulate information diffusion processes in social networks using DEVS, ABM and Network Theory. Their proposal is to represent multiplex networks based on a Server-Proxy architecture. The servers represent the behavior of the nodes about the information received, while the proxies represent the diffusion rules for each specific layer. Both the servers and proxies are modeled using DEVS. Then, the server and proxy are coupled in a DEVS model that represents a network node. The connections are used to build the Top Model, a multiplex network implemented in DEVS that connects the DEVS network nodes. To model dynamic networks, they store all possible network configurations and they use Dynamic DEVS (DS-DEVS) (Barros 1997) for modeling and simulation.

Through a research collaboration with Bouanan, we tested the applicability of their architecture to simulate information diffusion processes in multiplex networks. We found that it did not allow us to include all the attributes specified in our case study. Therefore, we needed to define a new one that (see Chapter 6).

In the next sections, we detail this preliminary work, that was published in (Ruiz-Martin et al. 2016).

5.2. A Server-Proxy Hybrid Architecture to Model Information Diffusion Processes

Formal methods, models, and tools for social data are largely limited to graph theoretical approaches informed by conceptual developments in social network analysis. Bouanan et al. (2016) have provided an integrated modeling approach to social data across the conceptual, formal and software realms. It uses a conceptual model for social data, a formal model of such data based on expert information and Network Theory, and a schematic model of a simulation module informed by the conceptual and formal models as shown in Figure 11.

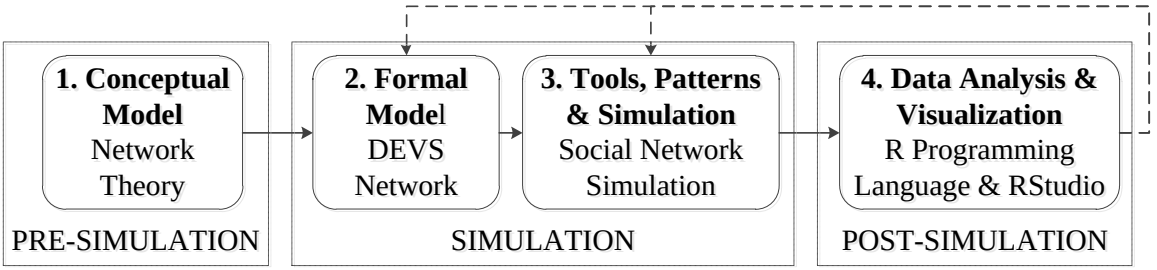


Figure 11: Hybrid architecture for modeling information diffusion processes

The architecture in Figure 11 is a hybrid combination of methods for M&S of information diffusion processes:

- **Pre-simulation:** We define a new conceptual model using Network Theory (Box 1), where the nodes are the agents (people) and the links the relations between them. We store the information in a database making each experiment accessible and re-playable. The repository contains all the individual static models.
- **Simulation:** The process starts with transforming the static networks models to a Formal Model (Box 2): a DEVS network is built automatically by instantiating an atomic DEVS model. Each agent is specified as an atomic model, or, if it has complex dynamics as a coupled model with micro behavioral and evolution rules. DEVS is used for defining each entity and simulating the behavior of the agents dynamically. This simulation is run using a DEVS simulator: Virtual Laboratory Environment (VLE) (Quesnel et al. 2009) (Box 3).
- **Post-simulation:** We use R to process the simulation results (Box 4). The result files are used to visualize the simulation and to conduct the analysis. The analysis can lead to a new cycle. This allows focusing on critical parameters (communication channels) that determine the model output (influenced agent).

The hybrid simulation uses a MySQL database for input, a DEVS-based simulation model implemented in VLE and RStudio to analyze and visualize the output results. These tools are used to generate and analyze the information propagation in the network generated in the first step (Box 1 Figure 11). The simulation starts with the experiment to simulate. The *GraphLoader*, a DEVS auxiliary model, connects to MySQL database to retrieve all the network configuration information from the experiment. It transforms this dataset into a DEVS coupled model called *DEVS network*. Another DEVS auxiliary model, the *Generator*, is used to prepare the information item to be spread through the network. The simulation is driven by an R script to produce simulation results, which are processed to conduct statistical analysis (Figure 12).

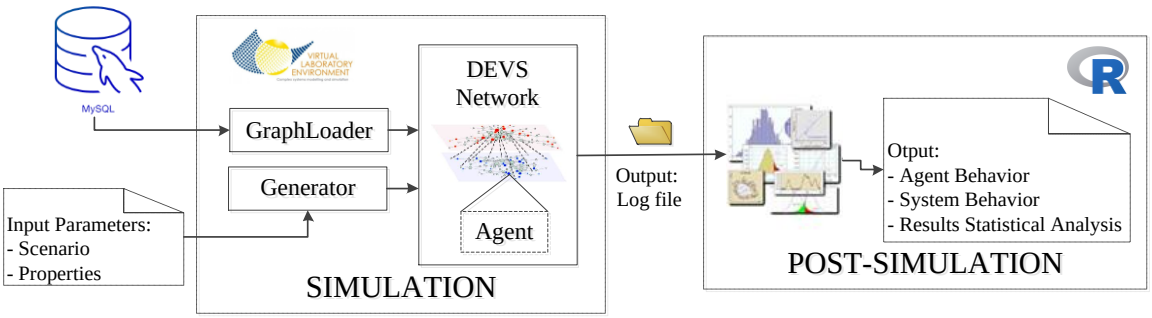


Figure 12 Detailed description of the simulation and post-simulation presented in Figure 11

This architecture supports dynamic reconfiguration of the models based on the properties of DS-DEVS. Nevertheless, this feature is not exploited in the application to the NEP (i.e. no add/suppress links and models at runtime). In the presented scenarios, links (layers) are de/activated at initialization (t0). This preliminary study is focused on the applicability of the Server-Proxy hybrid architecture to analyze communications in the emergency plan using its features increasingly.

Thinking about the relationships between people as a network, and seeing people inside the network as agents make it easier to design a conceptual model. In our model, we expect that information exchanged between people can arise at any time and on a great number of occurrences. DEVS answers to these needs; it is suitable to make a formal and executable model of the conceptual model developed using ABM and Network Theory.

The major problem with this approach is that network size (number of individual) is usually large. Besides the study attends to consider and simulate different network connections. If there are many individual's behaviors, it may be difficult to scale up the formal model. The modeler has to develop different DEVS models to include this variance. However, as DEVS models can be parameterized, we can significantly reduce the effort to customize after an appropriate design.

To solve the problem of simulating different configuration networks easily, the approach presented in this section and proposed by (Bouanan et al. 2016), has been based on the *Model Driven Engineering* (MDE) approach to transforming automatically a network configuration into DEVS network. As we have already explained, it implements a *GraphLoader* that instantiates DS-DEVS parameterized models from networks (graphs) stored in MySQL database. To define these parameterized models, they used two types of DEVS atomic models: *Server* and *Proxy* models. The *Server* models define the behavior of the agents and their reaction when they receive information. The *Proxy* models represent how the information is transmitted in the network taking into account the different layer properties. Each *Server* is connected to several input and output *Proxy* models to describe the multiplex network. When the input *Proxy* receives information, it resends it to the *Server* model. The output *Proxy* model resends the information to all its input *Proxy* model connections. Finally, a *GraphViewer* is used to log the state of the *Server* models.

5.3. Proxy-Server Hybrid Architecture Applied to the Nuclear Emergency Plan

To test the applicability of the architecture presented in section 5.2, we consider the information exchanged between agents (the people involved in the NEP) through discrete-event messages whose values match the messages exchanged between the NEP individuals. The information reception is specified by input messages on the model's ports. The behavior of the model is driven by the DEVS transition functions. The combination of all the transition functions of the component models defines the autonomous behavior of the agent.

We use a basic behavior: the agent receives information and it resends it to all its neighbors in all layers simultaneously. We assume that all nodes have the same behavior, as the main objective here is to show the applicability of the architecture to the proposed problem.

In Figure 13, we present an example of a multiplex network with 3 layers (i.e., 3 different types of communication) and 4 nodes (agents). *Servers* represent the agents and they are represented simultaneously in the three layers using the *Proxies*. Intra-layer interactions (represented by solid lines) show the connection between agents based on the communication channels. Dashed lines represent the inter-layer interactions (i.e. agents in different networks). Component P_x, P'_x, and P''_x are *Proxies.* They contain the specific diffusion rules for each agent in each layer. The *Proxies* are connected to *Servers* (S_x). The *Servers* represent the agent's state and contain the agent's rules for the information received.

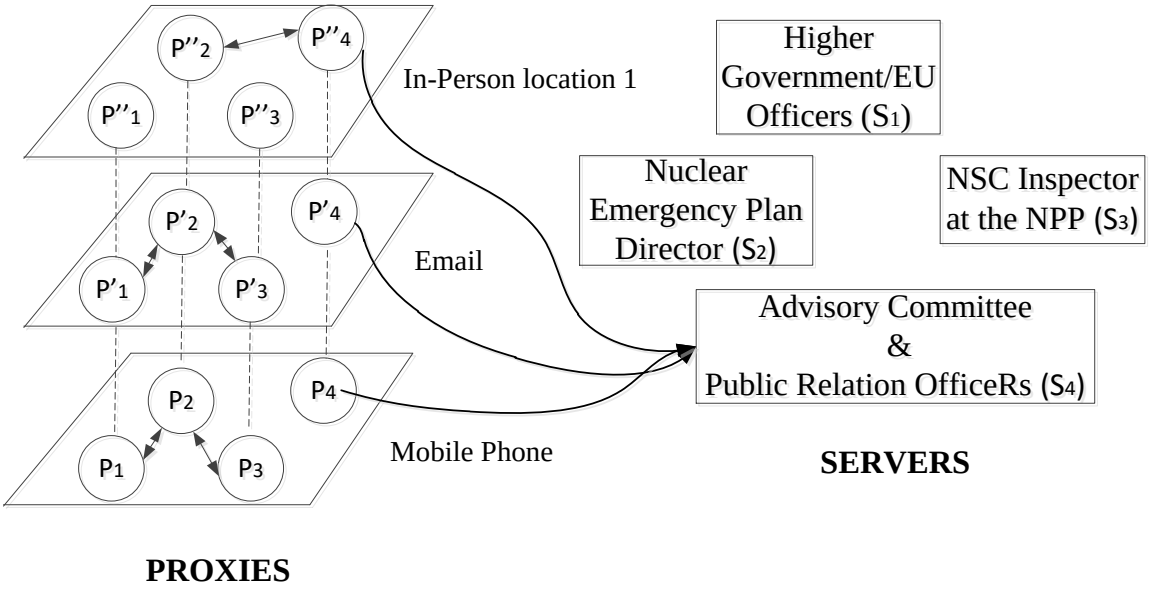


Figure 13 Sketch of the DEVS model architecture for a network with four nodes and three layers.

For instance, when information arrives at P_1 on the "Mobile phone" layer, then:

- The component P_1 sends an event to Server S_1
- S_1 reads the event and transmits the information to its networks, P_1, P'_1 and P''_1.

- Components P_1, P'_1 and P''_1 read the event and depending on their state and rules, can diffuse the information to their neighbors. In this case, P_1 sends an event to P_2; likewise, P'_1 sends an event to P'_2; P''_1 does not send any event.
- Components P_2 and P'_2 send an event to S_2.
- S_2 reads the event and transmits the information to its networks, P_2, P'_2 and P''_2, etc.

The model is constructed taking into account the connections defined in the network. We use the *GraphLoader* to building the DEVS Network. It connects the *Proxy* and *Server* models using the relations defined in the multiplex network and stored in a MySQL database.

Following, we discuss a case study for three different models for a Spanish NEP, representing different abstraction levels. Each DEVS Network model (Figure 18) is built as explained above using the *Graphloader*.

We will only discuss the two first levels in the model, and a more detailed version for the radiological group, as the objective is to show the application of the architecture.

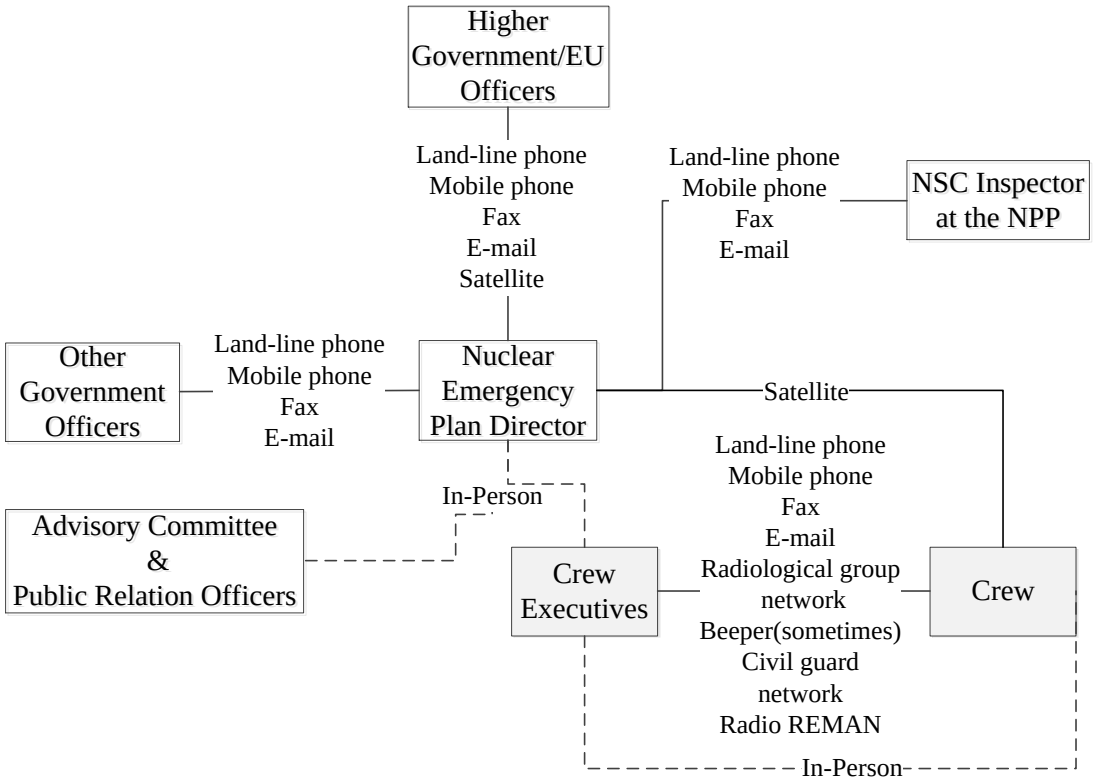


Figure 14: Higher abstraction level of the NEP model.

Figure 14 depicts the top level of the NEP structure. If we represent this model as a network, we have 7 nodes and 10 layers. This network will be translated into DEVS: each box in Figure 14 repents an agent (using ABM) which is translated into one *Server* model and several *Proxy* models in DEVS.

The crew executives and the crew can be defined as in Figure 16 and Figure 15. In these models, each executive agent represents one person.

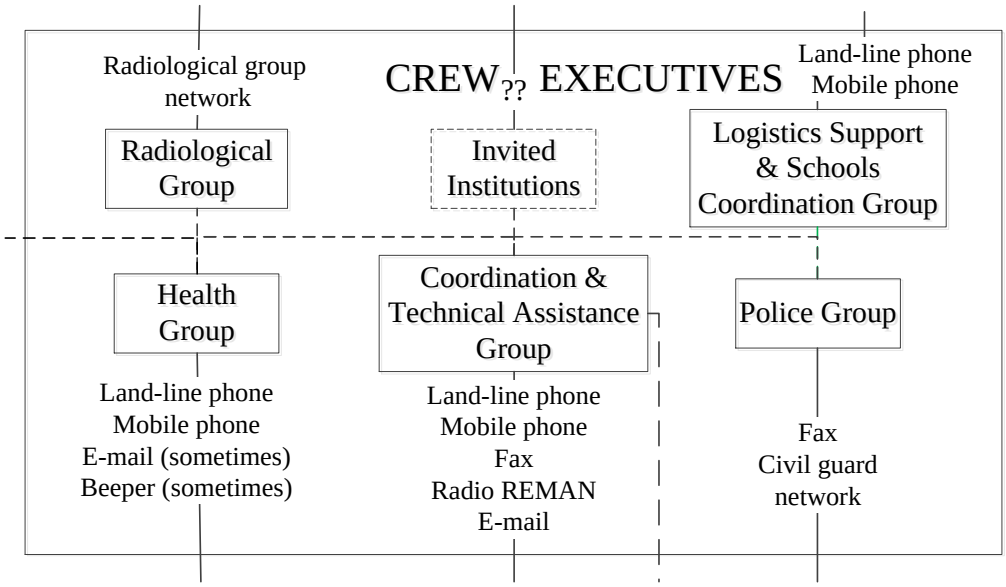


Figure 15 Second level of abstraction: crew executives

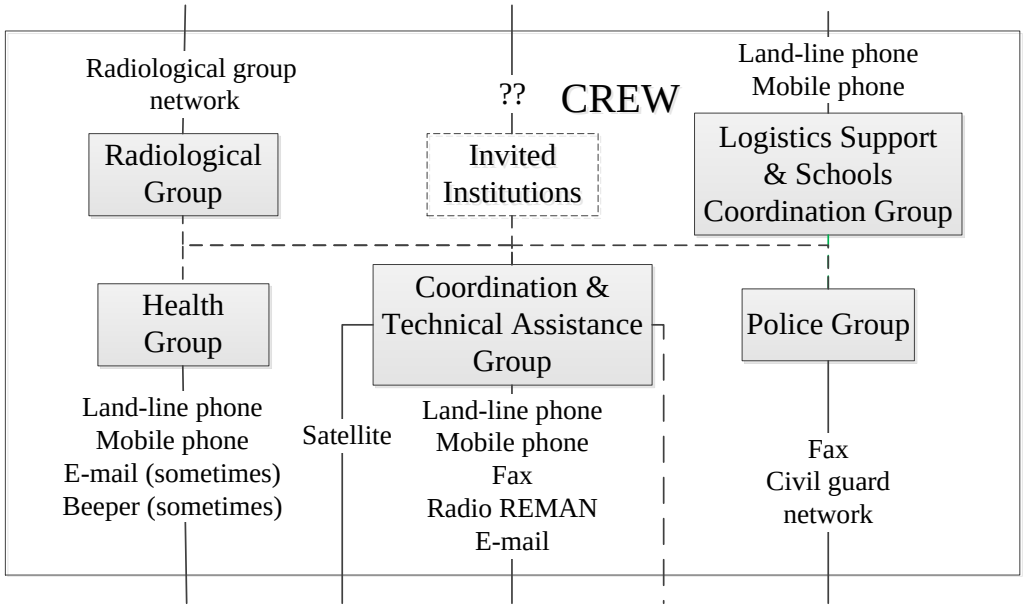


Figure 16 Second level of abstraction: crew

For the crew, we decompose the radiological group in a third level of detail shown in Figure 17. This hierarchical decomposition allows us to focus our interest in the level of detail we need depending on what we are analyzing. For example, if we are interested in studying how the radiological group works, we need the third level and maybe decomposing the teams in Figure 17 further.

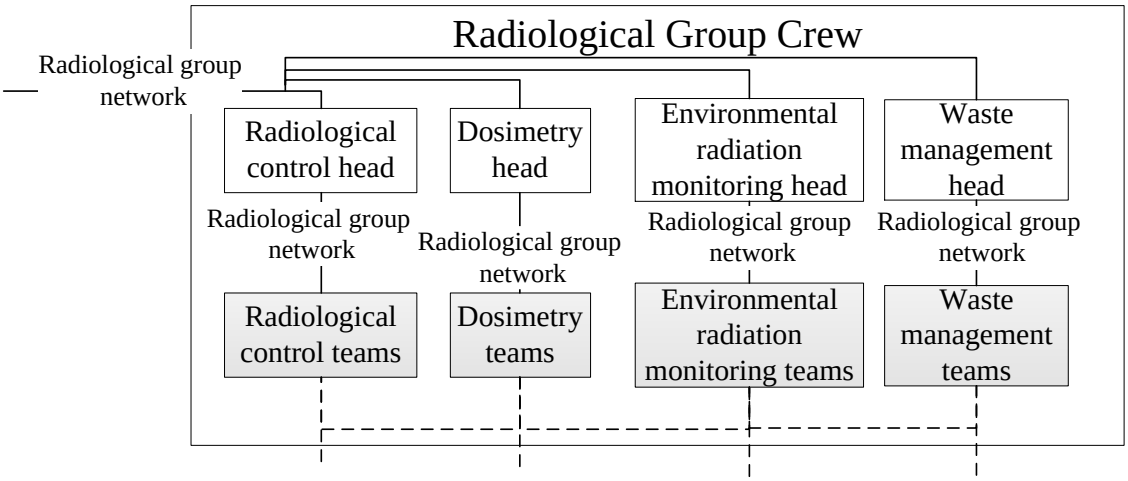
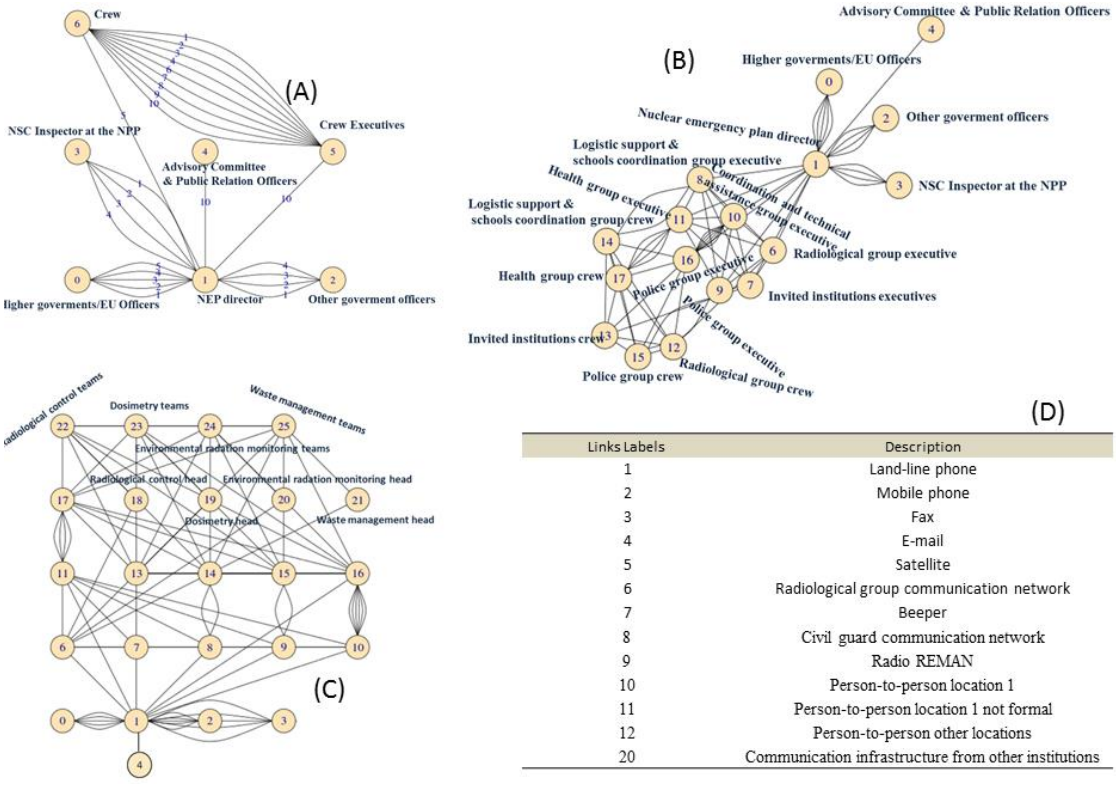


Figure 17 Third level of abstraction: radiological group crew.

Each node in Figure 18 represents an agent, which is defined as a Coupled model (a *Server* model and as many Input and Output *Proxy* models as communication layers in the model representing the communication mechanism detailed in Figure 18 D). The links in the figure represent the connections between Coupled models. Figure 18 A represents the highest level of abstraction, which considers all the Crew and Crew Executives as a single agent. It has 7 nodes and 10 communication layers. In Figure 18 B, we decompose the Crew Executives and the Crew in several agents as shown in Figure 15 and Figure 16. In this second level, we have 17 nodes and 12 communication layers. We show an example in further detail including one of the Group Crews (the Radiological). In this case, we obtained the network represented in Figure 18 C with 24 nodes and 13 communication layers.



Links Labels	Description
1	Land-line phone
2	Mobile phone
3	Fax
4	E-mail
5	Satellite
6	Radiological group communication network
7	Beeper
8	Civil guard communication network
9	Radio REMAN
10	Person-to-person location 1
11	Person-to-person location 1 not formal
12	Person-to-person other locations
20	Communication infrastructure from other institutions

Figure 18: Graph representation of three different models of the NEP and description of link labels.

We have run three scenarios with the same group of people but with different configurations for the communication channels. In each scenario, we generated the network based on the relationships defined in MySQL database and the micro behavioral model for each agent. In scenario 1, we simulate the information transmission process when the message starts in the director NEP (node 1) and the network includes all the agents without layers 11 and 12 (person-to-person communication may exist but it is not considered due to informal aspect and efforts to estimate agents' position). In scenario 2, we run 50 simulations by connecting the generator (at time 0) to the NEP director (node 1) and changing at random the inactive layers (we at random deactivate two layers for each iteration). The goal is to show the impact of the communication channels on the management of emergency plans. In this case, we simulated the level three (the one with most details) and then we analyzed the results to verify the different hypotheses. Finally, in scenario 3, we deactivated layers 1, 2, 3, 4, 7, 11 & 12 to observe who cannot receive the message. We have chosen this scenario because:

- The NEP does not specify who has access to Beeper (layer 7)
- Informal communications (layer 11) and communications based on the changing location of the agents (layer 12) are difficult to define
- We are interested in studying a failure in the whole phone communication network. The phone network supports landline communications (1), mobile communications (2), faxes (3) and Internet (i.e. email – 4). We chose this network because is the only one used that is also used by the population, and in case of emergency it is the more likely to collapse.

5.4. Simulation Results

In Figure 19, we show the simulation results of the three NEP models in scenarios 1 (first row) and 3 (second row). The results show which agents can get the information based on the communication relations in the NEP, when the information starts in the NEP director (node 1). The results of scenario 2 are shown in Figure 20.

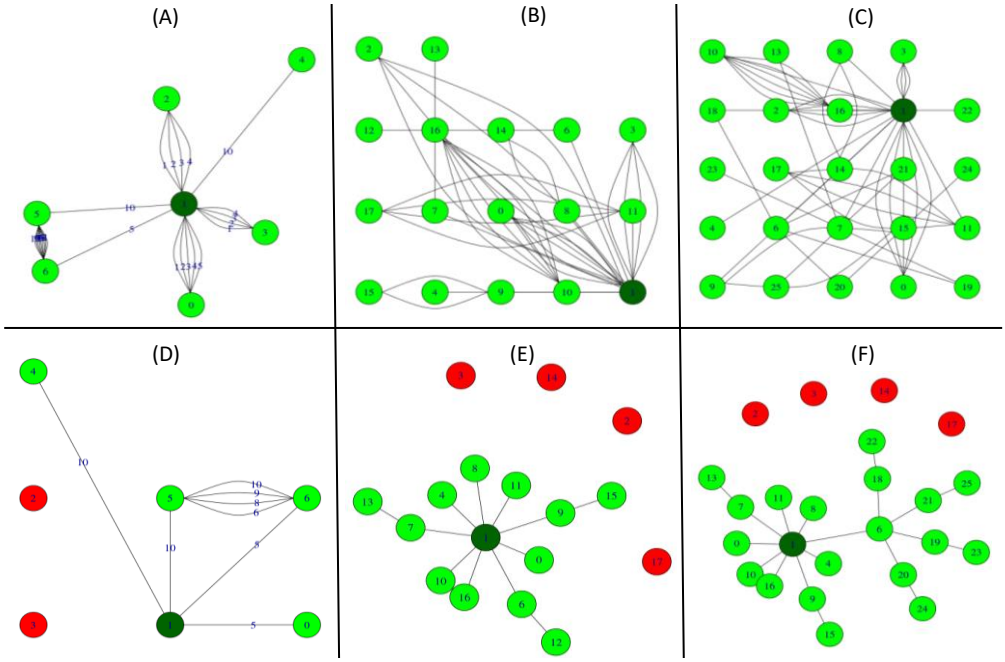


Figure 19 Simulation Results of information dissemination in NEP.

In the first scenario (i.e., all formal communications defined in the NEP are available, except those person-to-person communications based on the location), all the agents get the information. This is an expected result as everything is working as designed.

In scenario 3, we represent a downfall in the Landline phone, Mobile phone, Fax, E-mail, and Beeper communication mechanism. In this case, we see that some individuals do not get the information (represented in red color in the second row of Figure 19). These individuals differ in the different levels of abstraction of the model. However, as we get a more detailed model, the number of isolated individuals increase. This is because in the higher levels of abstraction, we group individuals under the same agent and we consider that the communications in this high-level agent are supported by all the individual agents' communications. These results concur with the results obtained when we analyzed a downfall in different communication channels using Network Theory (Ruiz-Martin, Lopez-Paredes, et al. 2015), where we got the same isolated individuals for this scenario. These results are detailed in Chapter 9

Figure 20 shows simulation results for scenario 2 (random downfall in two communication layers at the same time) for the more detailed model of the NEP (Figure 18c). We represented the number of agents that receive the information. As we can see, there are redundant communication layers; for example, Beeper (7) and Landline phone (1). Other layers are critical; for example, if we do not allow person-to-person communications among the agents in location 1 and the Satellite fails, the information only gets to four nodes. With this last analysis, we can find the combination of downfall that are more critical in the information transmission process during the emergency.

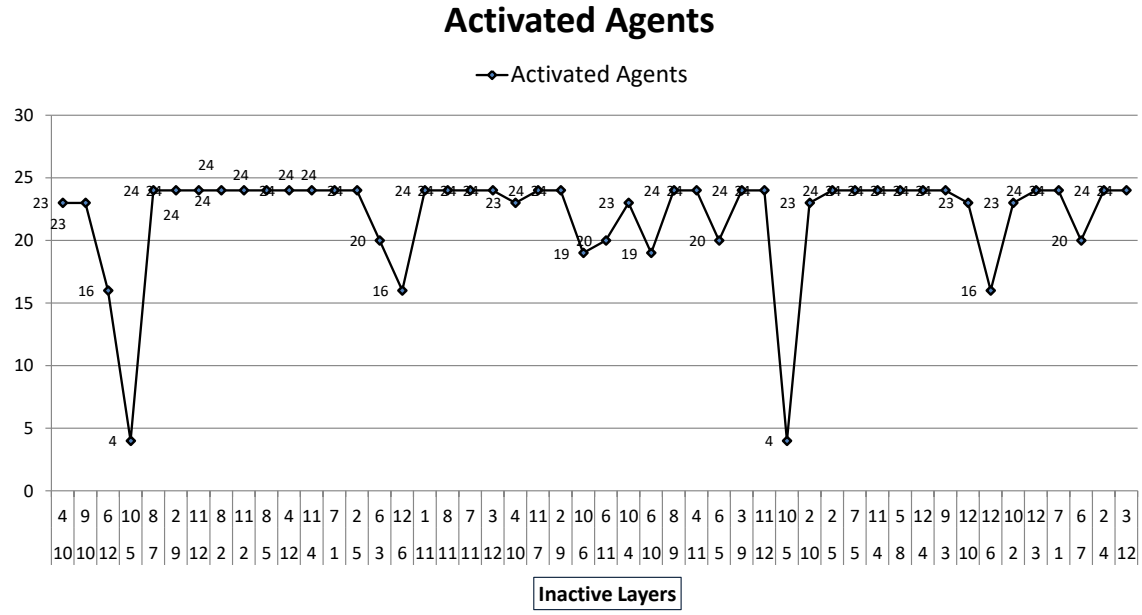


Figure 20 Number of active agent depending on communication channels.

5.5. Drawbacks of the Server-Proxy Architecture to the Study of Communications inside the NEP

The architecture presented in this chapter has been successfully proven to study the information diffusion process in social networks (Bouanan 2016). Moreover, it has been adapted to study business processes in the healthcare sector (Sbayou et al. 2017). The authors modified the architecture to include Business Process Model and Notation (BPMN) in order to study the impact of dynamic allocation of patients in the healthcare pathway. However, it has some drawback when including the behavior of the people involved in the plan and when simulating different scenarios.

In our case, the behavior of the people is defined using complex rules that do not fit in a table format. Storing the properties in a table is a restriction of this architecture since it uses MySQL. We need structures that are more complex in order to store the behavior of the agents.

Moreover, the *Server-Proxy* architecture does not allow us to include all the properties of our model. We are studying the diffusion of information between people, which use different devices, and these devices are connected through different networks. We are interested in studying different scenarios where both the devices and networks can fail. Therefore, it is crucial to include both of them in the model.

Finally, to simulate dynamic networks, the architecture presented in this chapter uses DS-DEVS. The use of DS-DEVS implies that we need to store all possible network configurations we want to simulate, which is not efficient in time of definition efforts and storage.

Based on the architecture presented in this chapter, we propose an improved version that overcomes these issues. We also use Network Theory, ABM, and formal M&S (in our case DEVS) as in the previous case. A main difference is that this new architecture is general and can be used for any type of diffusion process in multiplex dynamic networks. There are many other aspects that differ from (Bouanan 2016). We introduce a development process (and a generic implementation of the architecture). We define a generic Diffusion Abstract Model (DAM) that can be modeled and implemented using other formal M&S methodologies different from DEVS. The design of the DAM is flexible and it allows modeling diffusion processes without storing all possible network configurations. It also provides several advantages such as including other properties of the network, as detail in the rest of the thesis.

Chapter 6. Architecture to Simulate Diffusion Processes in Multiplex Networks

In this chapter, we present our proposed architecture to simulate diffusion processes in multiplex networks. Diffusion processes are models to represent the mechanisms by which a given object (i.e. a virus, an idea, a molecule, etc.) is spread out in an environment, starting from an area with a high concentration of the given object. Studying diffusion processes is useful in different domains. For instance, they can be used to understand how a disease spread through a population, how rumors are disseminated or how different political ideas can be spread through the Social Networks. We also introduce the development process built over the architecture. We then focus on the most important component of our architecture, the Diffusion Abstract model (DAM). We explain its general definition and implementation using DEVS.

6.1. Architecture for Multiplex Dynamic Networks

The architecture, which is presented in Figure 21, is generic and it is suitable for the study of different types of diffusion processes, as we will discuss later.

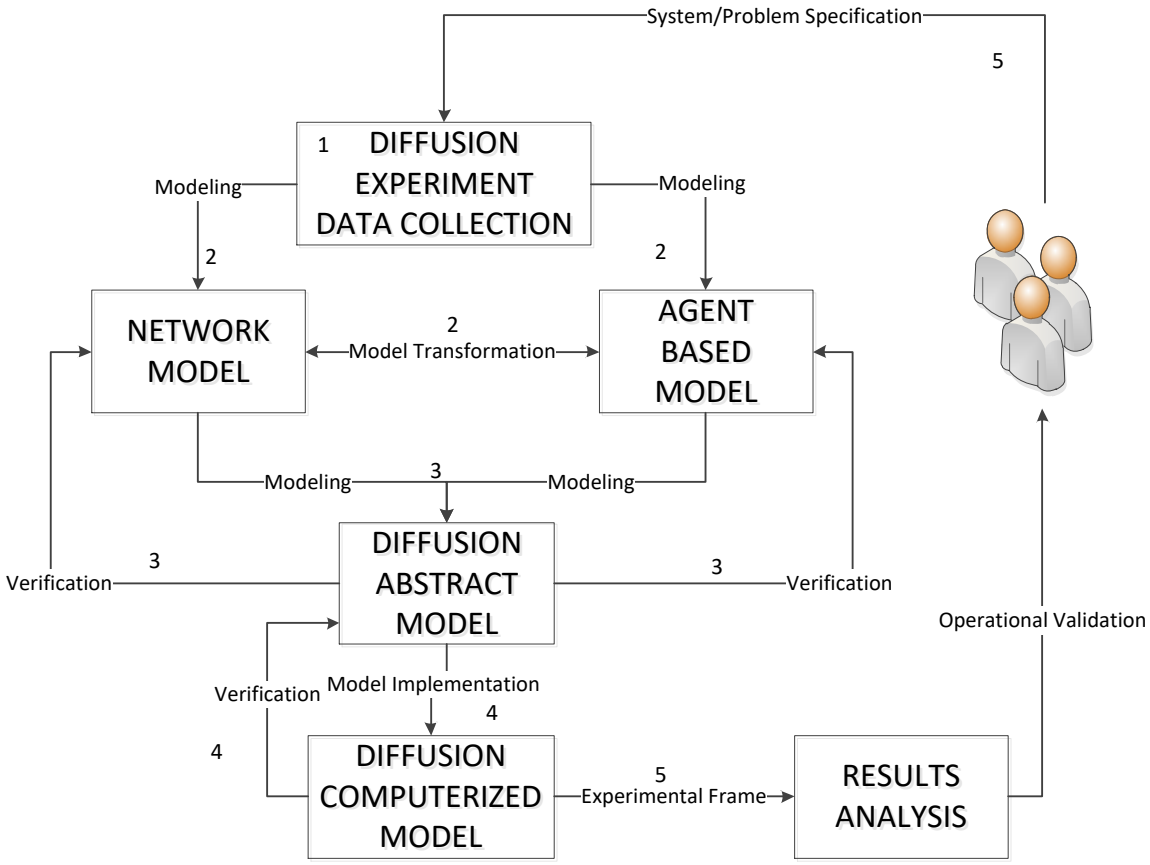


Figure 21. An architecture to simulate diffusion processes in multiplex dynamic networks

It contains six components. The *Diffusion experiment data collection* represents the process of collecting all the system/problem requirements. The *Network model* is a representation of the relations among the components of the system/problem under study. The *Agent Based model* is a representation of the behavior of those ones in charge of the diffusion process, the objects they use for diffusing the element and the properties of the relationships among these objects. The *Diffusion Abstract model* is an abstract and formal representation of all the requirements collected on Diffusion experiment data collection. It is built using both the Network and Agent Based models. The *Diffusion Computerized model* is a computerized model of the Diffusion Abstract model. The *Result analysis* represents the analysis of the simulation results.

The numbers in the figure represent the different steps of the development process we define over the architecture. They are detailed in section 6.2

A main advantage of this architecture is the provision of a Diffusion Abstract model, which should be defined using a formal specification method. Having a formal model helps with early validation prior to implementation. In our case, we will use DEVS, which, as discussed in Chapter 3, allows defining models originally specified in other formalism. Moreover, as discussed later on in section 6.1.4, this solves some of the limitations of Network Theory to study diffusion processes in dynamic networks (for instance, ad-hoc implementation for the simulation of diffusion models in networks, lack of clear implementation details, etc.). Moreover, the DEVS formalism provides rigor to the ABM, being able to separate model definition, formal verification, implementation, and simulation. Additionally, the same model can be implemented on different platforms.

Another advantage of using DEVS is that it allows to develop models in a modular way. This modularity allows model reusability and thus, reduce the development time and testing. The modularity also allows collaboration between developers of different components. Moreover, the simulation algorithm for DEVS has been formally verified, providing a well-defined method for simulating the models.

The human experts in Figure 21 are key as they can use the results of the architecture and act on hazard mitigation plans. They participate in the validation of the simulation results and they can propose new polices to improve the system. Any policy change can be tested using our architecture, saving time and money. For example, if we focus on the information transmission process to manage an emergency, new communication methods can be studied. If we are interested in studying an epidemic, we can see the effect of different means for slowing down or control the spread of the disease (educating the population using different techniques using advertisement, diffusion on TV and social media, vaccination policies, quarantine, etc.).

In the rest of this section, we will detail each component of the architecture.

6.1.1. Diffusion experiment data collection

The *Diffusion experiment data* is composed of all the requirements, the specifications, and all the data available from the problem or system we are interested in studying. All this information can be gathered manually or automatically based on the problem or system we want to study. For example, it can be gathered manually through interviews or text analysis. It can also be gathered automatically through different types of sensors.

In general, the specialists in charge of providing the Diffusion experiment data are familiar with the system or in the proposed problem. If not all the data, the requirements or the specifications are yet available (and we need to complete them), the specialist familiar with the system/problem should provide instructions about what information they must collect and how they must do it.

This information can be stored in natural language or in a structure way such as tables. Having the information structured eases developing the rest of the components of the architecture.

6.1.2. Network model

The *Network model* is an organized representation of some information provided in the *Diffusion experiment data*. The *Network model* provides a formal representation of the relations among the components of the system or the problem under study. In our architecture, this model is formalized using Network Theory, and it can be implemented and stored in different formats, such as a table, a graph or an XML file.

There is different software available to analyze properties of the network model: Gephi (Bastian et al. 2009), Pajek (Nooy et al. 2005), MuxViz (De Domenico et al. 2014), R (Ihaka & Gentleman 1996), which includes a package for network analysis called igraph (Csardi & Nepusz 2006), etc.

The designers who build the *Network model* development should be familiar with the concepts of Network Theory. To build the model, they use all the requirements, specifications and data collected in the *Diffusion experiment data* component.

6.1.3. Agent Based model

The *Agent Based model* is a representation of the behavior of those ones in charge of the diffusion process, the objects they use for diffusing the element and the properties of the relationships among these objects. It is formalized using ABM concepts, and it can be implemented using different methods: DEVS, an XML file definition, or specific software platforms such as NetLogo (Wilensky 1999), Repast (North et al. 2006), etc. In (Nikolai & Madey 2009), the authors provide a comprehensive review of more than 50 toolkits available for the implementation of Agent Based Models. Those in charge of this phase should be familiar with behavioral modeling depending on the diffusion problem we are studying and ABM techniques.

In Figure 22, we show an example of an agent implemented using XML. In this example, we show a generic agent with the minimum set of attributes to capture the connections in the Network Model and the behavior defined for the diffusion process.

The behavior of each agent is defined between the tags <*AgentBehavior*>. The behavior must contain at least the following attributes:

- *Id*: represent the Id of the agent (i.e. a node in the network model).
- *MyLinksTypes*: represents the types of the input and output links of the agent. The number of *Link* elements inside *MyLinksTypes* may vary between 0 and the total number of link types in the network model.
- *MyRelations*: defines the agent (node) output connections with other agents. The number of *MyRelations* elements in the XML file is equal to the number of output connections of

the node in the network model. Each *MyRelations* element has an *id* that represents the agent we can contact. It also has as many *Link* elements as *MyLinksTypes* we can use to contact the agent.

- *BehaviorRules*: defines the agent behavior regarding the diffusion process. The number of elements inside *BehaviorRules* (i.e. *Rule*) will depend on the model. Each *Rule* element can have different parameters.

```
1 <?xml version="1.0" ?>
2 <AgentBehavior>
3   <Id>MyId</Id>
4   <MyLinksTypes>
5      <Link Id=Link1/>   ...
6      <Link Id=Linkp/>
7   </MyLinksTypes >
8  <MyRelations id=AgentId1>
9       <Link Id=Link2/>
10      <Link Id=Link5/>
11  </MyRelations >       ...
12  <MyRelations id= AgentIdt/>
13      <Link Id=Link2/>
14  </MyRelations>          ...
15  <BehaviorRules>
16    <Rule Id = RuleId1 Parameter11 = Parameter11Value ... Parameter1n = Parameter1nValue>
17       ...
18    <Rule Id = RuleIdt Parameter1t = Parameter1tValue ... Parametern = ParametermtValue>
19  </BehaviorRules>
20  </MessageBehavior>
21 </AgentBehavior>
```

Figure 22. Example of the agent's definition using XML.

MyLinksTypes and *MyRelations* capture the multiplex part of the network in the agent definition. The name of the attributes can be modified to make the behavior readable in a specific context. For example, in an information diffusion process, *MyLinkTypes* can be *Devices* or *CommunicationMechanism.*

6.1.4. Diffusion Abstract model

The *Diffusion Abstract model* (DAM) is an abstract and formal representation of the *Diffusion experiment data* that matches the elements in both the *Network and Agent Based models*. It is formalized using a mathematical specification. In our case, we use DEVS, but this component of the architecture could be applied using other formal specifications such as System Dynamics, Finite State Automata, etc. It is also possible to use different methodologies for the different components as long as there is a high-level architecture to connect them. The modelers should know formal modeling techniques (in our case, DEVS modeling and simulation).

To design the DAM we use as starting point the server-proxy architecture presented in Chapter 5 introduced by Bouanan et al. (2016) and the limitations we identified to defined the model of the NEP using their approach.

The DAM is a generic container that follows the architecture depicted in Figure 23. It includes nine components: *Node, Indirect link, Link Connectors, Direct Link, Diffusion Elements Generator* and 4 *Updaters.* The arrows in the figure represent how the components are connected.

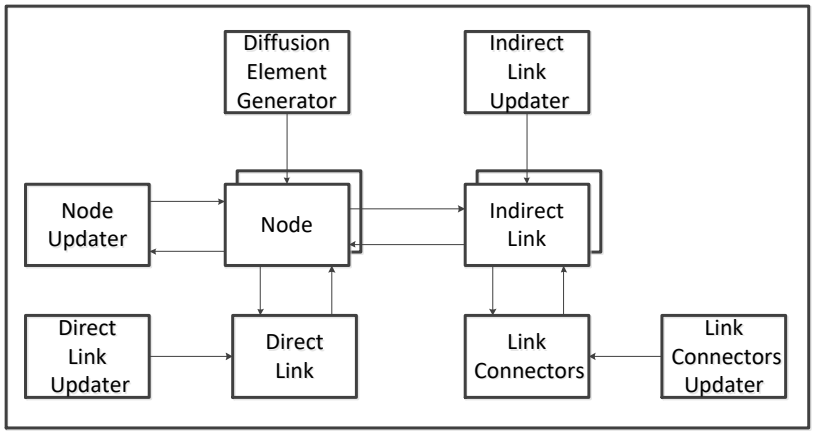


Figure 23. Diffusion Abstract model architecture

Node is a representation of a node in the *Network model*, including all its inputs and outputs connection of the node. It also represents an agent in the in the *Agent Based model* since agents also represent the nodes in the *Network model*. The number of *Node* models in the *Abstract Diffusion model* is equal to the number of nodes in the *Network model* or the number of agents in the *Agent Based model*.

As we have already mentioned, each agent can have a similar behavior (i.e. the same behavior but with different parameters) or different behaviors. This also applies to the *Node* model. It can be just one model with different parameter instantiations, completely different models or a combination of both.

Indirect Link represents the properties of the links of both the input and output connections of a node in the *Network model*. The *Indirect Link* also matches with the objects used by the agents to carry the diffusion process. The number of *Indirect Link* models in the *Abstract Diffusion model* is the same as the number of *Node* models.

Each of these models is different. Once all the objects used by the agents to carry the diffusion process have been modeled, each *Indirect Link* will contain a different subset of them based on the properties of the input and output connections of the nodes (or agents).

Link Connectors is a single model that represents how the objects used by the agents that carry the diffusion process are connected. It does not have a direct match to the *Network model*. As we have already mentioned, the *Network model* does not include all the information captured by the *Diffusion experiment data collection* component of the architecture. In the *Agent Based model*, it represents the properties of the relations among the *Indirect Link* models and how are they connected.

Direct Link is a model that represents direct connections between *Node* models (i.e. connections that do not use an intermediary object). For instance, in the context of disease dissemination, it could represent touching without using gloves. In the context of information dissemination, it could represent people talking in-situ instead of using a phone. This model represents the properties of the links in the *Network model* that establish a direct connection between nodes without using any intermediate object. In the *Agent Based model*, it represents the connections between agents that are handled without using any additional objects.

Diffusion Element Generator introduces the elements to be diffused over time. This can be done at the beginning of the simulation or at runtime. For example, in the study of the diffusion of a rumor through a population, one rumor can be placed in some nodes at the beginning of the simulation and many other rumors may appear at runtime.

In the *Network* and *Agent Based models*, the *Diffusion Element Generator* matches the location of the initial diffusion elements and the new diffusion elements introduced over time in the nodes or agents.

Link Connectors Updater, Indirect Link Updater, Direct Link Updater and ***Node Updater*** introduce modifications in the properties of the models over time. They allow us to model the dynamicity of the *Network* and *Agent Based models*. We can modify the properties of the *Indirect Link*, *Link Connectors*, *Node* and *Direct Link* models over time without modifying any model of the architecture.7

The proposed architecture is generic and it could be used to study different diffusion problems. The number of components used in the *Diffusion Abstract model* would depend on the type and characteristics of the problem we want to study. For example, to study a diffusion problem on a static network where the behavior of the nodes, links and connectors is not affected by external variables, the *Updaters* would be removed.

Based on the variables we want to analyze, at least one of the components we have described has to provide data for *Diffusion Abstract model analysis*. This is not identified in Figure 23 since there are multiple options depending on the problem we study and the formalism used to implement the architecture.

In section 6.3, we detail the DAM. We provide formal definition and explain the different components using DEVS. We want to remark that the DAM needs to be instantiated for the specific application since the behavior defined in the Agent Based model are different for different applications. For example, the behavior of the *Node* will be different if we use it to represent a group of people using their phones, or if it represents ticks spreading Lyme disease on a population. The same will happen with the other components. Here we provide several examples where the DAM can be used.

For example, in order to study a diffusion process of people in the public transportation system of a country, we would model it as a multiplex network where the nodes are the cities. The cities are connected by different means (airport, train station or bus station, etc.), represented as links in the network. In the proposed Diffusion Abstract model, cities would be the different *Nodes*, connected to an *Indirect Link* model (the different means of transportation, which would be different from each city). They would be connected through an infrastructure (roads, railways, flight corridors, etc.), represented as *Link Connectors*. In this case, we do not use *Direct Link* or its *Updater*. The *Indirect Link Updater* model would be used to model the conditions of the transport network over time (for example, an airport can be closed due to a snowstorm or a train station due to a strike). The *Link Connectors Updater* would model changes in the properties of the infrastructure (road closure due to a landslide; airspace altered due to weather conditions). The *Node Updater* model would introduce changes in the policies of the transportation system of the city due to external variables not included in the model (new bus routes to other cities, change in the frequency of trains). The *Diffusion Element Generator* would set the number of people that want to travel from one city to another over time.

A different context would be the use of social networks to study how rumors are spread. The social network would be modeled as a multiplex network where the *Node* models are the persons, connected through different social networks (they can also talk in-situ), represented as *Indirect Link* model (each *Indirect Link* model have a different combination of social network). All social networks are connected through an infrastructure (the social network Servers through the Internet), modeled as *Link Connectors*. The *Direct Link* would allow people in the same location talk to each other. The *Updaters* model would be used to model changes in the social networks (i.e., a user closing their account) or in the infrastructure. The *Diffusion Element Generator* will set which *Node* models initialize the rumor. It can be just one rumor or multiple rumors can be sent over time if we are interested in studying how they interact.

In this thesis, we will focus on a specific application of the architecture in detail: an information diffusion process inside the complex organization defined in a NEP from Spain. For this application, we detail how the DAM is instantiated and the whole development process. We have a double objective with the application of the proposed architecture to this specific example. First, we show how to use the architecture. Second, we explain how study and improve the resilience of the Emergency Plan using modeling and simulation.

The architecture provides various advantages:

- Different scenarios and network configurations can be run just updating the model parameters. There is no need to make changes in the model design.
- There is no restriction on the complexity of the behavior the agent. Any behavior can be modeled.
- Different agents can have very different behavior.
- We improve reusability (since the behavior of the agents and objects are separated, we can reuse these models for the study of other problems).
- Using four models to update the properties of the components, allows us to simulate diffusion processes where the topology or characteristics of the network change over time. We can update the network topology and the behavior of both the nodes and the links at runtime without modifying the simulation model. Moreover, we do not need to store the whole model again with the new properties. We just need to store the changes in the properties and when (i.e. at what time) they occur.

6.1.5. Diffusion computerized model (DCM)

The DCM is an implementation of the DAM, which can be defined using different simulators. The simulator we chose will depend on the formalism used to implement the DAM. In our case, we used a DEVS simulator: CDBoost. As explained in Chapter 3, we choose CDBoost (Vicino 2015; Vicino et al. 2015) among the DEVS simulators available based on its advantages. One of the main advantages of CDBoost is that the output format of the simulation is flexible. The user can configure the logs it in the way that better fits his needs. Therefore, it can be defined in such way to make simulation output analysis easy.

Once all the components of the *Diffusion Abstract model* are implemented, the top model can be implemented either manually or automatically using a script or software that processes the information in the *Network and Agent Based models*. The developers should be familiar with the implementation

of the formal methodology used to develop the DAM. Therefore, they can be the same that develop the DAM or different people.

6.1.6. Results analysis

The *Result analysis* component represents the process of analyzing the simulation results provided by the *Diffusion computerized model*. The analysis process can be carried using different statistical analysis and data visualization tools such as R (Ihaka & Gentleman 1996), PowerBI (Microsoft 2015) or any tool available for big data analysis. The analysts should be familiar with data analysis techniques.

6.2. M&S Development Process for Multiplex Dynamic Networks

In this section, we explain the development process proposed over the architecture depicted in Figure 21 and explained in Section 6.1. The different steps of the development process are identified with numbers in the figure.

6.2.1. Step 1 – System/problem requirement gathering

The first step is to gather the requirements of the system of interest or problem under study. This step is carried out with the help of technical people to collect all the specifications and problem/system details. Although every case is different, we identified a minimum set of requirements extrapolating the ones identified for the case study (i.e. the NEP). The requirements for the NEP were identified based on the information provided by the experts at the Civil Protection Agency. These requirements include:

- What are the elements to be diffused? Some examples of diffusion elements include rumors, viruses, vehicles, etc. It is also possible that a diffusion process includes more than one type of element.
- Who is going to diffuse the above-mentioned elements? For example, if we are studying the diffusion of a virus, both animals and persons could be responsible for the virus transmission.
- What is the behavior of those that are diffusing the elements? The behavior can be the same for all of them with different parameters, or they could have different behaviors. The behavior can range from a simple rule to a set of complex rules. All behavior rules may depend on different variables such as intrinsic characteristics, the relationship types they have, the diffusion element, etc. For example, in the diffusion of a rumor through a population, the behavior of the person who is going to spread the rumor may depend on intrinsic characteristics such as gender or age; it may also depend on the Social Networks they use and the content of the rumor, among other variables.
- Where will the diffusion element start? For example, it can start in a person, an animal, a robot, etc. It may also start in multiple places at the same time or there might be new diffusion elements that appear in different locations over time.
- What are the effects of the diffusion elements, and which of them are relevant in the study? For example, based on the problem that we are studying, the effects may be a change in the

person's opinion or behavior, unfriending a person on Facebook, a machine stops working, etc.

- How the diffusion elements are going to be spread (i.e. what the mechanisms are)? These diffusion mechanisms may be a phone, direct connection, etc. More than one mechanism could be used.
- What are the characteristics of the above-mentioned mechanisms? How are they connected? For example, in a diffusion information process in a population, what are the characteristics of the phones and how are they connected.
- Which diffusion mechanisms each individual may use? For example, in a transportation system, which cities have an airport, which have a bus station, etc. It may be something clearly defined as in the example or more high-level data such as 20% of the cities have an airport, 80% bus station, all cities that have airport also have a bus station, etc. It may also be a combination of the two examples.
- What are the variables of the system/problem we are interested in? For example, in the study of the diffusion of a virus, we may be interested in the number of infected people, their gender, the mechanisms for virus transmission, if a specific population group was infected, etc.
- What scenarios should be analyzed? Following the same example of virus diffusion, we may study what happens when people are vaccinated, what happens when a prevention campaign is spread, etc. These scenarios should be defined with all its characteristics and parameters (i.e., how effective is the vaccine, how many people will be vaccinated, etc.).

The output of this step is a *Requirements Document* that can have different formats and length. It may have tables, graphs, text, etc. It collects all the information that describes the system/problem. It is used in the next step to provide modelers with a detailed description of problem or system.

6.2.2. Step 2 – Network & Agent Based models development

In step 2, the *Network* and *Agent Based models* are developed in parallel. The developers are provided with the *Requirements Document* developed in step 1. It is important to establish communication mechanisms between the modelers and the people familiar with the system, as the *Requirements Document* contains natural language, which is ambiguous.

A draft of both models is developed using the information provided in the *Requirements Document.* Once they are ready, Network Theory and ABM modelers should discuss their models. Although very different perspectives can be used to make these models, our architecture establishes a relationship between them to be able to combine the models to develop the *Diffusion Abstract model*: each node in the *Network model* is represented by an agent in the *Agent Based model*. The different types of links in the Network should match with the objects the agents use to carry the diffusion process when they do not represent a direct link between agents.

If Network Theory and ABM modelers need to make assumptions to develop their models, they must be approved by the people familiar with the system. All the assumptions must be gathered in an *Assumptions Document* that must be approved by the people familiar with the system. It can be necessary to go back to step 1 and complete the *Requirements Document* to collect more.

The outputs of this step are the *Network* and *Agent Based models* and an *Assumptions Document* approved.

6.2.3. Step 3 – Abstract Diffusion model development

In step 3, the DAM is developed using the model architecture depicted in Figure 21 and explained in section 6.1.4. The model could be defined using different formal methodologies, provided that the modelers should be familiar with the formalism. As we have already mentioned, in our case, we use DEVS.

The model developers are provided with the *Assumptions Document* and the *Network* and *Agent Based models*. If there is a mismatch between the *Network* and *Agent Based models*, we might need to revisit step 2. The same happens if there is missing data or they need to make further assumptions.

If the data in the *Requirements Document* is well-structured and unambiguous (i.e. the connections between the system components and their behavior are perfectly defined using the specifications provided in step 2), it may be possible to skip step 2. However, in most cases, the *Requirements Document* is specified using natural language that needs to be processed and translated to an intermediary step to remove the ambiguity.

Based on the *Network* and *Agent Based models*, the model developers decide which components of the architecture should be included. In our research, the components of the architecture are DEVS models. The DEVS modelers translate the rules defined in the Agent Based Model into DEVS models. They can be either atomic or coupled models based on the complexity of the rules and the level of detail needed. A detailed explanation about the Diffusion Abstract model architecture using DEVS is presented in section 6.3.

The output of this step is *the Diffusion Abstract model* (in our case, a DEVS model).

6.2.4. Step 4 – Diffusion computerized model development

In step 4, the *Diffusion computerized model* is implemented by developers that are familiar with the formal methodology (DEVS in our case). Modelers and developers in steps 3&4 are usually the same individuals. To generate the *Diffusion computerized model*, a simulator for the formal modeling methodology is used. In our case, any DEVS simulator can be used to implement the DEVS model. If a simulator for the formal methodology does not exist, we recommend choosing a different formalism. The lack of a simulator reduces the advantages proposed at the begging of the sections such as having separation of concerns in simulation and implementation.

DEVS developers must verify that the model developed in step 3 and the simulation matches. If they do not match, reiterations of steps 3&4 are needed. Since the model developed in step 3 is formally defined, it is possible to do model checking or deductive verification to verify that the components of the DAM match the implementation.

The output of this step is the *Diffusion computerized model*. In our case, a DEVS simulation can be customized with different parameters to simulate the different scenarios proposed in the *Requirements Document*.

6.2.5. Step 5 – Analysis of simulation results

In step 5, the simulation results of the different scenarios are analyzed. The analysis is done by people knowing Data Analytics (which can be the same simulation developers). The results of their analysis are given to specialists familiar with the system/problem (those ones who developed the *Requirements Document*) for validation and decision-making purposes.

New iterations of the process could be needed until the results are validated. Once the results are validated, new iterations are also possible (and recommended) to test different policies before implementing them in the real system.

6.3. Diffusion Abstract Model: General Definition and Implementation using DEVS

In this section, we explain a general definition and implementation for the DAM (defined in section 6.1.4) using DEVS as the formal M&S methodology.

In this DEVS model, the architecture presented earlier in Figure 23 is represented by a Coupled Model (see Appendix Afor the formal definition) and each component is a DEVS sub model. As we have already mentioned, it is the task of the modeler to adapt which components to use for each specific problem.

In the *DEVS DAM* (i.e. DEVS Top Model), the diffusion elements are modeled as the Messages transmitted between the models *Node, Indirect Link, Direct Link and Link Connectors*.

In the rest of the section, we discuss and detail how each of these Models has been defined and implemented, and how they can be used to build DAM.

6.3.1. Node

The *Node*, presented in Figure 24, is a general Coupled model that provides a generic structure for the definition of the behavior of each node. It is a DEVS model of the agent's behavior specified in the Agent Based model. It contains three *Filters* to filter the broadcasted messages. It also has two *Switches* (one for *Direct Link* connections and another one for *Indirect Link Connections*), a *Behavior Rules* model, two other *Filters* to classify sending and receiving outputs of *Behavior Rules* base on the type of link used (Direct or Indirect), two *Sending* and two *Receiving Behavior* models (one pair for Direct Links and another for Indirect Links).The connections between models are shown in the figure.

Direct Link, *Generator* and *Node Updater Filter* filter the Messages based on if the message is for the *Node* DEVS model (remember that the messages are broadcasted). If the message is for the *Node*, they let it pass. Otherwise, the message is discarded.

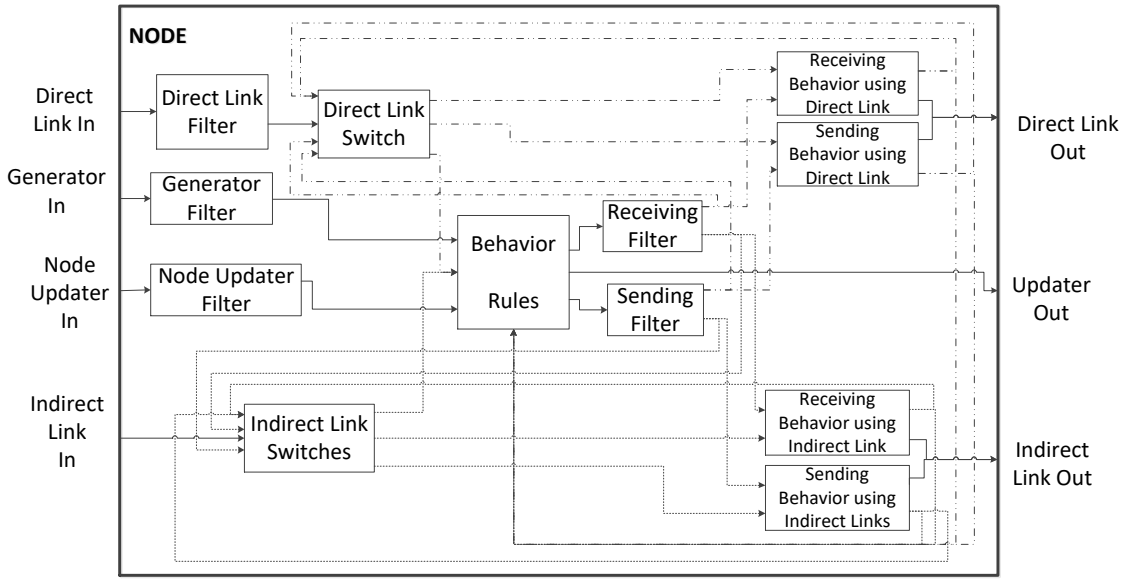


Figure 24 *Node* Coupled model

Direct Links Switch is an atomic model that classifies the incoming message based on the state of the *Switch*. The *Switch* can be in three states: decide, send and receive. It is initialized in decide state. The state is set based on the inputs in *Set Send, Set Decide and Set Receive* ports. Base on the switch state, it redirects the messages received from *Direct Link In* port to the appropriate port (*Send Out, Receive Out or Decide Out).*

Indirect Link Switches is a Coupled model instantiated based on the amount of different types of indirect links the *Node* has (i.e. the amount of object the agent can use to spread the diffusion elements). Therefore, *Indirect Link Switches* in different *Nodes* may have a different number of components. It always has four filters that classify the messages based on the link type they came from and they redirect the messages to the appropriate port. It also has a *Sink* and as many switches as objects to spread the diffusion elements. The way in each atomic *Switch* works is the same as *Direct Link Switch*. The connections in *Indirect Link Switches* are as follow. Each port of the filter models is connected to the *Switch* it refers to if the *Node* has the type of link. Otherwise, it is connected to the *Sink*. The *Sink* facilitates model verification and validation. If a message arrives at the *Sink*, it means that the model is not well defined. The agent is receiving a message from a link that they do not have access to.

Behavior Rules is a Coupled or atomic model instantiated with the parameters that define the behavior of the agent (i.e. the ones defined in the XML file – Figure 22). As explained in section 6.1.3, one of these parameters must be the connections with other agents including the type of link used (i.e. *MyRelations* in the XML). The multiplex network connections are defined here. The *Node* (specifically, the *Behavior Rules* component of the *Node*) contains the information about the relations with the other nodes (i.e. agents). The other parameters define the behavior of the agent when spreading the diffusion elements, and their effect on the agent's behavior. The outputs of this model are instructions either to assimilate or spread a diffusion element.

Receiving and Sending Filters classify the instructions generated by *Behavior Rules* model based on if the Direct or Indirect Links are used. They resend these instructions through the appropriate port.

Receiving Behavior using Direct Link represents the process the *Node* follows to assimilate the diffusion element. Once the process is finished, an acknowledgment is sent. Similarly, *Sending Behavior using Direct Link* represents the process the Node follows to spread the diffusion element.

Receiving Behavior using Indirect Links is a Coupled model instantiated base on the amount of different types of indirect links the *Node* has. For each *Node* DEVS model, *Receiving Behavior using Indirect Links* may have a different number of components. It always has two filters that classify the messages based on the link type come from and redirect them to the appropriate port. It also has a sink and as many atomic models to represent the receiving behavior as different link types the *Node* is connected to. The *Receiving behavior* models represent the process the *Node* follows to assimilate the diffusion element using that specific type of link. The connections follow the same rationality as *in Indirect Link Switches*. Each port of filter model is connected to the *Receiving Behavior* model it refers to if the *Node* has the type of link. Otherwise, it is connected to the *Sink*.

Sending Behavior using Indirect Links is a Coupled model that follows the same rationality as Receiving Behavior using Indirect Links. The only difference is that Receiving Behavior models are now Sending Behavior models. They represent the process Node follows to spread the diffusion element.

Some of these components are coupled models formally defined as explained in Appendix Afor the DAM. Other components, such as *Generator Filter* are atomic models. In Appendix B, we show how atomic models are defined formally using *Generator Filter* as an example.

Having the models formally defined, we can perform early validation without implementing the models. For example, in the *Generator Filter* (see Appendix B), we may forget to clear the state variable *messagesPassingFilter* when the messages have been already sent through the output. We can also find other errors such as a passivating the model when *messagesPassingFilter* is not empty. We can find this type of errors just looking at the formal definition of the model. We do not need to implement or simulate the model to find these errors. In the DAM (see Appendix B), we can check that the connections between the components are well defined. For example, we can verify that we are not connecting the *NodeUpdater* to *DirectLink* model. Otherwise, the model is not valid. These errors can be found easily in the formal specifications. We do not need to waste time on implementing a wrong model in the early phases.

Not all the components explained in this section will be used in all the problems. It is the task of the modeler to decide which ones should be included. For example, if there are no Direct Connections in the Network model, every model related to *Direct Link* (*Filter, Switch, Receiving and Sending Behavior*) are not included. In this case, *Sending* and *Receiving Filters* are neither needed. Every instruction output by *Behavior Rules* will be for *Indirect Links* models.

If we use the same examples discussed in the previous section, if we are studying the diffusion of people in a transportation system, the *Node* model will represent a city. We will not have the models related to *Direct Links*, the *Sending Filter* and the *Receiving Filter*. The number of *Switches* (inside *Indirect Links Switches*), *Receiving* and *Sending Behavior* models will be fixed by the number and type of transportation infrastructure the city has (e.g. airport, train station, etc.). The *Behavior Rules* will include the possible transportation routes to other cities. One of the *Behavior Rules* parameters that can be updated based on the diffusion elements (people who move between cities) may be the number of people in the city, the number of planes that have departed on time, etc. Regarding *Sending*

and *Receiving Behavior*, *Sending* using airport may represent the process of preparing a plane before it takes off.

In the case of diffusion of rumors in Social Networks, the *Node* model will represent a person. The number of *Switches* (inside Indirect *Links Switches*), *Receiving* and *Sending Behavior* models will be fixed by the number and type of Social Network the person has access to (e.g. Facebook, Instagram, etc.). The *Behavior Rules* will include the possible Social Media or In-situ connections that a person can use with any of their contacts. One of the *Behavior Rules* parameters that can be updated based on the diffusion elements (rumors) may be the opinion of a friend, the connections in the Social Networks, etc. Regarding *Sending* and *Receiving Behavior*, *Sending* using Facebook may represent the process of the person posting an actualization on the Social Network. The model can be as simple as output the message or a more complex one if the person checks that the connection really works and waits until she sees the post on the screen.

6.3.2. Indirect Link

The *Indirect Link*, presented in Figure 25, is a Coupled model (see Appendix Cfor the formal definition) that provides a generic structure for communication of information. It contains three *Filters*, as many *Link Type* models as input or output types of links a node in the Network model has, and a *Sink*. For each node in the Network model, the *Indirect Link* is instantiated based on the types of indirect links each node has.

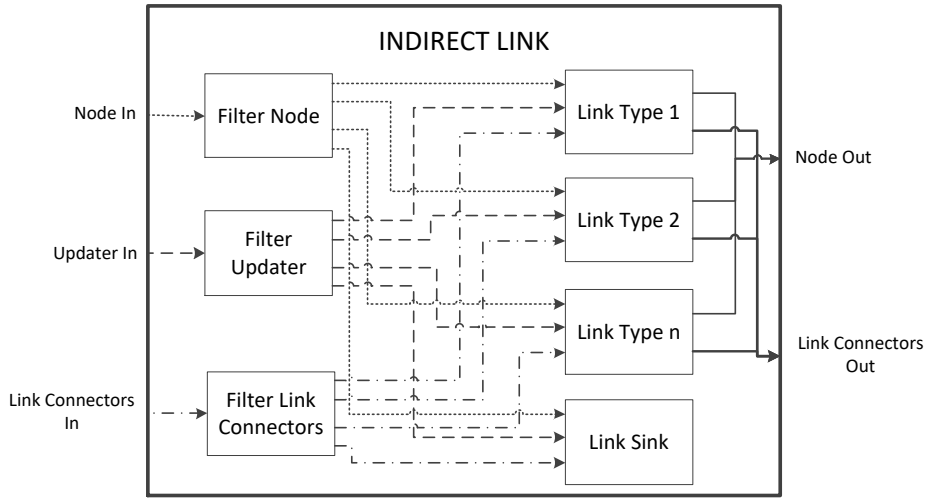


Figure 25. *Indirect Link* Coupled model

Link Type "i" represents the agent's object "i" to transmit the diffusion element. Each agent's object is translated to an atomic or coupled model based on its complexity and the level of detail needed.

If we use the same examples discussed in the previous section, if we are studying the diffusion of people in a transportation system, the *Link Type 1* may be a DEVS model of an airport, *Link Type 2* a DEVS model of a bus station, etc. In the case of diffusion of rumors in Social Networks, the *Link Type 1* may be a DEVS model of communication using Facebook, *Link Type 2* a DEVS model of Instagram, etc.

The three filters are Atomic models that redirect the diffusion elements (rumors in the case of social networks and people in the case of transportation system) or the updates in the properties of the indirect links types to the appropriate DEVS model. The three filters have as many ports as the total number of indirect link types in the Network model (which is equal to the number of different objects an agent may possess in the Agent Based model)

The *Sink* Atomic model collects all the Messages that arrive at an *Indirect Link* coupled model that do not have a matching *Link Type* model. It is connected to all output ports of the filters that remain unconnected after instantiating the coupled model.

The advantage of the proposed coupled structure for *Indirect Link* DEVS model is that, when implementing the model, the different instances of the coupled model (as many as *Node* Models) can be automatically instantiated based on the connections defined in the Network model.

6.3.3. Link Connectors

The *Link Connectors* Coupled model presented in Figure 26 is defined as a generic structure with two *Filters* and as many *Link Connectors* as object connectors are identified in the Agent Based model. In the Diffusion Abstract model, there is a single instance of the *Link Connectors* coupled model.

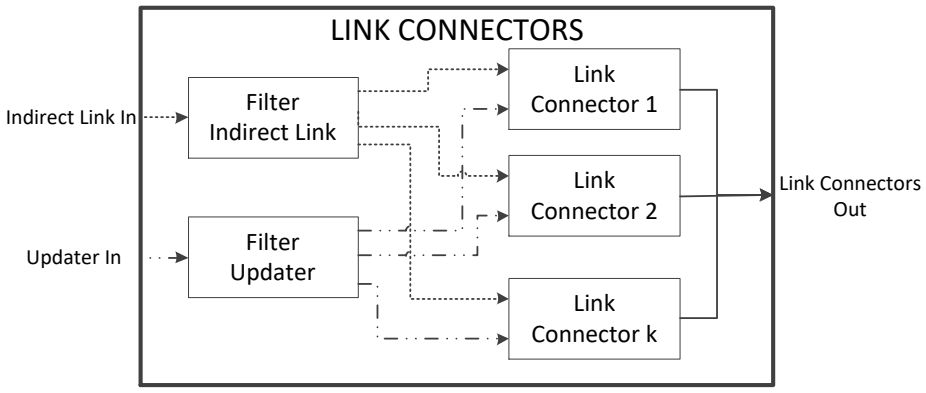


Figure 26. *Link Connectors* Coupled model

Each type of object connector in the Agent Based model is translated into a DEVS model. They can be either coupled or atomic models depending on its complexity and the level of detail needed. In the case of the transportation system of Section 3, *Link Connector 1* may represent the roads, *Link Connector 2* the railways, etc. In the case of a Social Networks, we could have a single *Link Connector*: the Internet (in that case, the filters are not used).

As in the previous case, the filters Atomic models redirect the diffusion elements or the updates in the properties of each Link Connector to the appropriate DEVS model.

6.3.4. Direct Link

A *Direct Link* is a DEVS model (atomic or coupled) that represents a direct connection between *Node* models. In the *Diffusion Abstract model*, there is a single instance of *Direct Link*. In our transportation system example, this model would not be used as cities are always connected by roads,

railways, the air, rivers, etc. In the case of Social Networks, it could represent the direct connection between people in the same location, who can talk in-situ.

6.3.5. Diffusion Element Generator

The *Diffusion Element Generator* is an atomic model that generates the elements to be diffused over time. When the model is implemented, it parses a text file that contains this data. The advantage of having the data in a text file is that different scenarios can be simulated just updating the text file without any modification in the model implementation.

In our transportation system example, it is a model that generates individuals that want to travel from one city to another including their preferences in means of transportation. In the case of a Social Networks study, it generates the rumors to be diffused and the nodes where they are created.

6.3.6. Updaters

There are four *Updater* models as we can see in the *Diffusion Abstract model* in Figure 23. The *Direct Link*, *Indirect Link*, and *Link Connectors* Updaters are three Atomic models that generate updates in the properties of the models that they are connected to. The models parse a file with this data (allowing us to simulate different scenarios by just updating the file without any modification in the model implementation).

As explained in section 6.3.1, *Node* models store the connections in multiplex networks and *Indirect Links* define the type of links the agent can access. Using the *Node Updater*, we can send updates to the *Node* model in order to change *MyRelations.* Modifying this parameter, we change the connections in the network dynamically. We can also use *Indirect Link Updater* to send updates to the *Indirect Link* models. For example, we can deactivate the connections in a *Node* model (i.e. the node connections in a layer in the Network model) changing the state of the specific *Link Type* to inactive.

In our transportation system example, the *Updaters* may generate the closure of a road segment, the closure of an airport or a train station, etc. As the *Direct Link* is not included in this model, the *Direct Link Updater* is neither included. In the case of Social Networks, they may generate changes in Facebook configurations (e.g., sharing with the public instead of friends), the unavailability of Internet signal in an area, etc.

The *Node Updater* model (atomic or coupled) generates updates in the properties of the *Node* models based on external information and the previous properties of the *Node*. When the model is implemented, the external information is parsed from an external file.

In our transportation system example, the *Node Updater* may generate new routes between cities, such as a new flight. It may also cancel routes or reduce the train timetable between two cities. In the case of Social Networks, it may generate updates in in-situ connections. It may also generate new behaviors for the people based on external parameters not related to the diffusion process.

6.3.7. Diffusion Abstract model implementation

The implementation of the DAM is the DCM. Using the formal definition of the components of the DAM introduced in this section, and the services of CDBoost introduced in Chapter 3, we translate

the atomic models into a CDBoost implementation. Appendix D shows the implementation of Generator Filter formally defined in Appendix B.

The DCM is based on the *DAM* (i.e. the atomic and coupled models we defined) and the agent based model (i.e. the XML files where the behavior of agents is defined), which are used to translate the *DAM* into a Computer Model for CDBoost. Figure 27 shows a schema of this process.

In order to implement the coupled models, we first need to instantiate the atomic models inside them. To do so, we use functions that use the XML file we discussed earlier in the Agent Based model (Figure 22). The rules are written in a way that the output of the function contains all the code needed. The top-level model is built using a program that takes the XML files where the agents are defined, reads each XML file, and transforms them into a structure to generate the parameters of all the functions explained earlier. The output is a file with thousands of lines of code for CDBoost. This file includes all the atomic and coupled models instantiated, which, once compiled, generates the *NEP DCM* ready to generate results. This process is automated to study different network sizes and configurations without any reimplementation. Appendix E shows a general implementation of the DAM.

The functions to instantiate the atomic and coupled models depend on the application and the behavior and parameters defined in the XML file. In Chapter 8, we show how these functions are defined for the application presented in this thesis.

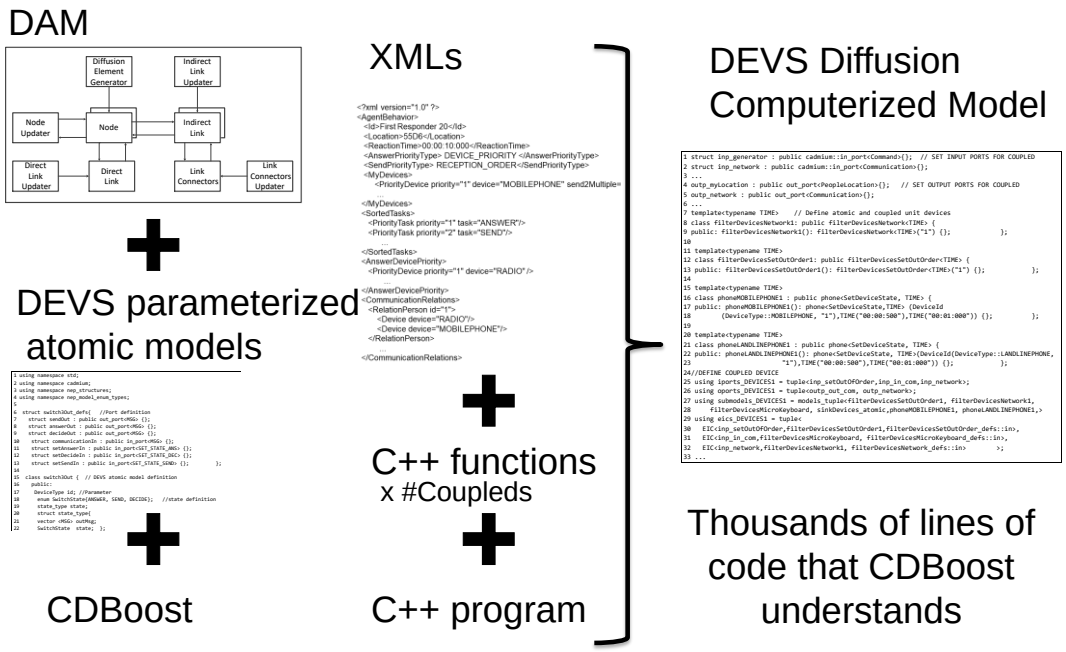


Figure 27. Scheme of the DCM implementation.

Once the DAM is implemented (i.e. we have our computerized model), CDBoost simulator engine and the computerized model are compiled together using any C++ compiler (e.g. GCC) to get the Diffusion Simulation.

Chapter 7. Case Study: Data Experiment Collection, Network and Agent Based Models for the Nuclear Emergency Plan

In this chapter, we present our case study discussing how to apply the *Data Experiment Collection* component of our architecture and how to use step 1 of the development processes.

As discussed in Chapter 6, the first step of the development process based on our proposed architecture consists of collecting data for all the requirements of the system of interest. In our case, this is a Plan coming from a Nuclear Power Plant (NPP) in Spain. The data was extracted from the plan that the Civil Protection Agency designed if an accident occurs in the NPP (due to a non-disclosure agreement, we cannot reveal the location of the NPP or other information, which is confidential; we discuss the main aspects that are important for the case study that can be shared with the public).

We then present the *Network* and *Agent Based models* of the NEP. We have a special emphasis on the process we follow to define them, so the reader can easily apply this to other problems

7.1. Data experiment collection

The NEP is a management plan that defines the structure and functions of a virtual Organization composed of different public organizations (such as the police, town halls, etc.) that are coordinated to solve the emergency. It also defines the tasks to be performed by every sub-organization and how they are related.

The data collection was done with the support of NEP experts that provided us with existing documentation of the NEP. We analyzed the documentation, extracted the data needed, and conducted follow-up meetings and interviews with the experts, following the procedures discussed in Section 6.2.1. We obtained a *Requirements Document* with a comprehensive definition of the NEP organization presented in (Ruiz-Martin 2013). In the rest of the section, we will briefly discuss the more relevant aspects of this document (which is 96 pages long), in order to show how it is used to define the Network model, Agent Based model and DAM. These aspects are the *organization structure*, the *communication systems* and the *rules for information transmission* defined in the NEP, which are discussed following.

7.1.1. NEP Organizational structure

Figure 28 shows a sketch of the organizational structure of the NEP. As we can see, there are different individuals in the organization. At the core, it is the NEP Director, who takes the decisions to manage and solve the emergency. However, as the emergency evolves, higher National Government ranks, such as the President, can take the position of the NEP Director.

The Nuclear Safety Commission (NSC) President, and the Central Response and Support Nuclear Emergency Plan (PENCRA) Director are at the same level as the NEP Director. The NSC Inspector at the NPP is in direct communication with the NEP Director. The Advisory Committee has to advise the NEP Director. The Information and Communication Cabinet is in charge of communications with the media. The Executive Body is composed of the leaders of several groups: Radiological, Health, Logistical Support, Public Security and Order, and Technical Assistance and Coordination group. The functions of the Executive Body are to get the commands of the NEP director done. Each group from the Executive Body has a predefined structure and functions. For example, the Radiological group is in charge of radiological control of the population and the first responders. The health group is in charge of the population and first responders' well-being and health. The Logistical Support Group is in charge of providing support for food, evacuation, and coordination. The Public Security and Order Group is in charge of the population safety and controlling the access to the emergency area. The Technical Assistance and Coordination group is in charge of managing the population for evacuation or confinement in the different municipalities. In each group, there are heads at different levels, people working in the emergency (first responders) and backup teams. Additional Team Leaders from other institutions can be asked to join the Executive Body.

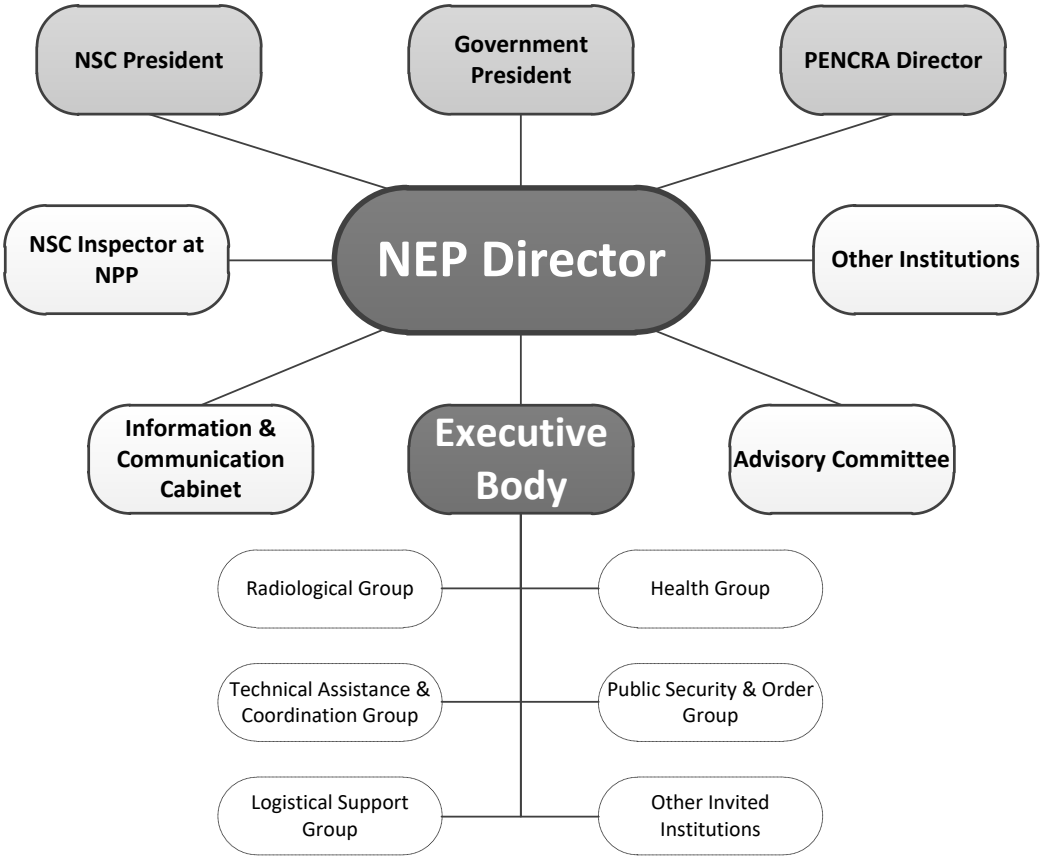


Figure 28. Sketch of the organizational structure of the NEP

The NEP also includes other institutions not directly involved in the emergency such as the Health Ministry, the Industry Ministry, etc. However, the NEP director has to keep them informed about the evolution of the emergency and ask for their help if needed.

As we discuss later on in Chapter 6, we use the organization structure to define the nodes in the Network model and the agents in the Agent Based model (they will be each person or team identified in this structure: the NEP Director, the Radiological Group chair, first responders, etc.).

7.1.2. Communication systems

The systems that handle the communications defined in the NEP include landline and mobile phones, fax, mixed radio/phone networks, satellite, Internet, Remer and Reman radio channels, and in-situ communication. The Health Group can also use beepers, although this is not fully defined in the NEP. Emergency managers and first responders can also use a military communication system. However, this is not available until the NEP Director makes a request and the entire infrastructure is deployed. Because of these characteristics, we do not consider it in our model; this is a backup plan in case everything else fails, and it will not be studied.

Every communication must follow the hierarchy defined in the NEP and the internal structure of each group. The most common communication system in the NEP is the *phone*: landline and mobile (the NEP does not provide information about who has access to each of them).

The following people have access to *Fax* and *Internet (e-mail)*:

- Staff in the Technical Assistance and Coordination group communication center.
- Staff in the town halls
- The NEP Director.
- The chair and vice-chairs of the Public Security and Order Group.

Anyone receiving a fax or email has to send an acknowledgment.

There are three independent *mixed radio-phone systems:* Radiological Group, Civil Guard, and Autonomous Police communication. Only the people in the group can use the system.

There also are two broadcast radio channels: Reman (Management Radio Network) and Remer (Emergency Radio Network).

In *Remer*, the communications are also broadcasted to the population. Therefore, it is not reliable to transmit confidential information. The following people can use the *Remer* radio channel:

- The chairs of local emergency plans in towns between 10-30 km from the NPP (Zone II)
- First responders
- Everyone that can use the *Reman* channel

The following people can use the *Reman* channel:

- The chair of the Technical Assistance and Coordination group
- The chairs of local emergency plans in towns at most 10 km away from the NPP (Zone I)
- One communication center of the Public Security and Order group
- The three communication centers of Technical Assistance and Coordination group

The following people can use a *Satellite phone:*

- The NEP Director
- The PENCRA Director
- The NSC President
- The Government President
- The city council of a single town located in Zone I.

As we explain in section 0, we use the requirements defined in this section to establish the relations between nodes (or agents). The communication systems are also used to identify the devices and the networks in the Agent Based model in section 7.3.

7.1.3. Communication Rules

The NEP defines every possible command (more than 30) to be handled in the case of an emergency. The NEP director selects what to do based on the evolution of the emergency. Table 3 shows the possible commands classified according to the emergency level (from level 0 to level 3). These levels indicate the state of the emergency, being level 0 a pre-alert situation and level 3 a general emergency. The set of command that can be used at each level also includes all the commands at lower levels.

Table 3. Summary of the commands to be handled in case of emergency

Level 0
Notify and verify the incident at the NPP
Establish Emergency Level 0
Request data about the state of the emergency
Level 1
Evaluate the available data to determine the category of the emergency
Establish Emergency Level 1 and activate every group in the NEP
Track communications in the NEP
Track the evolution of the emergency at the NPP
Ask first responders to show their accreditation, and classify them into working groups
Ask substitute teams to start working on the emergency. Tell the ones working to rest
Give first responders the materials they need to help in the emergency
Tell the population about the situation of the emergency
Establish controls to track and limit the people who go inside the emergency area
Evacuate schools in the emergency area
Evacuate visitors in the emergency area
Track the people who leave the emergency area
Tell first responders to protect themselves against radiation
Save and rescue people in dangerous situations such as fires
Provide health care and social assistance to people with severe problems such as disabled people
Level 2
Establish Emergency Level 2
Integrate the extraordinary resources needed in the emergency
Verify the safety and security in the emergency area (e.g. protect against looting)

Radiological Prophylaxis (Tell people to take medication to protect themselves against radiation)
Control the radiation in water and food
Put animals in shelters
Tell people to stay at home
Evacuate high-risk people (elderly, disabled, sick people in hospitals, etc.)
Level 3
Classify people based on their exposure to radiation and decontaminate them
Evaluate the state of the infrastructure and any other resources and decontaminate them
Classify animals base on their exposure to radiation and decontaminate them
Evacuate everyone in the emergency area and give them a place to stay
Control the exposure of first responders to radiation

The people working in the NPP determine the category of the emergency for management purposes. The category is based on the amount of forecasted radiation that could be released during an event. Using this information, the NEP Director selects the emergency level. The emergency level determines the set of commands that the NEP Director will use to protect the population or managing the emergency.

As we can see in the table, the NEP director can issue three different types of commands at level 0, 17 at level 1, 25 at level 2 and 31 at level 3. Each of these commands has a set of associated actions to protect the population. For each command, the NEP specifies how they should be transmitted until they arrive at first responders. At the end of this section, we explain in detail two commands: "Establish emergency level 0" and "Tell people to stay at home".

In the *Requirements Document*, some of the commands are defined using natural language while others are defined using a set of three tables: communications, acknowledgments, and actions. The definition of each command includes the set of actions each first responder must do when they receive the command.

As we will show in section 7.3, we use these rules to define attributes of the agents' behavior. Having the rules in table format facilitates this transition. This information is not needed to develop the Network model. The messages transmitted in the Network and Agent Based models (i.e. the diffusion elements) are the commands shown in Table 3 and their acknowledgments.

Let us show two examples of the commands listed in Table 3. "Establish emergency level 0" (which, in the *Requirements Document* has been defined using natural language) is used to tell those involved in the emergency plan to be on alert, as their services may be required. As we can see, it is a combination of different tasks, as follows:

1. The NEP Director sends the message "Establish emergency level 0" to all of the members of the Executive Body, the Advisory Committee and the Information and Communication Cabinet.
2. Then, the NEP Director send "Establish emergency level 0" to the PENCRA Director, the NSC President, the Government President, the NSC Inspector at the NPP, and other institutions (e.g. Health Ministry, Industry Ministry, etc.), in this particular order. The NEP does not consider that the message can be broadcasted to everyone at the same time. Each

member of the Executive Body sends the command to everyone in their group, following the hierarchy.

3. The members of the Executive Body must acknowledge to the NEP Director that "Establish emergency level 0" has been distributed of among their groups.

There are no actions associated with this command: only transmission of messages.

The command "Tell people to stay at home" is defined in the *Requirements document* using detailed information in tables, which has been summarized in Table 4-Table 6. Table 4 summarizes the behavior of the individuals, and how they transmit messages across the NEP structure. It includes a sender, a receiver and a type. The field type can be *information* or a *command* based on what each individual is expected to do. It will be *information* if the person or their supervisees do not need to do a specific action to solve the emergency and *command* otherwise. The type field can also include a specific device (like Fax Command) or define the message as optional.

Table 4. "Tell people to stay at home" command

Sender	Receiver	Type
NEP Director	Health Group Chair	Information
NEP Director	Public Security and Order Group Chair	Command
Public Security and Order Group Chair	Public Security and Order Group Vice-chair	Command
Public Security and Order Group Chair	Public Security Responsible	Command
Public Security Responsible	Public Security Manager in town 1	Command
Public Security Manager in town 1	Public Security First Responders in town 1	Fax Command
…	…	…

Table 5 specifies who needs to acknowledge the reception of the command. We identify the sender, the receiver and in the notes fields, any additional information regarding how the confirmation should be done.

Table 5. "Tell people to stay at home" acknowledgment reception

Sender	Receiver	Notes
Chairs of local emergency plans	Town Coordination Service Responsible	
Town Coordination Service Responsible	Coordination and Technical Assistance Group Chair	Confirm which chairs have sent the acknowledgment and which not
Coordination and Technical Assistance Group Chair	NEP Director	Confirm which chairs have sent the acknowledgment and which not
…	…	…

Table 6 shows the behavior with respect to the execution of actions. We identify all the actions to be performed for the command "Tell people to stay at home". For each action, we define the implementer, the estimated execution time, if an acknowledgment of completion is needed, and any additional information. If a table field is empty, it means that we do not have that data.

Table 6. "Tell people to stay at home" actions

Tell the population to stay at home			
Implementer	Estimated time	Acknowledgment?	Notes
First responders in town halls	--	No	Loudspeakers are used
Assure that nobody is on the street			
Implementer	Estimated time	Acknowledgment?	Notes
First responders from the police group	--	No	Only the responders in towns where people should be confined
…	…		…

7.2. Network model definition

The next step, after the *Requirements Document* is completed, is to use the *Requirements Document* to model the transfer of information in the NEP as diffusion in a multiplex network. To build the network, we use the following information from the *Requirements Document*: the people involved in the NEP (defined in section 7.1.1), and the systems they can use to communicate with each other (defined in section 7.1.2). The messages transmitted inside the network (i.e. the diffusion elements) are the commands and acknowledgments explained in section 7.1.3.

In the *Network model*, the nodes represent the people involved in the NEP and the links the relations between people. Each type of link represents a communication system. There are 832 nodes and 12 types of links (fax, the Internet, landline phone, mobile, satellite, Reman radio channel, Remer radio channel, Civil Guard radio-phone, Radiological Group radio-phone, Autonomous Police radio-phone, in-situ communications, and Beeper).

Figure 29 shows one of the multiple representations of the network at the beginning of the emergency (i.e. the same network can be graphically represented with the nodes arranged in different positions). We chose this representation because can see the groups working on the emergency as defined in the *Requirements Document.*

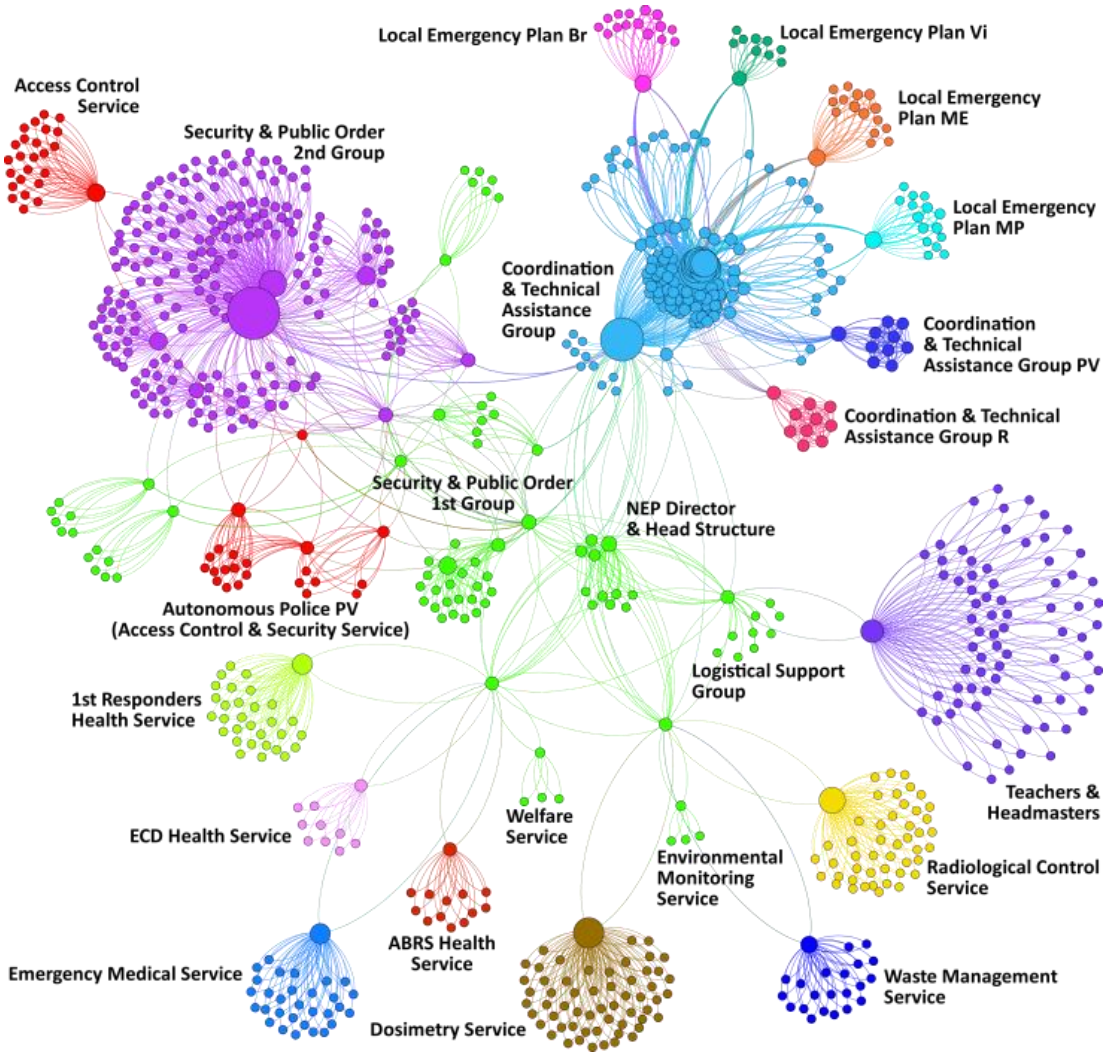


Figure 29. NEP network at the beginning of the emergency (Ruiz-Martin, Ramírez Ferrero, et al. 2015).

One of the main objectives of the study is to find problems with the current scheme; one problem we found at this stage and using this model, is that the *Requirements Document* is incomplete, and the experts were not able to provide more data. Therefore, when modeling the system we needed to make several assumptions (for instance, that people in an office have a mobile and a landline phone, that first responders do not have a landline phone, etc.). All the assumptions were approved by the experts and added to the *Requirements Document* as we explained in the development process. As we can see, one of the advantages of building the models and of running simulations is a better system specification using an iterative development process.

The network model is stored in two tables. Table 7 contains all the nodes in the network. Each node, represented as a row, has an id and a label. Table 8 contains all the relations between nodes. A relation, represented as a row, is defined using three fields: Source (i.e. source node), Target (i.e. target node) and Label (i.e. type of link that handles the relation).

Table 7. Nodes Ids and labels

Id	Label
1	NEP Director
2	PENCRA Director
3	NSC Inspector at the NPP
4	NSC President
5	Government Presidency
…	…

Table 8. Network connections

Source	Target	Label
16	323	Fax
16	324	Fax
16	325	Fax
16	326	Fax
325	368	Fax
325	369	Fax
1	2	Landline Phone
1	3	Landline Phone
1	4	Landline Phone
1	5	Landline Phone
…	…	…

Defining the network model as a table has two advantages: we can easily translate the model to other formats such as CSV or XML, and we can import it to different software applications to analyze the network. We analyzed different properties of the network such as the network diameter, the degree distribution, the average path length, etc. using Gephi (Ruiz-Martin, Ramírez Ferrero, et al. 2015). We used a simplex network because Gephi does not support multiplex network analysis. Moreover, as explained in Chapter 3, when this analysis was done, the tools that support multiplex network were under development, and the characterization of the properties of multiplex networks is still an open research field. We found that some relations between people on the Health and Logistical Support groups were only handled by a single communication system. We considered that communications handled by a single communication system are not resilient since a failure in this system may cause the isolation of agents, specific group services, or whole groups. In (Ruiz-Martin, Lopez-Paredes, et al. 2015), as we detail in Chapter 9, we used the network model to study how a failure in different communication systems affects the network connectivity and therefore the resilience information transmission process.

7.3. Agent Based model

Using our Architecture, we can build the *Agent Based model* in parallel to the *Network model*. Using the *Agent Based model*, we are able to include the behavior of the people regarding the information reception and transmission processes.

As we mentioned in Chapter 6, the *Agent Based model* represents the behavior of the people involved in the plan (modeled as agents), the devices (modeled as objects) and the networks that connect the devices (modeled as objects). In our Architecture, the Agent Based model should be built in parallel to the Network model since there is a correlation between the elements of both models. The nodes in the Network model are the agents in the Agent Based model and the types of links the devices. The networks that connect devices are not represented in the network model.

To define the behavior of everyone involved in the NEP, we first define the relevant characteristics of the behavior of a person for our problem. We then used the organizational structure, the communication systems and the communication rules defined in the *Requirements Document* to complete these characteristics.

In our model, some characteristics are attributes (i.e. they are completely defined in the *Requirements Document* and they remain constant for every analysis we do on the model) and other are parameters (i.e. they are not completely defined in the *Requirements Documents*, and we vary them to study their effect in the model)

In our model, the behavior of each agent includes the following characteristics:

- **Id (attribute):** identifies the agent, based on the organizational structure (e.g. NEP Director, Radiological Group Chair, etc.)
- **Location (dynamic attribute):** represents the location of the agent. When the emergency starts, their predefined actions determine their initial location.
- **Reaction Time (parameter):** indicates how long it takes to react to a stimulus.
- **Answer Priority Type (parameter)**: identifies the priority of the agent to receive a command. It can be based on who is sending the command, on the device that is receiving the message or a random priority.
- **Send Priority Type (parameter)**: identifies how the agent chooses the commands s/he will send. Their priority can be based on a priority list, on arrival time or a random decision.
- **My Devices (attribute)**: identifies the devices of each agent. We identify the agent's devices based on the communication systems defined in the Requirements Document. For each device, we define the relative priority of the device for the agent (parameter), its type (attribute), if it can broadcast/multicast (attribute) and if it is half/full duplex (attribute).
- **Prioritized Task (parameter)**: indicates how the agent sorts the tasks they do during the emergency.
- **Answer Device Priority (parameter)**: indicates how the agent prioritizes the response to commands based on the devices.
- **Answer Person Priority (parameter)**: indicates how the agent prioritizes the reception of commands based on who is sending it.

- **Send Command Priority (parameter)**: indicates how the agent prioritizes the commands they have to send. Each command in the list includes priority, destination, and content. For some agents, this it is defined in the *Requirements Document* (e.g. when the NEP director must "Establish emergency level 0"); for others, is not specified.
- **Action Execution Priority (parameter)**: indicates how the agent prioritizes the actions.
- **Communication Relations (attribute)**: is a list of the connections of the agent. Each connection has two attributes: target and device. This attribute is a direct translation from the Network model shown in Table 7 and Table 8. We use the agent's Id to retrieve the node Id in Table 7. We select all the rows in Table 8 with Source equal to node Id. The selected rows define the Communication Relations of the agent.
- **Message Behavior (attribute)**: defines the messages the agent has to send based on the messaged received. It is defined using the communication rules identified in the *Requirements Document*. For each command or acknowledgment the agent can receive, there is a list of messages to be sent (including those in Table 4 and Table 5) and a list of actions to do (from Table 6). Each command and acknowledgement includes (1) destination, (2) content (e.g. "Tell people to stay at home") (3) if sending the message is mandatory or optional, (4) if the command can be broadcast/multicast (5), if there is a mandatory device to use, and (6) if the agent should receive an acknowledgement (from Table 5).
- **Action Behavior (attribute)**: defines the set of actions the agent can do to solve the emergency. It is defined using the actions defined in the *Requirements Document* (Table 6). Each action includes: (1) average execution time (in the field), (2) location (defined in the field notes or implicit in the description of the action) and (3) messages to send (with the same attributes already defined in Message Behavior). In this case, the meaning of the message is an acknowledgment of the completion of the action.

We use an XML file (Figure 30) to store the behavior of the agent. As we can see, we use XML tags to define each of the parameters and attributes described in the previous paragraphs (and their values are defined as the content of the tags).

```
1 <?xml version="1.0" ?>
2  <AgentBehavior>
3   <Id>First Responder 20</Id>
4   <Location>55D6</Location>
5   <ReactionTime>00:00:10:000</ReactionTime>
6   <AnswerPriorityType> DEVICE_PRIORITY </AnswerPriorityType>
7   <SendPriorityType> PRIORITY_LIST </SendPriorityType>
8   <MyDevices>
9        PriorityDevice priority="1" device="MOBILEPHONE" send2Multiple="false"
10           sendSeparateFromReceive="false"/>
11          ...
12  </MyDevices>
13  <SortedTasks>
14     <PriorityTask priority="1" task="ANSWER"/>
15     <PriorityTask priority="2" task="SEND"/>
16          ...
17  </SortedTasks>
18  <AnswerDevicePriority>
19     <PriorityDevice priority="1" device="RADIO" />
20           ...
21  </AnswerDevicePriority>
22  <AnswerPersonPriority>
23     <PriorityPerson priority="1" id="1"/>
```

```
24          ...
25      <PriorityPerson priority="3" id="97"/>
26   </AnswerPersonPriority>
27   <SendCommandPriority>
28      <PriorityCommandTo priority="1" to="1" msg=" Tell population to stay at home" />
29              ...
30     <PriorityCommandTo priority="3" to="97" msg=" Tell population to stay at home" />
31   </SendCommandPriority>
32   <ActionExecutionPriority>
33      <PriorityAction priority="1" id="Tell population to stay at home"/>
34   </ActionExecutionPriority>
35   <CommunicationRelations>
36      <RelationPerson id="1">
37          <Device device="RADIO"/>
38          <Device device="MOBILEPHONE"/>
39      </RelationPerson>
40      <RelationPerson id="5">
41          <Device device="BEEPER"/>
42      </RelationPerson >
43           ...
44   </CommunicationRelations>
45   <MessageBehavior>
46      <MsgReceived from="1" content="Tell people to stay at home">
47           <Msg2Send to="5" content="Tell people to stay at home acknowledgement"
48       compulsory="true" send2Multiple="false"/>
49           <Action2Do id=" Tell population to stay at home "/>
50      </MsgReceived>
51   </MessageBehavior >
52   <ActionBehavior>
53      <Action id="Tell population to stay at home">
54           <AverageExecutionTime time="00:10:00:000"/>
55           <Location>55D6</Location>
56           <Msg2Send to="1" content=" Tell population to stay at home completed”
57       compulsory="true" send2Multiple="false">
58                <Device device="BEEPER"/>
59           </Msg2Send>
60      </Action>
61   </ActionBehavior>
62 </AgentBehavior>
```

Figure 30. Example of the agent's definition using XML.

The behavior of each agent is defined between the tags <*AgentBehavior*>. The agent includes all the parameters and attributes explained above, with tags *Id*, *Location*, *ReactionTime*, *AnswerPriorityType* and *SendPriorityType* as above. The value inside the tags represents the value of the attribute. Here, we have First Responder 20, located in position 55D6. Their reaction time is 10 s, and they prioritize the reception of commands based on the device they came from. For sending commands, they have a priority list.

MyDevices includes all the devices the agent has; as many elements as devices. Each device is represented as a tag (*PriorityDevice*) with the four attributes explained above (*priority*, *device*, *send2Multiple*, *sendSeparateFromReceive*). In this example, the agent has three devices.

SortedTasks represents how the agent sorts the tasks they should do under emergency. They may have equal or different priorities. In this example, the highest priority is for *ANSWER* (i.e. receive a command or acknowledgement from someone who is requesting a communication; e.g. answer the phone that is ringing), then *SEND* (i.e. transmit a command or an acknowledgement to another person; e.g. send a fax to person 1 that states "Tell people to stay at home"), etc.

AnswerDevicePriority can have as many entries as devices. If all devices have the same priority for receiving commands, it is empty. All the devices not included in this tag have the lowest priority. Each element has a priority and a type.

AnswerPersonPriority has as many elements as individuals the agent has relation with. Each element has two attributes: *priority* and *id*. In this example, receiving a message from person 1 has the highest priority. Person 97 has priority 3.

SendCommandPriority classifies the set of messages the person may send during an emergency. Every element has three attributes: a *priority*, a receiver (*to*) and the content of the message (*msg*). In this example, transmitting "Tell population to stay home" to person 1 has high priority. Transmitting "Tell population to stay home" to person 97 has priority 3.

ActionExecutionPriority has two elements: *priority* and an *id* for the action. In this case, "Tell people to stay at home" has the highest priority. The rest of the actions have the same priority.

CommunicationRelations identifies the relations with the different individuals. It has one element per individual the agent is connected to. Each individual is identified with the tag *RelationPerson*, with an attribute *id* that represents the person the agent is connected to. It also has as many elements as the number of devices the agent can use to communicate with this person. Each element is identified with the tag *Device*. This tag only has the attribute *device* that identifies the type of devices. In the example, First Responder 20 can communicate to person 1 using radio and mobile phones. They can communicate to person 5 using a beeper.

MessageBehavior represents how the agent behaves when they receive messages. It has as many elements as combinations message-person that they can receive. Each message is identified with the tag *MsgReceived* with two attributes: the sender (*from*) and the *content*. Each *MsgReceived* has as many elements as actions they must do (*Action2Do*) and combinations of command-person to send (*Msg2Send*). Each *Msg2Send* has the four attributes explained at the beginning of the section. In this example, the person has to send the message "Tell people to stay at home acknowledgment" to person 5. The message is compulsory and cannot be broadcasted or multicasted. They also have to do the action "Tell the population to stay at home"

Finally, *ActionBehavior* identifies how the person should behave when doing an action. It has as many elements as actions the person can do. Each action has an attribute (id) and at least two elements (*AverageExecutionTime*, with time, and *Location*). It may also have some *Msg2Send* elements. In this example, the agent can do a single action: "Tell population to stay at home". The average execution time is 10 minutes and it is done in the location 55D6. When the action is finished, the person has to send a message with the content "Tell population to stay at home completed" to person 1. The message is compulsory and cannot be broadcasted or multicasted. They must use a beeper to send the message.

As we have already said, we model the behavior of the devices (e.g. phone, radio, etc.) as objects. For each device, we identify the set of possible states and its behavior in each state. The device may also have different parameters such as the probability of being in a certain state and the delay introduced in the communication.

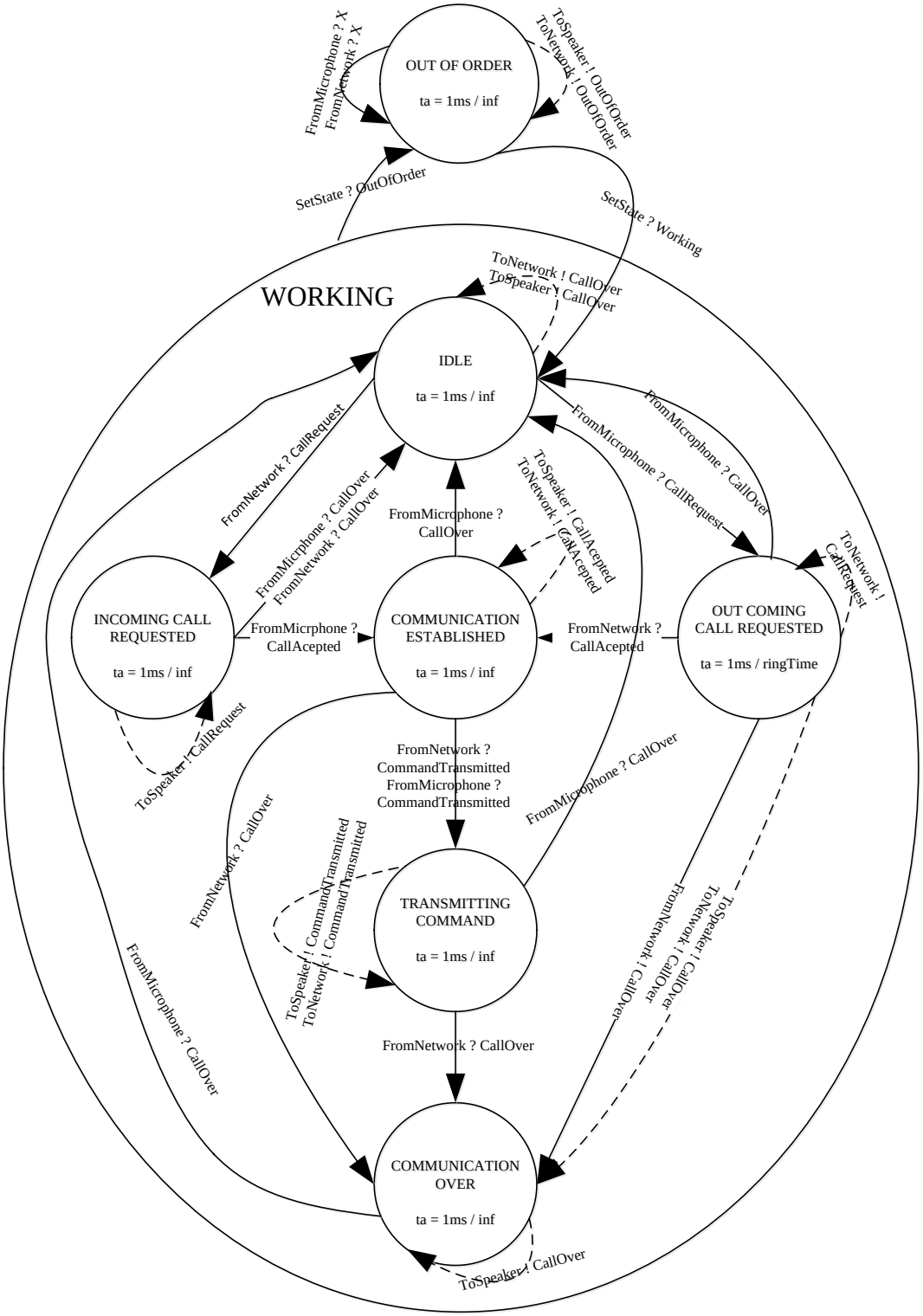


Figure 31. Diagram of the Mobile states using DEVS-Diagram notation.

For instance, for the Mobile model, shown in Figure 31, we have two parameters: ring duration and delay, and one attribute (id). The Mobile can be in seven states: *idle*, *incoming call requested*, *outcoming call requested*, *communication established*, *transmitting the command*, *communication over* and *out of order*.

We also model the networks (Internet, different radio frequencies, phone network, etc.) as objects. In this case, the focus is on whether or not a message is transmitted. Therefore, we defined them as objects with two parameters: a probability of transmitting the message and a delay. Each network can be in two states: active or broken. If the network is active, it will transmit the message with a certain probability. If it is broken, it will not transmit the message.

Chapter 8. Case Study. Diffusion Abstract Model of the Nuclear Emergency Plan using DEVS

In this chapter, we detail how to obtain the *NEP Diffusion Abstract model* and its computerized version using the general DEVS implementation proposed in section 6.3

8.1. Diffusion Abstract model for the Nuclear Emergency Plan

Once the *Agent Based model* is completed, the next step of the development process is to define the *Diffusion Abstract model* we introduced in section 6.1.4, and apply it to the NEP. In our case, we design the *Diffusion Abstract model* by instantiating the general DEVS implementation introduced in section 6.3 and combining it with the *Agent Based model* presented in section 7.3. This is summarized in Figure 32

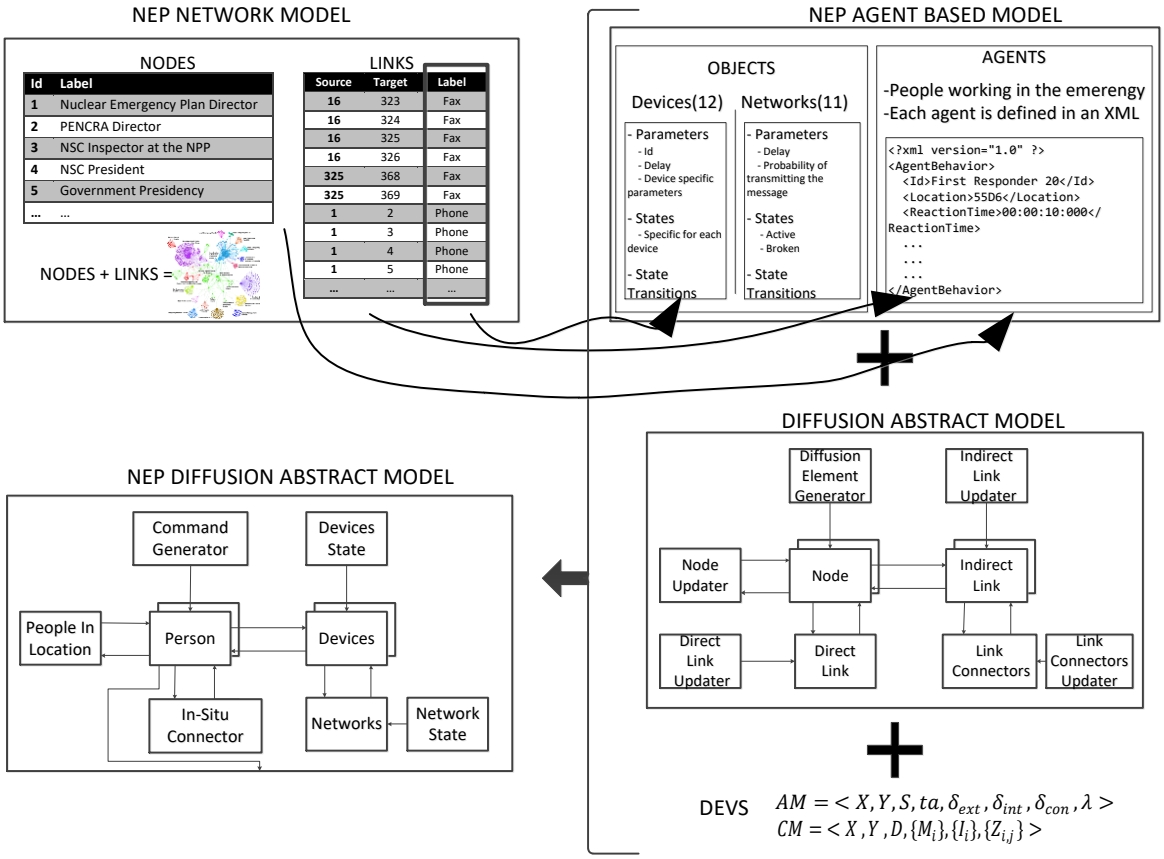


Figure 32. Schema of the *NEP Diffusion Abstract* model definition for the NEP.

We first use the *Agent Based model* to decide which components of the architecture are needed. A *Node* in our architecture corresponds to an agent of the *Agent Based model* (which is derived from a node in the *Network model* and it includes the connections where the id of the source node is the agent). Using our *Requirements document*, this corresponds to each person working on the emergency.

Likewise, the *Indirect Link* model in the architecture will be converted into the devices each person can use, which is a subset of all the existing devices (also defined in the *Agent Based model* and derived from the different labels in the *Network model*). Similarly, the *Links Connector* is mapped into the Networks connecting such devices (Internet, telephone network, etc.), each of which is also defined as an object in the Agent Based model. The *Direct Link* connector in the architecture allows connecting *Nodes* by a direct link. In this case, it represents connections in-situ. It does not have a direct translation from the *Agent* or *Network model*. The DEVS modeler defines its behavior and the level of detail needed based on the purpose of the Model. The *Updaters* (for *Indirect Links, Nodes and Links Connectors*) can introduce changes in the state of the devices and networks models (i.e. they model if they break or recover) and the persons involved in the NEP. We have mapped them into different models for the NEP. In our case, the only parameter of the *Node* that we are interested to update is the individuals that are within the same location. This determines which in-situ communications are feasible, and it is updated using the actual location of all the agents. To instantiate the model, it is defined based on the *Agent Based model* Location attribute, and it changes over time as the agent moves. Finally, the *Diffusion Element Generator* is converted into the *Command Generator*, a model that generates the commands according to the *Requirements Document*. This will trigger the diffusion process, and the set of commands generated. How all these generator models influence the NEP study is defined by the scenarios we want to analyze.

As we can see, we have used every component in our Architecture except the *Direct Link Updater* (used to update the properties or states of the *Direct Link Connector* when there are external factors not included in the model). In our case, this represents in-situ communications, and we are not interested in updating it at runtime (which would include modeling detailed factors in in-situ communications such as environmental noise, distance, etc.). In our case, the *Direct Link Connector* will just broadcast the messages received.

As discussed earlier, the modeler needs to decide which will be the model output. In our case, we want to study what messages the person sends, which ones are received (including information about the communication channels used) and what action they conducted. Therefore, the output of our model is defined in the person model.

We define each component using the general implementation in DEVS explained in section 6.3, which is also based on the *Agent Based model* and the architecture. First, we need to define all the objects (i.e. devices and networks) as DEVS models. The parameters and states of each object are translated as parameters and states of the DEVS model. The State Transitions are converted into the internal and external transitions in the DEVS model. In our case, each network object (i.e. Internet, satellite, etc.) is defined as a DEVS model with the two parameters identified in the *Agent Based model*: the probability of transmitting the message and a delay. Every network DEVS model has two states: active or broken. The devices are also defined as parameterized DEVS models using the parameters defined in the *Agent Based model*. All of them include at least an *Id* and a delay. The *Id* parameter is used to instantiate the different devices used by different individuals. The rest of the parameters, states, and transitions are different for each device.

We use these Models (i.e. devices and network) to instantiate the *Devices* and *Networks* Coupled models of the *NEP Diffusion Abstract* model as we explain later on in this section.

In the rest of this section, we briefly explain how we instantiate the components of the *NEP Diffusion Abstract model* using the *Agent Based model* definition and the DEVS parameterized models of the devices, networks, sending and receiving behaviors. Since there are hundreds of *Persons* and *Devices* in the NEP, it is important to define a generic process that can be automated when implementing their computerized models. We focus on this automation, which takes the agents' XML files as inputs and generates a computerized model (see section 8.2).

8.1.1. *Person* model instantiation

The *Person* is a Coupled model that is instantiated following the *Node* architecture presented in section 6.3.1. As we can see in Figure 33, we use all the components of the architecture because our agents use both direct (i.e. in-situ) and indirect (i.e. devices) communications.

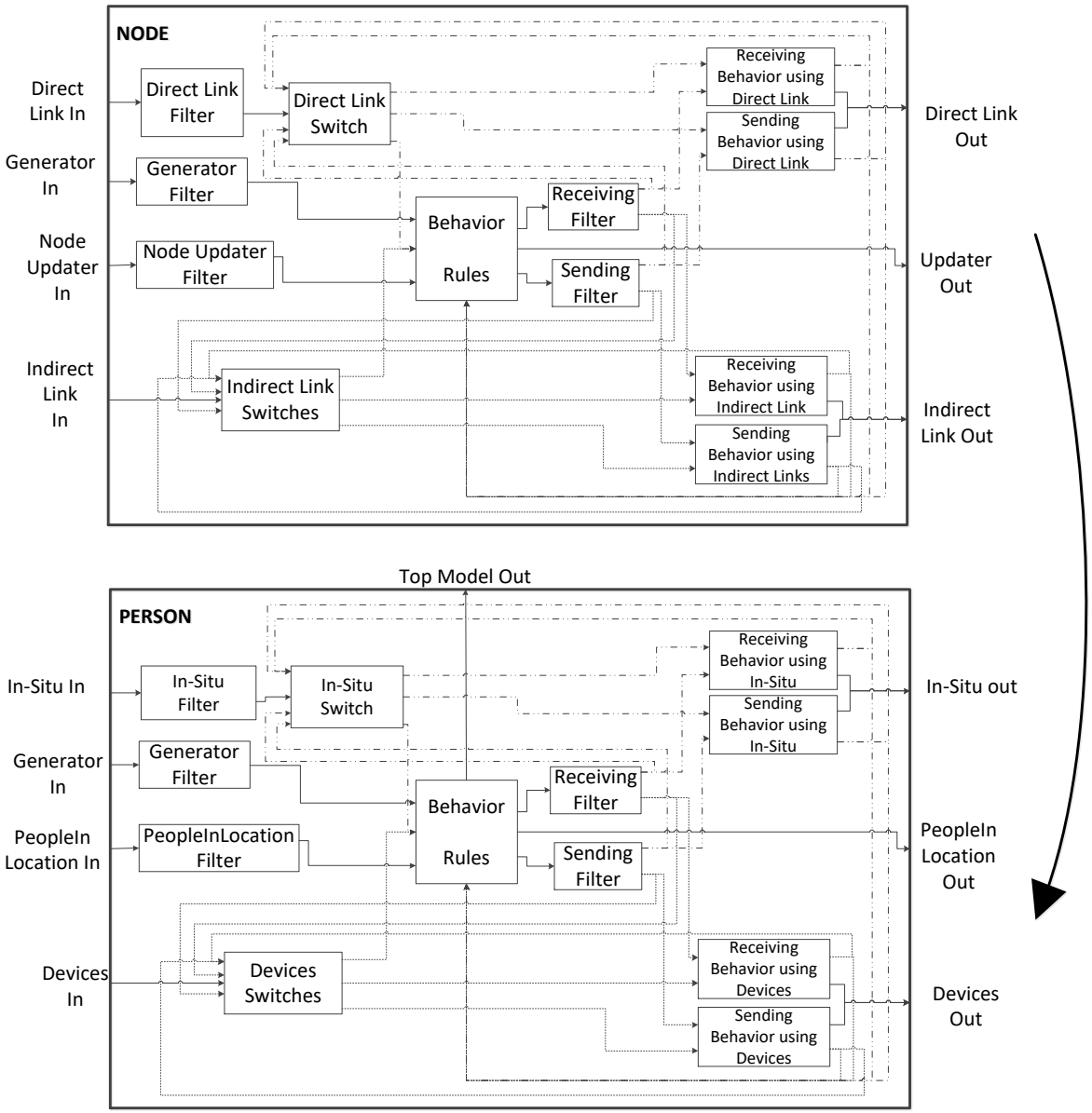


Figure 33. Coupled model definition of a *Node* and its translation to a *Person* model for the NEP.

The *Behavior Rules* is the main core of the model. It is defined as a coupled model (see Figure 34) that includes the actions the agent takes to solve the emergency. It is instantiated for each *Person* model using the agent defined in the XML file. All the parameters and attributes defined in the XML file are included in the *Behavior Rules* model. It models how the person takes decisions about the messages and how it executes tasks based on the behavior defined in the XML file, the devices that are activated, the people around, and a to-do list.

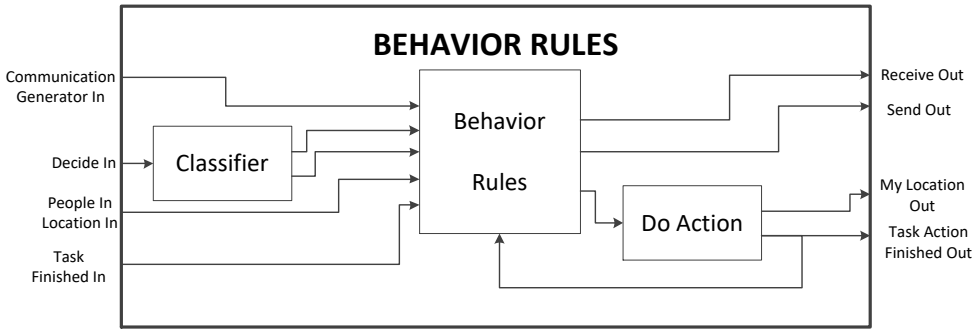


Figure 34. Coupled model definition of *Behavior Rules*

When there is only a task to do (i.e. just a device activated, just an action to do, etc.), the model sends an output that actives the component in charge of this task. If there is more than one task, *Behavior Rules* decides what to do based on the agent behavior defined in the XML file (Figure 30). Lines 13-17 define which type of task has priority: responding to a device that is activated or to a person that is trying to establish in-situ communication (from now on, called *answer*), sending a command or acknowledgement (from now on, called *send*) or executing an action to solve the emergency (from now on, called *do action*). In the example, the model will first respond to an active device or person talking in-situ. If there is more than one task to do in each category, the decisions are as follows:

- *AnswerPriorityType* (line 6) defines how to answer. The decision can be based on the type of device, on the person or at random. In this example, the decision is based on the device that is activated, and *AnswerPriorityDevice* (lines 18-21) defines the priority of each device. If the decision is based on the person who is calling, *AnswerPersonPriority* (lines 22-25) defines the priority.
- *SendPriorityType* (line 7) defines how the sending task is prioritized. It can be First In – First Out (FIFO), Last In – First Out (LIFO) or using priorities. In the example, it uses a priority list, defined in *SendCommandPriority* (lines 27-31).
- *ActionPriorityExecution* (lines 32-34) defines how do action is prioritized. The actions not included in the list have the same priority.

The commands and acknowledgments to be included in the sending to-do list and the actions to be included in the actions to-do list are select based on the messages received either through devices or in-situ. For each command/acknowledgement received, the model looks into *MessageBehavior* (lines 45-51) and extracts all the commands, acknowledgements and actions. They are included in their respective to-do list.

As discussed, the model can select among three tasks: *answer*, *send* or *do action*. If *answer* is selected, it generates an output through the receiving port with the active device or the person that is

requesting the communication. If *send* is activated, once the model has decided which message it should send, it chooses to send the message in-situ if the person is in the same location. If the person is in another location, the device is chosen based on the following parameters: (1) mandatory devices (*Msg2Send* in the XML file), (2) devices available (*MyDevices* – lines 8-12), and (3) devices that can be used with the receiver (*CommunicationRelations* – lines 35-44). The model output is a Message through the sending port that includes the receiver, the command/acknowledgement content and the device.

If *do action* is selected, once the model chooses the action, the information regarding the average execution time, the place where the action should be done and the acknowledgement to be sent when the action is completed are retrieved from the *ActionBehavior* attribute of the XML file (lines 52-59). The Behavior rules model executes the action.

The *filters* for *Direct Link*, *Generator* and *Node Updater* select which of the messages are used in the model, based on the Id of the agent defined in the XML file (as messages are broadcasted, they have to decide if they are the destination). The *Direct Link Switch* model is transformed into an *In-situ Switch* (the model's behavior is the same in both cases, although we renamed it to make it easier to study). The *Sending/Receiving Filters* also remain unchanged. The *Receiving/Sending Behavior Using Direct Link* models are mapped into parameterized Models using the agent *Id* attribute found in the XML file.

The *Indirect Link Switches* model is instantiated using the *MyDevices* attribute in the XML definition of the agent and converted to a *Devices Switches* Coupled model, which includes four filters, a sink, and as many atomic models as elements in *MyDevice* (as explained in section 6.3.1).

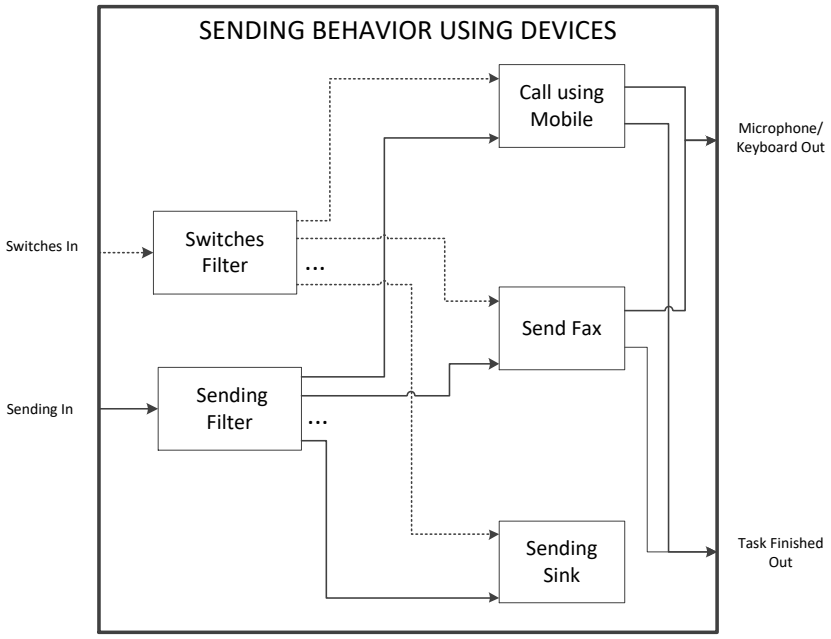


Figure 35. Example of *Sending Behavior using Devices* coupled model instantiated for a person with mobile and fax

The models *Sending/Receiving Behavior using Indirect Link* are instantiated using the *MyDevices* and *Id* attributes in the XML file and the Models defined for each device. The devices included in

MyDevices establish which Models we need to include. These are instantiated with the Id attribute. In Figure 35, we show an example of the *Sending Behavior using Devices* coupled model instantiated for a person with a fax and a mobile. The *Switches Filter, Sending Filter* and *Sending Sink* are common for all instantiations. The models that change from one instantiation to other are the ones representing the use of different devices.

All the remaining models in this section are built using a similar procedure. We will discuss the main features of each model.

8.1.2. Devices Coupled model instantiation

The *Devices* Coupled model is instantiated following the *Indirect Link* architecture presented in section 6.3.2. We use all the components of the architecture, which are instantiated using the *MyDevices* attribute of the XML file, and the devices parameterized Models using the agent *Id* attribute in the XML file. The devices Coupled model includes three filters, a sink, and as many devices as elements in *MyDevice* (as explained in section 3.3.2). In Figure 36, we can see an example for of a *Devices* coupled model with a mobile, a fax and a satellite phone.

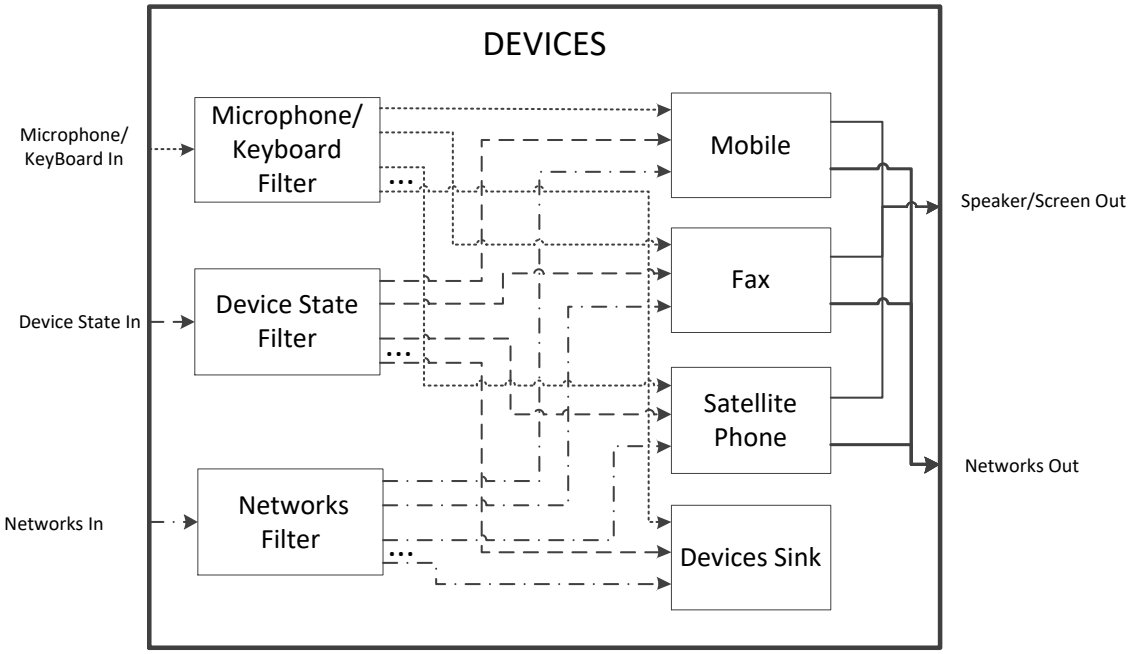


Figure 36. Example of *Devices* coupled model instantiated for a person that has a mobile, a fax and a satellite phone.

8.1.3. Networks Coupled model instantiation

The *Networks* Coupled model is instantiated following the *Link Connectors* architecture presented in section 6.3.3. In this case, we also use all the components of the architecture. It has two types of components, *filters*, and *Link Connectors*. The model has as many *Link Connectors* as networks in the Agent Based model. Each *Link Connector* is instantiated using a network parameterized model.

The model has two filters. Each filter has as many output ports as network types in the Agent Based model (11 ports in this case since there are 11 networks in our case study). Each output port

represents a network type (i.e. satellite network, Internet, etc.). Since the messages in our model are broadcasted, each filter redirect de message to the appropriated port based on the device that sends the message.

8.2. NEP Diffusion Computerized Model

As we explained in section 6.1.5, the *Diffusion computerized model* is a computerized model of the *Diffusion Abstract model*. We built the *NEP Diffusion computerized model* using the CDBoost DEVS simulator introduced in Chapter 3. The model is based on the NEP Diffusion Abstract model (i.e. all the atomic and coupled models we defined and their connections) and the agent definition (i.e. the XML files where the behavior of agents is defined), which are used to translate the NEP Diffusion Abstract model into a computerized model written in C++ that CDBoost understands.

Each atomic model defined in section 8.1 is implemented as a structure or a class as discussed in Section 6.3.7 using the template provided in Figure 4. Each Atomic model includes a set of ports, a set of parameters, a constructor that instantiates the model parameters and initializes the model variables and the five DEVS functions. These parameterized atomic models (37 in our case study) are used to instantiate all the atomic models included in the different coupled model using the XML files where the agents are defined and some functions we define in C++ as we explain later on. The model implementation is automated using a script that calls these functions to define the top model. The script returns the computerized NEP Diffusion Abstract model as a file ready to compile.

In Figure 37, we show the implementation of the *switch* atomic model used to instantiate the *Devices Switches* coupled model. The model redirects the message in the *DecideIn* port to the appropriate output port based on the model state. The model can be in three states (answer, decide, send). We use different output ports according to the model's state. We have chosen this example since it allows to fully explain the implementation and has simple functions. The rest of the atomic models are implemented using the same logic.

```
1 using namespace std;
2 using namespace cadmium;
3 using namespace nep_structures;
4 using namespace nep_model_enum_types;
5
6  struct switch3Out_defs{   //Port definition
7    struct sendOut : public out_port<MSG> {};
8    struct answerOut : public out_port<MSG> {};
9    struct decideOut : public out_port<MSG> {};
10    struct communicationIn : public in_port<MSG> {};
11    struct setAnswerIn : public in_port<SET_STATE_ANS> {};
12    struct setDecideIn : public in_port<SET_STATE_DEC> {};
13    struct setSendIn : public in_port<SET_STATE_SEND> {};         };
14
15  class switch3Out {  // Atomic model definition
16    public:
17     DeviceType id; //Parameter
18      enum SwitchState{ANSWER, SEND, DECIDE};   //state definition
19      state_type state; // State
20      struct state_type{
21      vector <MSG> outMsg;
22      SwitchState  state;  };
23
24      switch3Out(DeviceType Id) noexcept {  //constructor & initial state definition
```

```
25        id = Id;
26        state.outMsg.clear();
27        state.state = SwitchState::DECIDE;  }
28
29        void internal_transition() { // internal transition
30          state.outMsg.clear();    }
31
32       void external_transition(TIME e, typename make_message_bags<input_ports>::type mbs) {
33         int bug = 0;
34         if (!get_messages<typename defs::setAnswerIn>(mbs).empty()){
35           state.state = SwitchState::ANSWER;
36           bug++;    }
37         if (!get_messages<typename defs::setSendIn>(mbs).empty()){
38            state.state = SwitchState::SEND; //multiple call equal one call
39            bug++;   }
40         if (!get_messages<typename defs::setDecideIn>(mbs).empty()){
41            state.state = SwitchState::DECIDE; //multiple call equal one call
42            bug++;   }
43         for (const auto &x : get_messages<typename defs::communicationIn>(mbs)) {
44             if(x.to.type == id) state.outMsg.push_back(x);   }
45         if (bug > 1) throw std::logic_error("Contradiction to set states");
46       }
47
48          typename make_message_bags<output_ports>::type output() const { // output function
49            typename make_message_bags<output_ports>::type bags;
50              switch(state.state){
51                 case SwitchState::ANSWER:
52                    for (int i = 0; i < (state.outMsg.size()); i++)
53                      get_messages<typename defs::answerOut>(bags).push_back(state.outMsg[i]);
54                    break;
55                 case SwitchState::SEND:
56                    for (int i = 0; i < (state.outMsg.size()); i++)
57                      get_messages<typename defs::sendOut>(bags).push_back(state.outMsg[i]);
58                    break;
59                    case SwitchState::DECIDE:
60                      for (int i = 0; i < (state.outMsg.size()); i++)
61                       get_messages<typename defs::decideOut>(bags).push_back(state.outMsg[i])
62                      break;
63             }
64             return bags;
65         }
66
67         TIME time_advance() const { // time_advance function
68           return (state.outMsg.empty() ? std::numeric_limits<TIME>::infinity() : TIME());
69         }
70  };
```

Figure 37. Computerized model of the Switch Atomic model

We start by defining the CDBoost library name spaces we are using to implement the model. Then, we include the input and output ports, each of which is defined as a structure that inherits from the in_port/out_port structures defined in the simulator. Each port can transfer different types of data.

We then define the model as a C++ class: first, we include the model parameters (line 17), which, in this case, is an Id that identifies to which device the switch refers to (in a more complex model, there may be several). Then, we define the state as a structure named state_type (lines 18-22), with two variables: state (an enum that identifies the state of the switch) and *outMsg* (which stores the messages the switch needs to reroute). In lines 24-27, we define the model constructor. It is used to instantiate the parameters of the atomic model and initialize the state variables.

The next step is to implement the internal transition function (lines 29-30), the external transition function (lines 32-46), the output function (lines 48-65), and the time advance function (lines 67-69) translating the formal DEVS specifications into C++ functions that define the Switch's behavior. The model includes code to help with verification; for example, in the external function, we have defined that the switch cannot have more than one state at the same time. In the external function, we include a variable *bug= 0*. When the messages bags from the *SetDecideIn*, *SetSendIn*, and *SendAnswerIn* ports are processed, if they are not empty the variable bug is incremented. An error message is issued when that happens (line 45).

The internal transition function clears the *OutMsg* variable. The external function stores the messages received through *CommunicationIn* port in *OutMsg* variable. It also sets the value of the state variable based on the inputs in the other ports *SetDecideIn*, *SetSendIn* and *SetAnswerIn*. The output function sends the messages stored in *OutMs*g through the corresponding output port. Finally, the time advance function passivates the model if there is nothing to send, and it sets the time advance in 0 is there is something to send.

In the *NEP Diffusion Abstract model*, there are two types of coupled models: those instantiated a single time (e.g. *Networks*) or multiple times (*Person*, *Devices*, *Devices Switches*, etc.). The models instantiated a single time are implemented manually. The coupled models that have more than one instance are generated automatically.

In order to implement the coupled models, we first need to instantiate the atomic models inside them. To do so, we use one function for each type of coupled model, as seen in Figure 38.

```
1 create_atomics_text_msg_device(string DeviceType,string Id,TIME delay,TIME outOfOrderAcknow){
2  pair<vector<string>,vector<string>> AtomicsCoupled;
3
4  /***Instantiate atomics inside the coupled***/
5  create_atomic_inbox(DeviceType, Id, delay, outOfOrderAcknow);
6  string inbox = "inbox"+DeviceType+Id;
7  create_atomic_outbox(DeviceType, Id,delay, outOfOrderAcknow);
8  string outbox = "outbox"+DeviceType+Id;
9  create_atomic_ msgClassifierNewReadCon(DeviceType, Id));
10
11  /****Define coupled: first the I/O ports ****/
12  string("using iports_")+DeviceType+Id+string("=<inp_setOutOfOrder, inp_network,
13      inp_fromKeyboard>;"));   // input ports
14  string("using oports_")+DeviceType+Id+string("=<outp_toScreen,outp_network>;"));
15
16  string("using submodels_")+DeviceType+Id+string("= models_tuple<")+inbox+
17        string(",") +outbox+string(",")+msgClassifierNewReadCom+string(">;"));     // SUBMODELS
18
19  //External Input Couplings - eics
20  string("using eics_")+DeviceType+Id+string(" =std::tuple<"));
21  string("EIC<inp_setOutOfOrder,")+inbox+string(", inbox_defs<SetDeviceState>::setStateIn>,"));
22  string("EIC<inp_setOutOfOrder,")+outbox+string(",outbox_defs<SetDeviceState>::setStateIn>,"));
23  string("EIC<inp_network, ")+inbox+string(", inbox_defs<SetDeviceState>::newIn>, "));
24  string("EIC<inp_fromKeyboard,")+ msgClassifierNewReadCom+string(", msgClassifierNewRead_defs
25      <Communication>::in>"));      string(">;"));
26
27  //External Input Couplings - eocs
28  string("using eocs_")+DeviceType+Id+string(" =std::tuple<"));
29  string("EOC<")+inbox+string(", inbox_defs<SetDeviceState>::displayOut, outp_toScreen>,"));
30  string("EOC<")+outbox+string(", outbox_defs<SetDeviceState>::displayOut, outp_toScreen>,"));
31  string("EOC<")+outbox+string(", outbox_defs<SetDeviceState>::networkOut, outp_network>"));
```

```
32  string(">;"));
33
34  //Internal Couplings - ics
35  string("using ics_")+DeviceType+Id+string(" =std::tuple<"));
36  string("IC<")+msgClassifierNewReadCom+string(",msgClassifierNewRead_defs<Communication>::
37        newOut,")+outbox+string(",outbox_defs<SetDeviceState>::newIn>,"));
38  string("IC<")+msgClassifierNewReadCom+string(", msgClassifierNewRead_defs<Communication>::
39       readout,")+inbox+string(" ,inbox_defs<SetDeviceState>::readIn>"));     string(">;"));
40 }
```

Figure 38 Generating the DEVS computerized model of coupled models e-mail, beeper, and fax

These functions use the XML file we discussed earlier and/or the parameters for the devices defined in the Agent Based model. We build a vector with all the atomic models inside the coupled model instantiated and a second vector with the coupled model we implemented. The connections inside the coupled model are defined using the definition explained in section 8.1 for the coupled model's instantiation. The rules are written in a way such that the output of the function (shown in Figure 39) will include all the code needed (based on our discussion in section 6.3.7 and Figure 5, where we showed how coupled models are defined in CDBoost). Figure 38, shows the implementation of the function used to instantiate a coupled model representing devices that send/receive text (i.e. *email*).

```
1 template<typename TIME>  //Atomic models inside the instantiated coupled model
2 class msgClassifierNewReadCom : public msgClassifierNewRead<Communication, TIME> {
3 public:
4 msgClassifierNewReadCom(): msgClassifierNewRead<Communication, TIME>(TIME("00:00:500")) {};
5  };
6 template<typename TIME>
7 class inboxFAX1 : public inbox<SetDeviceState, TIME> {
8 public:
9 inboxFAX1(): inbox<SetDeviceState, TIME>(DeviceId(DeviceType::FAX, "1"),TIME("00:00:500"),
10      TIME("00:01:000")) {};
11 };
12 template<typename TIME>
13 class outboxFAX1 : public outbox<SetDeviceState, TIME> {
14 public:
15 outboxFAX1(): outbox<SetDeviceState, TIME>(DeviceId(DeviceType::FAX, "1"),TIME("00:00:500"),
16      TIME("00:01:000")) {};
17 };
18 // instantiated coupled model
19 using iports_FAX1 = std::tuple<inp_setOutOfOrder,inp_network,inp_fromKeyboard>;
20 using oports_FAX1 = std::tuple<outp_toScreen,outp_network>;
21 using submodels_FAX1=models_tuple<inboxFAX1,outboxFAX1,msgClassifierNewReadCom>;
22 using eics_FAX1 =std::tuple<
23   EIC<inp_setOutOfOrder,inboxFAX1, inbox_defs<SetDeviceState>::setStateIn>,
24   EIC<inp_setOutOfOrder,outboxFAX1, outbox_defs<SetDeviceState>::setStateIn>,
25   EIC<inp_network, inboxFAX1, inbox_defs<SetDeviceState>::newIn>,
26   EIC<inp_fromKeyboard,msgClassifierNewReadCom,msgClassifierNewRead_defs<Communication>::in> >;
27
28 using eocs_FAX1 =std::tuple<
29   EOC<inboxFAX1, inbox_defs<SetDeviceState>::displayOut, outp_toScreen>,
30   EOC<outboxFAX1, outbox_defs<SetDeviceState>::displayOut, outp_toScreen>,
31   EOC<outboxFAX1, outbox_defs<SetDeviceState>::networkOut, outp_network>       >;
32
33 using ics_FAX1 =std::tuple<
34  IC<msgClassifierNewReadCom, msgClassifierNewRead_defs<Communication>::newOut, outboxFAX1,
35      outbox_defs<SetDeviceState>::newIn>,
36  IC<msgClassifierNewReadCom, msgClassifierNewRead_defs<Communication>::readOut, inboxFAX1 ,
37      inbox_defs<SetDeviceState>::readIn>              >;
```

Figure 39. Output of the function explained in Figure 38

Figure 39 shows the output of this function: the atomics inside the coupled are instantiated and the coupled model is defined following CDBoost definitions, so it can be simulated. We have chosen a simple example to explain the logic behind it. The rest of the functions are implemented following a similar logic taking into account more parameters of the XML file.

In Figure 38 (lines 4-9), we instantiate the atomic models used inside the couple as we see in Figure 39 (lines 1-16). We call a function that takes as inputs the atomic model parameters and returns the model instantiated in CDBoost format. The function takes as inputs the type of text message device (i.e. e-mail, fax or beeper), the id of the person that owns the device (i.e. the Id in the agent XML file), and two attributes of the devices: their communication delays and the time it takes to acknowledge that it is out of order.

The rest of the figure defines the coupled model instantiation. Lines 11-17 (Figure 38) returns the coupled model input and output ports and the sub models inside the coupled implemented as a tuple as shown in Figure 39 (lines 19-21). Lines 19-25 (Figure 38) generates the External Input Couplings (EIC) as a tuple of tuples with 3 elements: the name of the coupled model input port, the name of sub model connected to the input port and the input port name of the sub model as shown in Figure 39 (lines 22-26). External Output Couplings (EOC) are defined as the EIC but with a different order: sub model name, output port name of the sub model and output port name of the coupled model (see lines 27-32 in Figure 38 for the function definition and lines 28-31 in Figure 39 for the output). Finally, Internal Couplings (IC) are defined as a tuple of tuples of four elements: name of the outcoming sub component, sub model output port name, the name of the incoming sub model, sub model input port name. In Figure 38 (lines 34-40), we show the code that generates the implementation. The output of the code is shown in Figure 39 (lines 33-37).

```
1 int main(int argc, char ** argv) {
2     int numberOfPersons = stoi(argv[1]);
3     string folder = argv[2];
4     string mainModel = string("../TOPMODEL/MainTop.cpp");
5     string content, tSUBMODELS, tIC, tEIC, tEOC, tIPORTS;
6     string tOPORTS = "outp_taskDeviceFinished, outp_taskActionFinished";
7
8      myModelfile.open(mainModel);
9      TOP = open_coupled(string("TOP"));
10
11    ifstream infile("NEP_Cadmium_Headers"); //Define Headers and I/O ports inside MainTop.cpp
12    for(int i=0; infile.eof()!=true ; i++)        // get content of infile
13        content += infile.get();
14    myModelfile << content << endl;
15
16    for(int i = 1; i <= numberOfPersons; i++){     // DEVICES
17     in = folder+string("P")+to_string(i)+string(".xml");
18     person.load(in);
19    DEVICES = DevicesCoupledModel(person);
20      for(int j = 0; j<DEVICES.first.size(); j++)
21        myModelfile << DEVICES.first[j] << endl;
22      for(int j = 0; j<DEVICES.second.size(); j++)
23        myModelfile << DEVICES.second[j] << endl;
24    }
25 ...
```

Figure 40. Code snippet of the program that generates the top model

To build the top-level model, we implement a program that takes all the XML files where the agents are defined, it parses each file and transforms them into a structure to generate the parameters

of all the functions explained earlier in this section. The output is a file with thousands of lines of code that CDBoost understands. This file includes all the atomic and coupled models' instantiated, which, once compiled, generates the NEP Diffusion Simulation ready to generate results. In Figure 40, we show a code snippet showing a part of this program, which output (i.e. a code snippet of the top-model) can be seen in Figure 41.

We use two parameters: the number of agents (i.e. the number of XML files to be loaded) and the path to the folder where their XML descriptions are stored. The number of agents is used to define the number of instances of *Devices* and *Person* models inside the coupled model, as shown in lines 2 and 16 (Figure 40). In lines 5-6, we define all the variables needed to define the top model.

```
1 struct inp_generator : public cadmium::in_port<Command>{};  // SET INPUT PORTS FOR COUPLED
2 struct inp_network : public cadmium::in_port<Communication>{};
3 ...
4 outp_myLocation : public out_port<PeopleLocation>{};   // SET OUTPUT PORTS FOR COUPLED
5 outp_network : public out_port<Communication>{};
6 ...
7 template<typename TIME>    // Define atomic and coupled unit devices
8 class filterDevicesNetwork1: public filterDevicesNetwork<TIME> {
9 public: filterDevicesNetwork1(): filterDevicesNetwork<TIME>("1") {};             };
10
11 template<typename TIME>
12 class filterDevicesSetOutOrder1: public filterDevicesSetOutOrder<TIME> {
13 public: filterDevicesSetOutOrder1(): filterDevicesSetOutOrder<TIME>("1") {};            };
14
15 template<typename TIME>
16 class phoneMOBILEPHONE1 : public phone<SetDeviceState, TIME> {
17 public: phoneMOBILEPHONE1(): phone<SetDeviceState,TIME> (DeviceId
18        (DeviceType::MOBILEPHONE, "1"),TIME("00:00:500"),TIME("00:01:000")) {};          };
19
20 template<typename TIME>
21 class phoneLANDLINEPHONE1 : public phone<SetDeviceState, TIME> {
22 public: phoneLANDLINEPHONE1(): phone<SetDeviceState, TIME>(DeviceId(DeviceType::LANDLINEPHONE,
23       "1"),TIME("00:00:500"),TIME("00:01:000")) {};             };
24//DEFINE COUPLED DEVICE
25 using iports_DEVICES1 = tuple<inp_setOutOfOrder,inp_in_com,inp_network>;
26 using oports_DEVICES1 = tuple<outp_out_com, outp_network>;
27 using submodels_DEVICES1 = models_tuple<filterDevicesSetOutOrder1, filterDevicesNetwork1,
28     filterDevicesMicroKeyboard, sinkDevices_atomic,phoneMOBILEPHONE1, phoneLANDLINEPHONE1,>
29 using eics_DEVICES1 = tuple<
30   EIC<inp_setOutOfOrder,filterDevicesSetOutOrder1,filterDevicesSetOutOrder_defs::in>,
31   EIC<inp_in_com,filterDevicesMicroKeyboard, filterDevicesMicroKeyboard_defs::in>,
32   EIC<inp_network,filterDevicesNetwork1, filterDevicesNetwork_defs::in>        >;
33 ...
```

Figure 41 Code snippet of the output in the program defined in Figure 40

Then, we start defining our coupled model. First, we parse a file where the headers of CDBoost and of the parameterized Atomic models are defined (lines 11-14). The top model ports are also defined in that file. The output of this part of the program is shown in Figure 41, lines 1-6. Then, we call the functions explained earlier in this section to generate the component of the DEVS top model. In lines 16-24 (Figure 40), we show the definition of all the *Devices* coupled models. For each agent, we define a *Devices* model by loading the proper XML and calling the function that generates the coupled (line 19). We then generate all the atomic instantiated and the coupled in the MainTop.cpp file (lines 20-23). In Figure 41 (lines 7 - 33), we show a code snippet of the output of this last part of the program.

This proposed implementation has the following advantages:

- If the behavior of any agent or any relation (defined in our XML file) changes, we do not need to modify our model or change the implementation. We just update the XML file, run the program again with different parameters and compile the top model. This facilitates running different scenarios where the connections between agents and/or the agent behavior vary. It also allows us to run a subset of the NEP such as specific group just reducing the XML files that the program takes as input to this subset of people.
- It reduces implementation time as the TOP model is automatically built based on the rules we defined in the program and the functions explained in this section.
- This implementation also facilitates verification and debugging since all instances of the atomic and coupled models are created by a small set of functions and atomic models. Finding a bug in the parameterized Atomic model or function will fix the bug for all the models.
- It facilitates model reusability. For example, if we want to update a model of the architecture (e.g., we want to replace the Networks coupled model for a more accurate one), we just need to define the new atomics inside the coupled model, and update the function that defines the Networks coupled model. This implementation saves time and effort in model definition, implementation, and verification.

Chapter 9. Case Study: A Collapse in the Communications

In this chapter, we present the results of the study of a collapse in different communication mechanisms using the Network model.

As we explain in our architecture, to construct the network representing the NEP, we used the information provided in the *Requirements Document*, which identifies all the people involved in the emergency and all their communication mechanisms. The nodes represent the agents or the people involved in the nuclear emergency. The links represent the communication relations between the agents or the people involved. The resulting network is a multiplex network composed of 832 nodes and 10 layers. The layers represent specific communication mechanisms: fax, Internet, landline and mobile phone, Reman radio, Remer radio, satellite, Autonomous Police communication network, Radiological Group communication network, Civil Guard communication network, and communications in-situ.

9.1. Assumptions

The following assumptions are made during the analysis:

1. Although we distinguish different communication layers, we assume that the communication links in the different layers are equivalent (i.e. all of them establish the same type of connection between agents). This is justified because all the links allow communications between agents. Consequently, we analyze the multiplex network using simplex networks. To include redundancy in the communications, we assigned different weights to the links. The weight represents the number of communication mechanisms available to establish the communication. For example, if the communication is supported by one communication channel the weight is one, if it is supported by two, the weight is two and so on.
2. We group Reman and Remer radio channels under the radio communication channel. This assumption is done because they are different frequencies in the radio network. Moreover, if the individuals only have a radio device, they can only tune a single channel.
3. We assume that in-situ communications do not fail. This means that communication between two people in the same is robust.
4. We assume that the Civil Guard communication network, the Radiological Group communication network and the Autonomous Police communication network are robust and they do not fail because these communication mechanisms can use both the phone and radio network indistinctly. A failure in these communications mechanisms implies a simultaneous interruption of both radio and phone networks. We can make this assumption is justified because we are interested in studying the effects of single failures in the network (i.e. we are not interested in the consequences of two different networks failing at the same time).
5. We assume that the mobile and landline phones use the same communication channel. Most communications in the NEP involve mobile to mobile or mobile to landline

communications because first responders cannot use landline phone. Therefore, if the mobile network fails, the communication cannot be established. Although there maybe landline to landline phone communications, making this assumption simplifies the model being on the conservative side since we are using the worst case scenario.

6. The NEP communication network is dynamic. The potential communications between people change as their locations do. We assume a static network and we analyze it using different scenarios to lighten this assumption.

We use two scenarios for the analysis of a downfall in the different communication channels. In Scenario 1, we represent the NEP communication network after declaring the nuclear emergency and the members of Executive Body are together, but no further action has been taken. That means that the field teams have not been positioned. In Scenario 2, we represent a situation in which all teams have been situated.

Within each scenario, two cases are analyzed. Case 1 includes the whole network (i.e. all the people involved in the NEP), composed by 832 agents containing the replacement teams. Case 2 represents the network excluding the replacement teams (i.e. only the individuals working at a certain moment are included). This reduces the size of the network to 522 agents.

To analyze how a collapse in the different communication channels affects the communication structure in an organization, we evaluate if these downfalls isolate some groups or individuals, and therefore they cannot receive communications. The aim is to conclude if the network is robust against these types of failures or not. For our analysis, we use a network metric: the number of connected components in the network.

A network component consists of all the connected nodes of the network. A network can have a single component or multiple ones. The number of connected components allows us to identify whether or not these downfalls isolate individuals/groups. If the network has one component, all individuals are connected in a single group. Otherwise there are isolated groups of one or more individuals. Since we study an emergency situation and we should be able to transmit information to every person in the network, we assume that the network is not robust if there is more than one component.

In the rest of the section, we present the effects of a downfall in the communication infrastructure, and its effects the NEP. In particular, we analyze following in phones, fax, Internet, radio, satellite and in the phone, fax, and Internet channels at the same time

The figures and the analysis are made with Gephi. The figures depict one particular view of the network. We choose these views because they provide a good visualization of the number of components in the network.

9.2. Downfall in phone communication channel

To simulate the effects of a downfall in the phone communications while managing the emergency, we remove from the network model of the NEP the links that only handle landline and mobile phone communications and we calculate the number of connected components using Gephi. As

we will see below, the number of connected components is greater than one in both analysis scenarios and both cases (i.e. this downfall isolates some people while managing the emergency).

9.2.1. Scenario 1

In Figure 42, we show the NEP network in Case 1 if the phone communication channel does not work. In this case, the number of connected components in the network is 207.

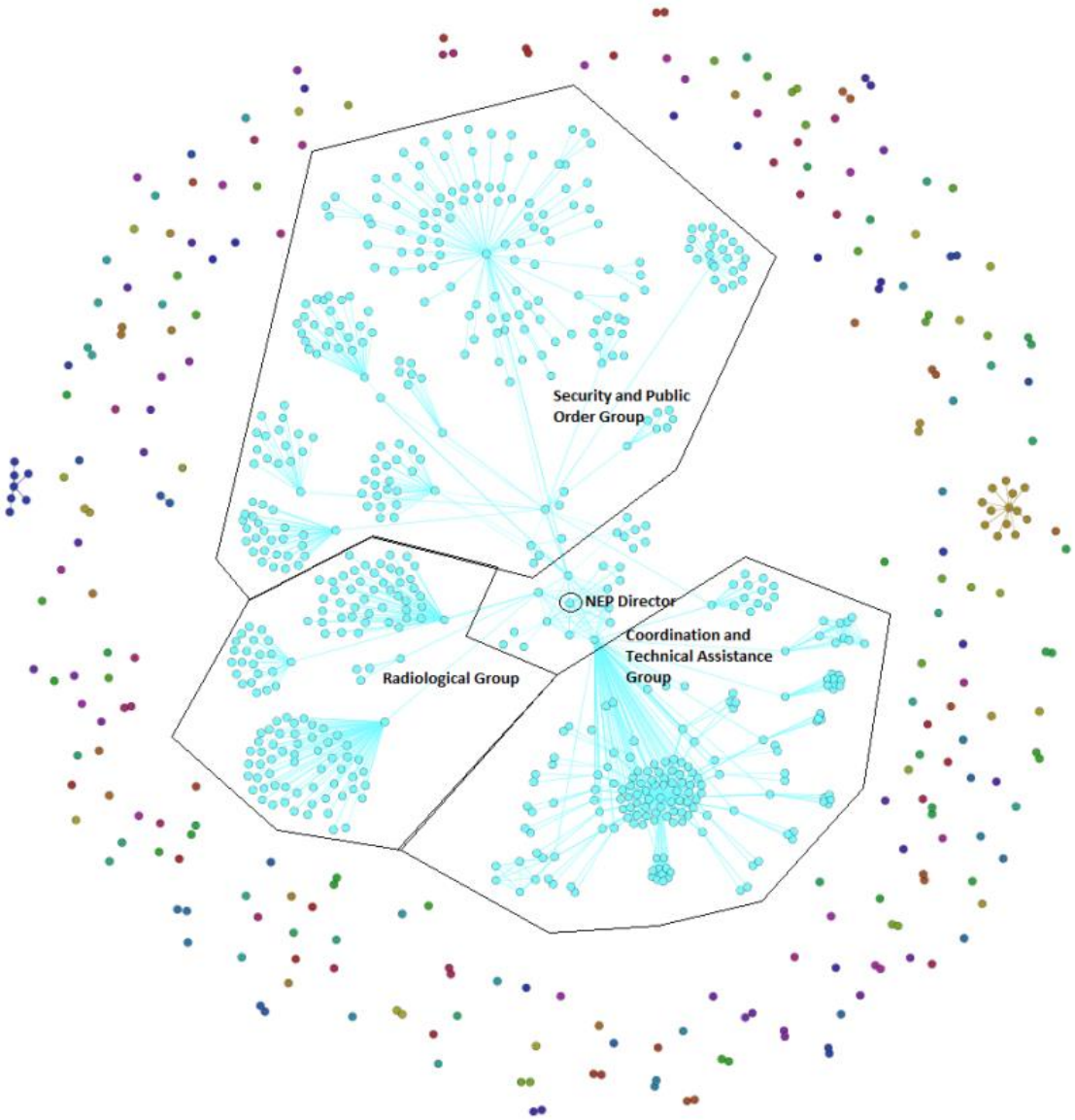


Figure 42. Collapse in the phone communication channel. Scenario 1. Case 1.

In the middle of the figure is the largest component, which includes 68.63% of the nodes (571 nodes). The NEP Director and the heads of the Executive Body belong to this component. Members of the Radiological Group, Public Security and Order group, and Technical Assistance and Coordination group also belong to it. The other 261 nodes, depicted in the figure surrounding the largest component, represent isolated individuals or teams.

From this analysis, we see that the network is not robust against a collapse in the phone communication channel since the number of connected components is greater than one. To better identify the isolated individuals we made the same analysis in Case 2 (i.e. not including the replacement teams). In section 9.3, we discuss how the robustness of the network can be improved.

In Figure 43, we show the NEP network in Case 2 if the phone communication channel does not work. In this case, the network has 87 connected components. In the middle of the figure is the largest component, which includes 74.71% of the nodes (390 nodes). The NEP Director and the heads of the Executive Body belong to this component. Members of the Radiological group, Public Security and Order group and Technical Assistance and Coordination group also belong to it. The other 132 nodes represent the isolated teams or individuals. They are depicted in the figure around the major component. From this analysis, we conclude that the phone communication channel is not robust in Case 2 in Scenario 1.

Figure 43. Collapse in the phone communication channel. Scenario 1. Case 2.

The difference in the number of nodes that composes the largest component between both cases indicates that some replacement teams remained connected to the major component despite the downfall. All the replacement teams connected to the largest component belong to the Public Security and Order group (specifically to the Civil Guard) and the Radiological group. This affirmation is done comparing the nodes that belong to the largest component in Case 1 and 2.

The isolated individuals or teams in Case 2 are:

- The Autonomous Police
- The Local Police
- The directors and teachers in the schools in Zone I and II
- The Government Education Supervisor
- The different ministries involved in the emergency
- The NSC Inspector at the NPP
- The whole Health group

Since the individuals in Case 2 are a subset of individuals in Case 1, we can conclude that the isolated individuals in Case 1 are the same as in Case 2 but including all the replacement teams, except for the one from the Radiological Group and the Civil Guard.

9.2.2. Scenario 2

When the phone communication channel does not work, in Case 1, the NEP network has 139 components. The largest component includes the 83.41% of the nodes (694 nodes). The other 138 nodes are the isolated individuals or teams. If we analyze the number of components, it is significantly reduced to 19 components due to in-situ communications.

The resultant network in Case 2 (i.e. without replacement teams) is shown in Figure 44. In this Scenario, due to the in-situ communications, the communities are mixed and it is not possible to distinguish the different groups in the figure. In the middle of the figure is the largest component, which includes 96.55% of the nodes (504 nodes). The other 18 nodes, depicted in the figure around the major component represent the isolated individuals. We see that all the people working in the emergency are connected except the ones labeled around the largest component. From this analysis, we conclude that we conclude that the network is not robust against a collapse in the phone communication channel in Scenario 2 since the number of connected components is greater than one.

As in Scenario 1, the number of nodes in the largest component differs from Case 1 and 2. This difference indicates that some replacement teams remained connected to the largest component despite the downfall. All these teams belong to the Public Security and Order group (specifically to the Civil Guard) and the Radiological Group. This affirmation is done comparing the nodes that belong to the largest component in Case 1 and 2.

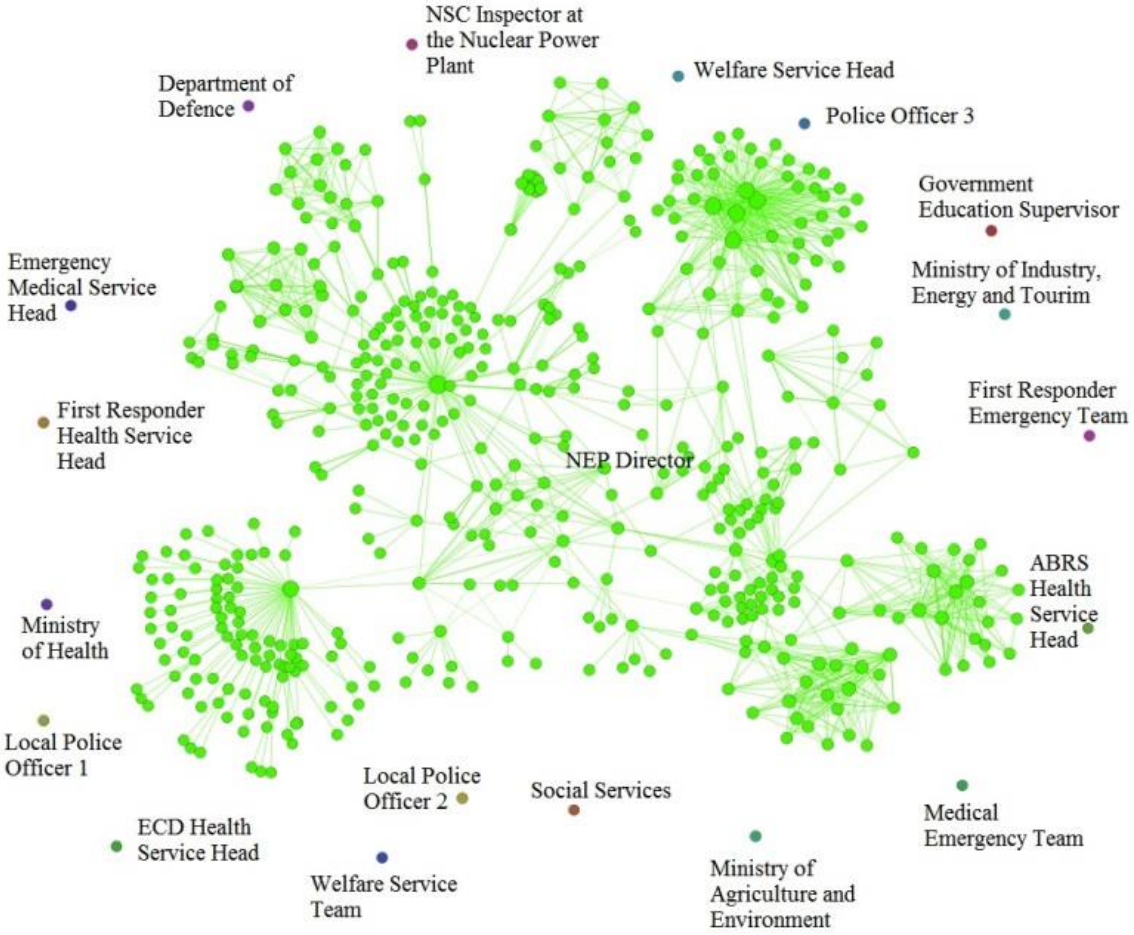


Figure 44. Collapse in the phone communication channel. Scenario 2. Case 2. Not possible to distinguish the groups due to in-situ communications

Comparing Case 1 and 2, we can see that when the network is positioned, the amount of isolated individuals decrease considerably, but the isolation problem does not disappear. Apart from the replacement teams, other 18 agents or individuals are isolated as seen in figure 6. These agents are:

- The 5 Ministries
- The NSC Inspector at the NPP
- The 5 heads in the health group
- The first response and emergency medical health teams in the province capital of the NPP location
- The welfare team
- The Government Education Supervisor
- The Local Police heads

9.2.3. Discussion

After the analysis of a downfall in the phone communication channel in different scenarios and cases, we conclude that the phone communication is not robust. If this communication mechanism fails, the transmission of the information to solve the emergency will be cut down for some of the

people involved. This will probably carry out a loss of efficiency in the overall performance of the NEP.

To solve the isolation problem in the network is not necessary to duplication all links; we just need to take into account the isolated individuals or groups and connect them to the network. The minimum number of communication links to be added will be the number of connected components minus one. This solution is only possible after the analysis if the downfall in the communication channel and the identification of the isolated individuals or communities. In Figure 45, we show an example of how we can connect a network with three components adding just two communication links. We also show how we can create redundancies adding just three more links

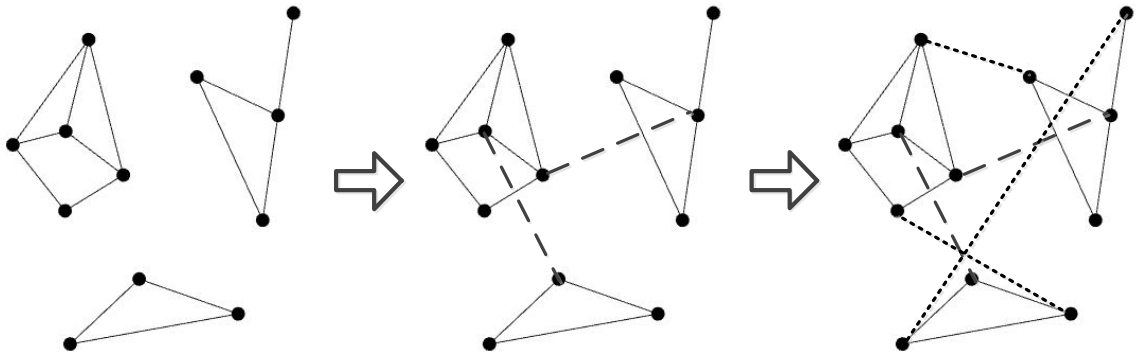


Figure 45. Connection of the three components of a network with two links. Building redundancies adding three more links.

In some cases, it may not be enough to add redundancy. An example may be if the redundancy we add fails in the same cases as the original link. Other cases may be if the nodes involved in the redundancy are not reliable to transmit some information or they are not efficient in the information transmission process. With our network model we can test different scenarios where we add different links to create redundancies and analyze what happens is a downfall in the communication mechanism occurs. However, we cannot include the behavior of the individuals.

This limitation is solved using the DAM, and the simulation results of this study will be presented later. With the DAM, we can do this type of analysis including the behavior of the agents. We can also simulate different probabilities of failure for the links. These studies allow us to probe different solutions and decide which one is better.

9.3. Downfall in the fax communication channel

To simulate the effects of a downfall in the fax communications while managing the emergency, we remove from the network model of the NEP the links that only handle fax communications and we calculate the number of connected components using Gephi.

In scenario 1 case 1, we obtain the same network topology as the one presented in Figure 29 (the wait of the links varies). As we can see in Figure 29, the number of connected components is one (i.e. this downfall does not isolate some people while managing the emergency). We just analyzed scenario

1 case 2 for validation purposes, because we already know that the number of connected components is going to be one (the network in case 2 is a subset of the network in case 1).

We also analyzed scenario 2 for validation purposes. Based on the results of scenario 1 and taking into account that the network in scenario 2 is built over the network in scenario 1 adding the links that represents in-situ communications, we expect to have one connected component. In Figure 46, we present the network after simulating a downfall in the fax communications in scenario 2 (case 1). Looking at the figure, we corroborate that the number of connected components is one.

Based on this analysis, we conclude that the fax communications are redundant and therefore the network is robust against this type of failure.

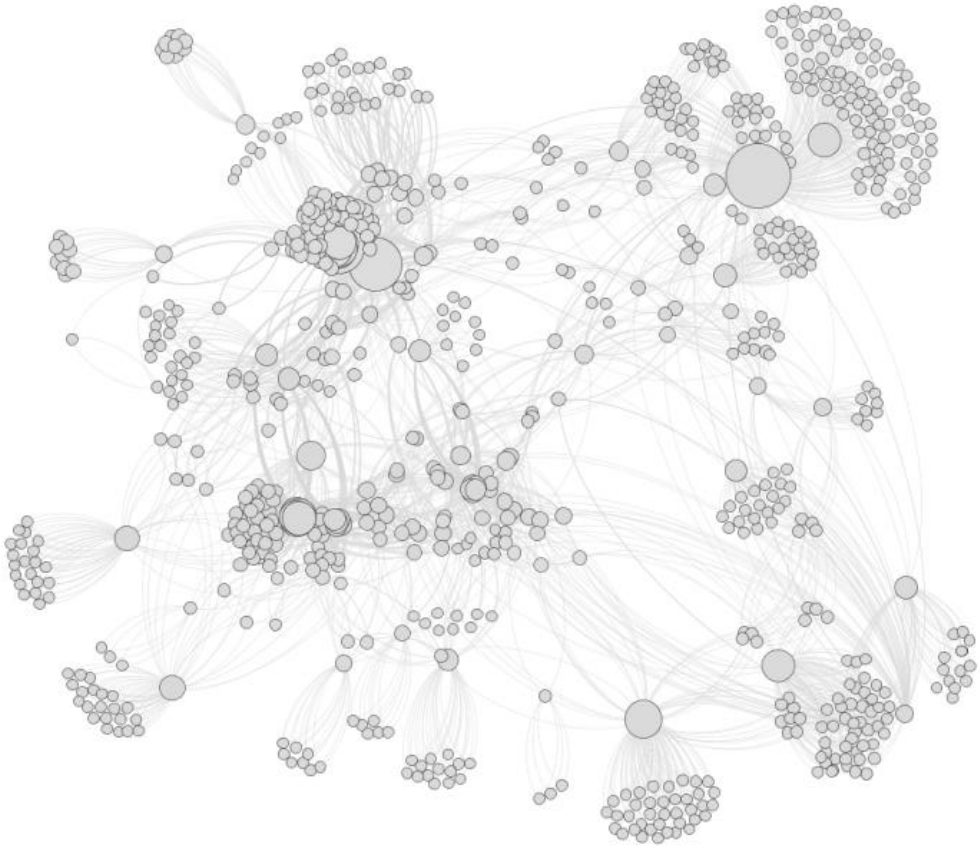


Figure 46. NEP network after a collapse in the fax communication channel. Scenario 2. Case 1. Not possible to distinguish groups due to in situ communications.

However, the use of this communication channel has some limitations.

For example, it cannot be used to send information or commands to field teams. The communication with these teams is important because they are who implement the actions to solve the emergency. Moreover, to be sure that the person has received your information or command we need to wait for confirmation. Therefore, fax redundancies are not enough to build a robust network.

9.4. Downfall in the Internet communication channel

In the NEP, the Internet is used to send information via e-mail. To simulate the effects of a downfall in the Internet communications while managing the emergency, we remove from the network model of the NEP the links that only handle e-mail communications and we calculate the number of connected components using Gephi.

The results we obtain are the same as in the downfall of the fax communications (see section 9.3). Only the weight of the links in the network varies. Since the number of connected components is one in both analysis scenarios and both cases (i.e. this downfall does not isolate some people while managing the emergency), we conclude that this type of communication is redundant and therefore the network is robust against this failure. However, despite e-mails can be read on mobile phones, the use of this communication channel has the same limitations as the fax. The NEP does not define e-mail communications to field teams and a confirmation of reception is needed. A way to improve the robustness of the communication system in the emergency plan would be allowing e-mail communications to field teams.

9.5. Downfall in radio communication channel

To simulate the effects of a downfall in the radio communications while managing the emergency, we remove from the network model of the NEP the links that only handle communications through the Reman and Remer radio communication channels, and we calculate the number of connected components using Gephi.

The results we obtain are the same as in the downfall of the fax communications (see section 9.3). Only the weight of the links in the network varies. Since the number of connected components is one in both analysis scenarios and both cases (i.e. this downfall does not isolate some people while managing the emergency), we conclude, as in the previous case, that this type of communication is redundant and therefore the network is robust against this failure.

However, this communication mechanism has some limitations related to both Reman and Remer channels. The radio coverage in Zone I and II is limited due to the landform. Moreover, the Raman radio channel does not cover all municipalities. It is only deployed within the municipalities that are closer to the NPP. Related to Remer channel, it is important to remember that the population can also hear the communications along this channel. This issue makes the Remer radio channel not suitable for confidential information. Consequently, this channel cannot be used to transmit some information. Therefore, radio communication redundancies are not enough to build a robust network.

9.6. Downfall in satellite communication channel

To simulate the effects of a downfall in the satellite communications while managing the emergency, we remove from the network model of the NEP the links that only handle this type of communications and we calculate the number of connected components using Gephi.

The results we obtain are the same as in the downfall of the fax communications (see section 9.3). Only the wait of the links in the network varies. As in the previous cases, we conclude that this type of

communication is redundant and therefore the network is robust against this failure since the network connectivity does not critically depend on the satellite communications.

However, this communication mechanism has some limitations. The main drawback of this redundancy is that it is not an extended communication mechanism along the NEP. It only supports the communications between five nodes: *NEP Director*, *NSC President*, *PENCRA Director*, *Government Presidency and the town hall of a municipality in Zone I*. Moreover, four out of the five individuals connected by this communication channel are high government ranks that do not take direct action to solve the emergency. They only have to be informed about the situation. This network does not support communications between field teams (who are people that take direct actions to recover from the emergency) and their supervisors. As this network is not yet use in the above-mentioned type of communications, it can be used to define redundancies between field teams and their supervisors.

9.7. Joint Downfall in phone, fax and Internet channels

Although we already know that the network is not robust against a downfall in the phone communications, here we show a study of a joint failure in phone the phone, fax, and Internet communication. The landline phone and fax are handled by the same infrastructure. In the NEP, Internet is used to send e-mails. These e-mails are sent to institutional e-mail accounts. These accounts are read from the computers in people's job places. It is likely that if the landline phone fails, then the Internet will also stop working. This affirmation is based on the type of infrastructure that supports Internet communications in the NPP area. Following the Internet map coverage and infrastructure deployed in Spain (Spanish Goverment 2015), the most probable Internet connection in in the area include the ADSL line (Asymmetric Digital Subscriber Line). ADSL is a transmission technique applied over the traditional phone landline networks. We want to analyze if this joint failure isolates more individuals than the previous study.

To simulate the effects of this downfall while managing the emergency, we remove from the network model of the NEP the links that only handle these types of communications and we calculate the number of connected components using Gephi.

The results of this analysis are similar to the phone communication downfall. It causes the isolation of the same individuals. This means that the communications that can be handled by phone, e-mail, and fax at the same time, they are also handled by other communication mechanisms. For example, in the Public Security and Order group, they are also handled by the special mix radio-phone communication network. In the Technical Assistance and Coordination group are also handled by the Reman and Remer communication channels. However, the redundancies implemented in the communication network are not enough to create a robust communication network against this failure.

9.8. Discussion

With this study, we showed that the NEP communication network in our case study is robust against a collapse in the satellite, Internet, fax, and radio communications. However, it is not robust against a downfall in the phone communication channel. The redundancies implemented in the network are not enough to provide robustness against a collapse in the phone communication channel. This downfall would result in important consequences as stated below.

We could see that only the replacement teams in the Public Security and Order group (specifically the Civil Guard) and in the Radiological Group remain connected to the largest component (i.e. to the component that most of the people that work in the emergency are connected to, including the NEP Director) when a downfall in the phone communication channel occurs. This means that it will only be possible to establish communication with the replacement teams within these two groups. The other replacements teams, as they are isolated, will not be able to receive the orders and information to take action.

As we saw in the previous section, when a downfall in the phone communication channel occurs, the NSC Inspector at the NPP is isolated. This means the lack of in-situ information about the incident evolution at the NPP and the actions taken inside the plant to solve the situation. These cause that the actions taken outside to control the effects of the emergency will be defined without up-to-date data. It is important to remark that the phone, which connects the NEP director with the NSC at the NPP, is a special cable line phone between these two people. The communication is not held by the regular phone network. A lesson learned is the need to establish communication redundancies between the organization and the outside management centers.

The isolation of the individuals increases the complexity of communications. Therefore, it also probably increases the time required to transmit the information to a specific person. For example, the isolation of the Local Police implies that the only way to receive information and orders is through the Civil Guard. This means that the Civil Guard must reach the Local Police location. In Scenario 1, this problem is more important than in Scenario 2. In Scenario 1, the network is not deployed and the Civil Guard teams and the isolated individuals are probably not located in the same area. However, in Scenario 2, the teams are deployed and the chances to be in nearby positions are greater. This means that the time to reach the location of the isolated individuals is higher in Scenario 1 than in Scenario 2.

The isolation of the individuals in the Health group in scenario 1 implies that these people will not receive instructions and information about the evolution of the emergency. This issue may cause different problems such as that the teams do not know what they have to do. In this uncertainty situation, we do not know the decisions each team will take, and if they will be accurate or erroneous. This problem is partially mitigated when we move from Scenario 1 to Scenario 2. In both scenarios, the health group teams cannot communicate with their supervisors directly. In Scenario 2, this problem is partially solved due to in-situ communication. The teams can receive instructions and information from the Public Security and Order group teams. However, the time and the complexity of the communications increase: the information transmission process will take at least five steps instead of three. When all the communication mechanisms work properly, the NEP Director transmits the information to the head of the Health group. The head of the Health group transmits the information to the subheads, who send the information to first responders (3 steps). When the communication fails, we need more steps: the NEP Director transmits the information to the head of the Public Security and Order group; then the information is transmitted inside the group following the hierarchy until it arrives to first responders in Public Security and Order group. This process takes at least 4 steps depending on the subgroup. Then, the first responders in the Public Security and Order group transmit the information to first responders in Health group.

Another issue is related to receiving feedback. When a downfall in the phone communication occurs, the feedback provided by the Local Police and the Heath Group teams will not often occur since it can only be transmitted through the Civil Guard. This group also needs to carry out their own

tasks. In Scenario 1, the Civil Guard teams must reach the location of the other teams to make these communications. In Scenario 2, this problem is less critical as the teams are positioned and there would be a Civil Guard team in a nearby location.

The isolation of schools directors and teachers in Scenario 1, and the Government Education Supervisor in both scenarios is relevant if the schools must be evacuated. This issue is solved in Scenario 2, since the Civil Guard and/or Local Emergency Teams from the Technical Assistance and Coordination group will communicate the order. However, in Scenario 1, the evacuation will be delayed as the Civil Guard and the Local Teams are not positioned yet. In both Scenarios, the Government Education Supervisor will lack of information about emergency and the evacuated schools.

If the five ministries are isolated, the NEP Director will not be able to ask for their support of for additional information. Moreover, they are not going to be directly informed about the evolution of the emergency.

In scenario 1, the isolation of the Autonomous Police carries some consequence in the emergency management. The Autonomous Police will not receive the commands of the NEP Director until the satellite communication is deployed, the Civil Guard is positioned or communications in-situ appears. We have to remark that the Autonomous Police and the Civil Guard Group, in case their communications systems fail or they found it is the needed, they can activate satellite communications. However, since they request the activation until it is operative, it takes an elapsed time.

In this chapter we have made a static analysis of the communication network. This analysis has some limitations. For example, we cannot study dynamic scenarios and the behavior of the people (i.e. the behavior of the nodes) is not included. In Chapter 10, we address these limitations using the NEP DAM. We will recall some of the conclusions and assumptions made in this chapter in order to validate them.

Chapter 10. Case Study. Results Analysis from the NEP Diffusion Abstract Model

In this chapter, we present the simulation results of the NEP DAM. After implementing the computerized model, we conducted different simulations. We will show how we used the *NEP computerized* model to explain how can we get and analyze the results of our model. To understand how the *NEP Diffusion Abstract model* results are generated (see Figure 32), we will first explain some of the components generated using the architecture and the methods explained in previous section: *Command Generator, Device State, Network State* and *Behavior Rules* (the main component of the *Person* coupled model presented earlier in Figure 33). Then, we will present different versions of the *NEP Diffusion Abstract* model (presented in Figure 32) to analyze different groups involved in solving the emergency. We will use the head of the NEP to explain how we get the results. Then, we focus on the analysis of the Radiological and Health Groups. We will also discuss how some of these results match the ones presented in Chapter 10.

We will validate our results through expert validation. In this section, we also present some analysis that validate our results against the NEP specifications defined in the *Requirements Document* (Chapter 7)

All the simulation results presented in this section have been obtained running the model in a machine with the characteristics presented in Table 9.

The time taken by the simulation depends on the size of the model we are analyzing and the scenarios. Taken a sample that includes the different sizes and scenarios (i.e. using the simulation of the Health and Radiological groups), the average simulation time is 953 s. The 95% confidence interval for the mean is [953±124] seconds.

Table 9. Characteristics of the computer

Attribute	Value
Processor	Intel® Xeon(R) 3.20GHz × 4
RAM	48GB
OS	Ubuntu 16.04
OS type	64bit

10.1. Testing individual components

10.1.1. Command Generator, Devices State & Networks State

These three components of the *NEP Diffusion Abstract model* presented in Figure 32 generate the inputs that trigger the simulation and update the state of the *Networks* and *Devices* models (i.e. if they are faulty or not).

These models are automatically generated as three different instances of the same generator atomic model, which reads a file and generates the messages found there as output at the specified time. This atomic model generates Messages on its output port based on the data (time and message) specified in a structured file that determines the different simulation scenarios, as we explain in the next example. The main difference between the three generators is the type of message they generate (*Command, SetDeviceState or SetNetworkState*), which is a parameter of the model. This parameter is defined when the model is instantiated.

We show the execution of *Devices State*, which *State* generates *SetDeviceState* messages that have a device id with two fields (type of device & id) and a *broken* variable that takes the value 1 if the device is faulty and 0 otherwise. In Figure 47, we see an input file that shows which devices will be faulty, and which ones will recover. In this example, at time 00:00:00:000, the mobile from the person with id 8 is broken. At time 00:02:15:000, the landline phone from person 8 also fails. The rest of the input file includes similar information for different users and devices.

```
00:00:00:000 MOBILEPHONE 8 1
00:02:15:000 LANDLINEPHONE 8 1
00:05:00:000 RADIOLOGICAL_GROUP_DEVICE 5 1
00:10:00:000 BEEPER 8 1
00:10:00:000 RADIO_REMAN 4 1
00:10:05:000 RADIO_REMER 6 1
00:10:05:000 PRIVATELINEPHONE 8 1
00:10:06:000 FAX 8 1
00:10:06:000 TRANKI_E 8 1
...
```

Figure 47 Input file for the model *Devices State*

This file is read by the *Device State* model, and it generates the simulation output shown in Figure 48. As we can see, *Device State* model translate the inputs in Figure 47 into messages generated through the model's output port. These messages can be sent to other models. In this case, if we use the coupled model structure defined in Figure 32, the simulation outputs seen in Figure 48 would be transmitted to the *Devices* model, as we show in Figure 32. Using this input, the *Devices* models would update their state (i.e. faulty or recovered) at runtime.

```
00:00:00:000 [Devices State::out: {MOBILEPHONE 8 1}] generated by model Devices State
00:02:15:000 [Devices State::out: {LANDLINEPHONE 8 1}] generated by model Devices State
00:05:00:000 [Devices State::out:{RADIOLOGICAL_GROUP_DEVICE 5 1}] generated by Model Devices State
00:10:00:000 [Devices State::out: {BEEPER 8 1, RADIO_REMAN 4 1}] generated by model Devices State
00:10:05:000 [Devices State::out: {RADIO_REMER 6 1, PRIVATELINEPHONE 8 1}]
        generated by model Devices State
00:10:06:000 [Devices State::out: {FAX 8 1, TRANKI_E 8 1}] generated by model Devices State
...
```

Figure 48 *Devices State* log file when simulated with the input file in Figure 47

As we have already mentioned, the advantage of using this configuration to generate which *Devices* will change its state to defective (or will recover) is that we can define different scenarios by updating the file shown in Figure 47.

Command Generator, based on its *State*, generates *Command* messages that have three attributes the content of the command, the receiver and the sender. In Figure 49, we see an input for the model with four commands that will be generated at four different times. In this example, at time

00:00:00:000 the NEP Director decides to establish the command “Establish Emergency Level 0”. The rest of the input file is interpreted in a similar way.

```
00:00:00:000 “Establish Emergency Level 0” – “NEP Director”
00:25:00:000 “Establish Emergency Level 1” - “NEP Director”
00:50:00:000 “Establish Emergency Level 2” - “NEP Director”
02:00:00:000 “Establish Emergency Level 3” - “NEP Director”
```

Figure 49. Input file for the model *Command Generator*

This file is read by the *Command Generator* model, and it generates the simulation output shown in Figure 50. As we can see, *Command Generator* model translate the inputs in Figure 49 into messages generated through the model’s output port as in it occurs with the *Device State* model. These messages can be sent to other models. In this case, if we use the coupled model structure defined in Figure 32, the simulation outputs seen in Figure 50 would be transmitted to the *Person* model, as we show in Figure 32. Using this input, we can simulate different scenarios where different commands are transmitted.

```
00:00:00:000 [Command Generator <Command>::out: {“Establish Emergency Level 0” – “NEP Director”}]
         generated by model Command Generator
00:03:40:000 [Command Generator <Command>::out: {“Establish Emergency Level 1” – “NEP Director”}]
         generated by model Command Generator
00:05:00:000 [Command Generator <Command>::out: {“Establish Emergency Level 2” – “NEP Director”}]
         generated by model Command Generator
00:10:00:000 [Command Generator <Command>::out: {“Establish Emergency Level 3” – “NEP Director”}]
         generated by model Command Generator
```

Figure 50. *Command Generator* log file when simulated with the input file in Figure 49.

Networks State has a similar behavior than *Devices State* and *Command Generator*. The difference is that it generates *SetNetworkState* messages that two attributes the id of the target network and a *broken* variable that takes the value 1 if the device is faulty and 0 otherwise. Figure 51 shows an input for the model with seven states for the networks at different times. At time 00:00:00:000, the phone network is set to broken state. The rest of the lines are interpreted in the same way.

```
00:00:00:000 PHONE_NET 1
00:01:00:000 FAX_NET 0
00:02:15:000 INTERNET 1
00:02:30:000 PRIVATELINEPHONE_NET 0
00:05:00:000 PHONE_NET 0
00:10:00:000 BEEPER_NET 0
00:10:05:000 RADIO_REMAN_NET 1
```

Figure 51. Input file for the model *Networks State*

This file is read by the *Network State* model, and it generates the simulation output shown in Figure 52. In this case, if we use the coupled model structure defined in Figure 32, the simulation outputs seen in Figure 52 would be transmitted to the *Networks* model, as we show in Figure 32. Using this input, the *Networks* model would update their state (i.e. faulty or recovered) at runtime.

```
00:00:00:000 [Networks State<SetNetworkState>::out:{PHONE_NET 1}] generated by model Networks State
00:01:00:000 [Networks State<SetNetworkState>::out:{FAX_NET 0}] generated by model Networks State
00:02:15:000 [Networks State<SetNetworkState>::out:{INTERNET 1}] generated by model Networks State
00:02:30:000 [Networks State<SetNetworkState>::out:{PRIVATELINEPHONE_NET 0}] generated by model
        Networks State
00:05:00:000 [Networks State<SetNetworkState>::out:{PHONE_NET 0}] generated by model Networks State
00:10:00:000 [Networks State<SetNetworkState>::out:{BEEPER_NET 0}]generated by model Networks State
00:10:05:000 [Networks State <SetNetworkState>::out:{RADIO_REMAN_NET 1}] generated by model
        Networks State
```

Figure 52. *Networks State* log file when simulated with the input file in Figure 51

10.1.2. Behavior Rules

As we explained in section 8.1.1, the main component of the *Person* model is the coupled model *Behavior Rules* (see Figure 33), in which the behavior of the agent is modeled and instantiated based on the agent parameters stored in an XML file as the one presented in Figure 30. As we explained in section 7.3, the behavior of each agent (person) is defined by the experts in Agent Based modeling, and specified and documented using an XML file (one for each agent), based on the *Requirements Document* (Chapter 7). We use these XMLs to automatically instantiate each *Behavior Rules* DEVS model for the different *Person* models in the *Diffusion Abstract model*.

In this example, we will show how to verify the output of agent behavior defined in an XML to later compare it against the results provided by the *Behavior Rules* model. This comparison allows us to check the correctness of the simulation results of *Behavior Rules* model against the one defined in the XML file. For this purpose, we use the XML file presented in Figure 30 instantiated with different values for the attributes (i.e., some of the arguments presented in Figure 30 will change). There are various important attributes we need to introduce in order to understand the example; for instance, *ReactionTime* (10 s), *AnswerPriorityType* (in this case, *DEVICE_PRIORITY*, which means that the response is sorted based on the priority of each device), and *SendPriorityType* (in this case, we use *RECEPTION_ORDER*, i.e., the commands are sent in FIFO order). The person has three devices, a mobile (priority 1), email (priority 2) and a landline phone (priority 3).

As seen in Figure 30, line 14, the agent prioritizes the task in the following order: *ANSWER*, *SEND* and *DO_ACTION*. The tag *AnswerDevicePriority* (line 18) shows the priority assigned when we *ANSWER* to a device. In our instantiation for this particular example, a mobile call will have priority over email and landline calls.

The communication relations in this example will be as follows (Table 10): the person we are defining can send messages to *person 150* using email and mobile; to *person 151* using a landline phone and mobile; to *person 152* using email and mobile; to *person 153* using email; to *person 50* using landline phone, email, and mobile; and to *person 51* using email and mobile. These relations are defined inside the tag *CommunicationRelations* in Figure 30 (line 35).

Table 10 Communication Relations

Target	Devices
Person 150	Email & Mobile
Person 151	Landline phone & Mobile
Person 152	Email & Mobile
Person 153	Email
Person 50	Landline phone & Email & Mobile
Person 51	Email & Mobile

The agent's behavior when sending/receiving messages is defined in *MessageBehavior* (line 45). In this example, when the agent receives the command *"Establish Emergency Level 0"* from *person 1*, it has to send three messages and do an action. The agent has to send *"Establish Emergency Level 0"* to *person 50* (using any device available), *person 51* (using the email) and *person 52* (using any device). The agent also has to do the action *"Transmit Emergency Alert to the population"*. When the agent receives *"Tell people to stay at home"* from *person 39*, it has to send a reception acknowledgment (i.e. *"Tell people to stay at home received*") to *person 55* and *58* using any device.

Once we complete the XML file as discussed above, we automatically instantiate the *Behavior Rules DEVS* model passing the path to the XML file as a model parameter.

Figure 53 shows an input set scenario we used to simulate both the XML file and the *Behavior Rules* model. In the figure, we can see two groups. The first part represents the inputs that come from *Switches* (i.e. *In-Situ Switch* and *Devices Switches* – see Figure 33.b). The second part represents inputs that come from *Receiving and Sending Behavior* models (on the right of Figure 33.b). Repeated times in the same port (e.g. 00:00:01:000) means that we are receiving a bag with multiple messages. In Figure 33, we can see that *Behavior Rules* model has two other ports. Here, we do not use *Generator Filter* and *People In Location Filter* (i.e. nobody is in the current location of the agent, and there are no commands sent to this person from *Command Generator* in Figure 32).

```
Inputs in port connected to Switches models
00:00:01:000 PHONE_MESSAGE CALL_REQUEST MOBILEPHONE 1 MOBILEPHONE 20
00:00:01:000 PHONE_MESSAGE CALL_REQUEST LANDLINEPHONE 50 LANDLINEPHONE 20
00:00:15:000 PHONE_MESSAGE CALL_OVER LANDLINEPHONE 50 LANDLINEPHONE 20
00:10:20:000 PHONE_MESSAGE CALL_REQUEST LANDLINEPHONE 39 LANDLINEPHONE 20

Inputs in port connected to Sending and Receiving Behavior models
00:00:20:000 ANSWER MOBILEPHONE 20 MOBILEPHONE 1 0 “Establish Emergency Level 0” 1 20
00:10:00:000 SEND MOBILEPHONE 20 LANDLINEPHONE 50 0 “Establish Emergency Level 0” 20 50
00:10:30:000 SEND EMAIL 20 EMAIL 51 0 “Establish Emergency Level 0” 20 51
00:50:00:000 ANSWER LANDLINEPHONE 20 LANDLINEPHONE 39 0 “Tell people to stay at home” 39 20
```

Figure 53 Inputs for the model *Behavior Rules*

As we can see, the *Switches* receive two call requests at 00:00:01:000: one from a mobile and another one from a landline phone (one on each line). At 00:00:15:000, we receive an input telling that the call on the landline finished, and at 00:10:20:000, we receive a new call from the landline.

Likewise, in the input port for *Sending* and *Receiving Behavior models*, we receive an input stating that the task *ANSWER* the phone call on the mobile has been finished at 00:00:20:000, with a

command “Establish Emergency Level 0”. At time 00:10:00:000, the task *SEND* “Establish Emergency Level 0” to person 50 has been successfully finished. At time 00:10:30:000, the task *SEND* “Establish Emergency Level 0” to person 51 was successfully finished. Finally, at time 00:50:00:000 we receive that the task ANSWER the phone call on the landline phone has been successfully finished and the person has received the command “Tell people to stay at home”.

We use these inputs to simulate the *Behavior Rules* model. The simulation results are presented in Figure 54. In the log (Figure 54), we have different variables depending on the output port. In the port *ReceivingOut*, we have the following variables (explained in the order as they appear in the log):

- Simulation time: is the time when the event took place
- Type of communication: it can be *PHONE_MESSAGE*, TE*XT_MESSAGE* or *RADIO_MESSAGE*.
- Type of message: it can be *CALL_REQUEST* in the case of *PHONE_MESSAGE*, *RADIO_MESSAGE* or *NEW* in the case of *TEXT_MESSAGE*
- Device Id from: it has two attributes - type and id. For example MOBILEPHONE 1.
- Device Id to: it has two attributes - type and id. For example MOBILEPHONE 20.

```
00:00:11:000 [Behavior Rules::ReceivingOut: {PHONE_MESSAGE CALL_REQUEST MOBILEPHONE 1 MOBILEPHONE 20}]
        generated by model Behavior Rules
00:00:20:000 [Behavior Rules::topModelOut: {ANSWER MOBILEPHONE 20 MOBILEPHONE 1 0 “Establish
        Emergency Level 0”  1 20}] routed from taskDeviceFinished input port
00:00:30:000 [Behavior Rules::SendingOut: {MOBILEPHONE 20 LANDLINEPHONE 50 “Establish Emergency
        Level 0” 20 50}] generated by model Behavior Rules
00:10:00:000 [Behavior Rules::topModelOut: {SEND MOBILEPHONE 20 LANDLINEPHONE 50 0 “Establish
        Emergency Level 0” 20 50}] routed from taskDeviceFinished input port
00:10:10:000 [Behavior Rules::SendingOut: {EMAIL 20 EMAIL 51 “Establish Emergency Level 0” 20 51}]
        generated by model Behavior Rules
00:10:30:000 [Behavior Rules::topModelOut: {SEND EMAIL 20 EMAIL 51 0 “Establish Emergency
        Level 0”  20 51}] routed from taskDeviceFinished input port
00:10:40:000 [Behavior Rules::ReceivingOut: {PHONE_MESSAGE CALL_REQUEST LANDLINEPHONE 39
        LANDLINEPHONE 1}] generated by model Behavior Rules
00:50:00:000 [Behavior Rules::topModelOut: {ANSWER LANDLINEPHONE 20 LANDLINEPHONE 39 0 “Tell
        people to stay at home” 39 20}] routed from taskDeviceFinished input port
00:50:10:000 [Behavior Rules::taskDoActionOut: {“Transmit Emergency Alert to the population”
        00:10:00:000 55D6}] generated by model Behavior Rules
```

Figure 54 *Behavior Rules* log file when simulated with the input file in Figure 53.

For example, the out at time 00:00:11:000 means that *Behavior Rules* has decided to answer the phone call on the mobile from person 20.

In the port *SendingOut*, we have the following variables (explained in the order as they appear in the log):

- Simulation time: is the time when the event took place
- Device Id from interpreted as in the previous case.
- Device Id to interpreted as in the previous case.
- Command: it has three attributes content, receiver, and sender

For example, the out at time 00:10:10:000 means that *Behavior Rules* has decided to send the command “Establish Emergency Level 0” to person 51 using email.

In the port *topModelOut*, we have the following variables (explained in the order as they appear in the log):

- Task: is the task carried by the agent. It can be *SEND*, *ANSWER* or *DO_ACTION*.
- Device Id from interpreted as in the previous case.
- Device Id to interpreted as in the previous case
- Defective: takes the value 1 if the device used to transmit the command is faulty and 0 otherwise
- Command: it has three attributes content, receiver, and sender

For example, the output at time 00:50:00:000 means that person 20 has received the Command "Tell people to stay at home" from person 39 using the landline phone. The communication works well (i.e. defective value is equal 0).

At time 00:00:11:000, the model outputs a message through the *ReceivingOut* port, which is connected to *Receiving Filter* (see Figure 33b); this is a request to answer the phone call on the mobile. This output is generated based on the input bag at time 00:00:01:000 and the behavior of the agent defined in the XML file. Based on the figures above, we see that the agent receives two call requests at time 00:00:01:000, one from the mobile and one from the landline phone. Since the agent's priority to answer a device is *DEVICE_PRIORITY* (defined in the tag *AnswerPriorityType)* and the mobile has priory over the landline phone (defined in *AnswerDevicePriority*), the agent chooses to *ANSWER* the mobile, and this takes 10s (the *ReactionTime*). It sets the task *ANSWER* the landline phone in its to-do list.

At 00:00:15:000, the input in Figure 53 tells that the call on the landline is finished. Therefore, the agent removes this task from the to-do list. There is no output associated with this input.

Then, as seen in Figure 53, at time 00:00:20:000, the agent receives a new message indicating that the task *ANSWER* the phone call on the mobile has been finished with a command "Establish Emergency Level 0". This answer is immediately redirected to the *topModelOut* port as seen in Figure 54. Based on the *MessageBehavior* (defined in the XML file), the model selects the tasks to do and it adds them to the to-do list. Based on the other parameters defined in the XML and the to-do list, the next output is generated after the agent reaction time of 10s. In this case, the output is generated at time 00:00:30:000 as shown in Figure 54. The output means that the model has decided to send the command "*Establish Emergency Level 0*" to person 50 using the agent's mobile and calling to person 50 on the landline phone. The rest of the outputs in the log presented in Figure 54 are analyzed following a similar reasoning.

As we can see, the output generated by *Behavior Rules* model matches the expected output manually deduced using the XML, which meets the requirements in the Requirements document. In this way, we have verified the correct implementation of the *Behavior Rules*, which also matches the requirements defined by the experts in the NEP.

10.2. NEP DAM: Head of the NEP

In this section, we present a version of the *NEP Diffusion Abstract* model in which we instantiated 13 *Person* and 13 *Devices* models. The computerized model is automatically generated using the 13

XML files that represent these individuals (provided by the Agent Based models) and the program explained in section 8.2 to generate the *NEP Computerized model* automatically.

All the results presented in this section should be understood in the context that only the head of the plan is simulated and the people they have to manage are not included in the model. They are partial results we used to explain how we generate and analyze the simulation results of the *NEP Diffusion Abstract model* to be used in decision-making.

The inputs that trigger our simulation are generated by the three generators explained at the beginning of this section (*Command Generator, Devices State & Networks State*) using the three input files presented in Figure 55.

Command Generator takes as input the four commands (Establish Emergency Level 0, 1, 2 and 3) the NEP Director decides to transmit at four different moments of the emergency. *Device State* input is used to define the devices that change their state to faulty or recovered (if they were faulty). In this example, the Radiological Group Communication Device from person 13 is set to broken state at time 00:15:00:000. *Network State* takes as input an empty file because none of the networks fail.

As we explained in our architecture, we can update the state of the *Devices* and *Networks* models dynamically at runtime, by the values of the input files for *Device State* and *Network State* models. We can also easily define different Commands at different times and varied persons by changing the values in the input file of the *Command Generator* model. This allows us to run different scenarios for the same instantiation of the *NEP Diffusion Abstract model*.

```
Command Generator Input file
00:00:00:000 “Establish Emergency Level 0” – “NEP Director”
00:25:00:000 “Establish Emergency Level 1” - “NEP Director”
00:50:00:000 “Establish Emergency Level 2” - “NEP Director”
02:00:00:000 “Establish Emergency Level 3” - “NEP Director”
Devices State Input file
00:15:00:000 RADIOLOGICAL_GROUP_DEVICE 13 1
Networks State Input file
```

Figure 55 Input files for the *NEP Diffusion Abstract model*

In Figure 56, we show the simulation results of the *NEP Diffusion Abstract model* using the input files presented in Figure 55. The output of the model is the tasks finished by each agent (Person) as we already explained in the *Behavior Rules* model.

Each *Person* selects the tasks to do base on the values of the messages in the different input ports of the *Person* model and the behavior defined in the XML file as we explained in the *Behavior Rules* model. The messages in the input ports of a *Person* are determined by other *Person* models (i.e. the persons determine the messages to be transmitted through the *Devices* and *In-Situ*). Therefore, the behavior of the *NEP Diffusion Abstract* model is a complex one that emerges from the interactions between the *Person* models and the states of the *Devices* and *Networks* (i.e. if they are faulty or not).

The variables in the log presented in Figure 56, are the same as the ones in the port *topModelOut* of the model *Behavior Rules* we have just explained. They are interpreted in the same way. For example, the output at time 00:04:31:300 means that person 13 has received the Command “Establish Emergency Level 0” from the NEP Director using in-situ communications. The communication works well since the defective value is equal 0. The rest of the log is interpreted using the same reasoning.

```
00:04:31:300 [NEP::taskFinished [{ANSWER IN_PERSON 13 IN_PERSON “NEP Director” 0 “Establish
         Emergency Level 0” “NEP Director” 13}] routed from model Person13
00:04:32:400 [NEP::taskFinished[{SEND IN_PERSON “NEP Director” IN_PERSON 13 0 “Establish Emergency
         Level 0” “NEP Director” 13}]routed from model “NEP Director”
00:05:42:302 [NEP::taskFinished [{SEND RADIOLOGICAL_GROUP_DEVICE 13 RADIOLOGICAL_GROUP_DEVICE 19 0
         - 19 13}] routed from model Person13
00:07:03:700 [NEP::taskFinished [{ANSWER IN_PERSON 12 IN_PERSON “NEP Director” 0 “Establish
        Emergency Level 0” “NEP Director” 12}] routed from model Person12
00:07:04:406 [NEP::taskFinished [{SEND RADIOLOGICAL_GROUP_DEVICE 13 RADIOLOGICAL_GROUP_DEVICE 19 0
        - 19 13}] routed from model Person13
00:07:04:800[NEP::taskFinished[{SEND IN_PERSON “NEP Director” IN_PERSON 12 0 “Establish Emergency
         Level 0” “NEP Director” 12}]routed from model “NEP Director”
00:08:15:408 [NEP::taskFinished [{SEND RADIOLOGICAL_GROUP_DEVICE 13 RADIOLOGICAL_GROUP_DEVICE 19 0
        - 19 13}] routed from model Person13
00:09:36:100 [NEP::taskFinished [{ANSWER IN_PERSON 11 IN_PERSON “NEP Director” 0 “Establish
        Emergency Level 0” “NEP Director” 11}] routed from model Person11
...
```

Figure 56. ***NEP Diffusion Abstract model*** **log file when simulated with the input file in Figure 55**

If we want to extract useful information from the log, we need to process the results as indicated in the *Results Analysis* component of the architecture. In this example, we are interested in identifying the most used devices, how many people get each command, and who are the busiest people.

In this case, we use PowerBI (Microsoft 2015) for the log analysis but other software or programming language for statistical and big data analysis can be used (in the rest of the sections of this chapter we use Python because it allows to implement a script that automatically process the logs and generate the graphical results). As we explained in section 6.1.6, PowerBI is a tool for big data analysis that allows data visualization.

Identifying the most used devices allows us to identify the most critical networks in case a disruption occurs. By having an idea of the most used devices, we can simulate other scenarios where the network that handles those specific devices fails, and see what happens. Based on the results of those simulations, we will see if it really is a critical network or there are other ways to transmit the commands.

If all the devices and networks work as expected, knowing how many people get the command allows us to identify if the message transmission rules are well defined. It will also allow us to study the effect of failures in the devices and networks. For this purpose, we need to identify how many people were expected to get the command, and how many people got it in our simulation scenario.

Identifying the busiest people help decision-makers to identify the people with a heavy workload, and it allows them to allocate resources properly to balance the workload.

In Table 11, we represent the use of the devices using the field *Device Id from* in the log file. In this case, we can see that the most used device is the fax with 36 instances. It is followed by in-situ communications and the radiological group device. The other devices are not used since they do not appear in the log file, and therefore neither in the figure. These results point that a scenario where the Radiological Group network does not work should be simulated. The other devices are not used in this scenario because of the defined behavior of the people. Although some people have email, mobile, etc., based on the availability of the devices and their preferences defined in the person behavior, the emergent behavior of the model is the one we show in Table 11. In this scenario, the fax is mainly

used by the NEP Director to send commands to the people that are not in the same location. The NEP usually defines the use of the fax to send commands, so the command sent is registered.

Table 11. Visualization of simulation results. Use of the devices

Communication Device	#Uses
FAX	36
IN_PERSON	12
RADIOLOGICAL_GROUP_DEVICE	10

We can also study the number of people that get each command. In this case, 12 people got each command. Since there are 13 people in this version of the model and the NEP Director generates the command, everyone gets all the commands. Therefore, we can conclude that the breakdown of a device from person 13 has not affected the transmission of commands.

These results can also be used to identify who is the person in the plan that does more tasks, and the type of task each person does based on the *Task* attribute of the log (described in Figure 57). In this case, person 1 (the NEP Director) is the one with the heaviest load, and it only sends commands. The other ones (except person 13) only receive commands. None of the persons involved execute actions. These results point out that it would be a good idea to analyze different policies to relieve the work load of the NEP Director. One of these policies may be to add a person in the head of the NEP that helps the NEP director to transmit commands to other people. Analyzing this policy with our model will determine if it really helps or the overhead included in the communications has the opposite effect.

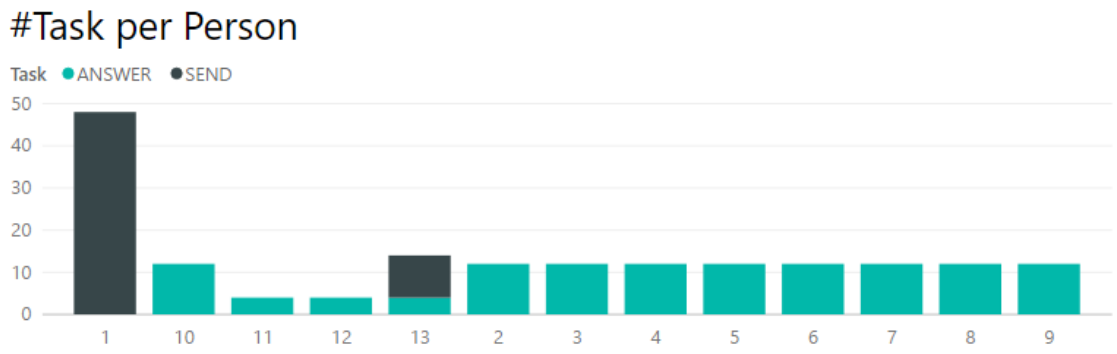


Figure 57 Number of tasks per person, classified by send and answer tasks.

Although these results are contextual, they can be automatically determined over a number of different scenarios as we will show in the next sections.

10.3. NEP DAM: Radiological Group

In this section we focus on the study of the Radiological Group. To conduct this analysis, we instantiate the NEP DAM with 149 individuals and their communications devices. These individuals include the Head of the NEP and the whole Radiological Group. To generate the graphical results we

have used Python because it allows integrating both processing the logs and generating the graphical results once the simulations are finished.

We have focused on studying what happens when the command “Establish Emergency Level 0” is decreed by the NEP Director and the specific communication device inside the Radiological Group fails with different probabilities. The failure may represent that the device runs out of battery, it does not receive a signal, it breaks, etc. We have simulated different scenarios where this device fails with different probabilities (i.e. 10%, 20%, etc.).The simulations represent a 95% Confidence Interval for the mean of people that receive the command “Establish Emergency Level 0”. The confidence interval is represented as notches in the plot (Figure 58). This analysis provides some information to decision makers and it is useful to validate our model.

Based on the NEP specifications, we know that 63 people should receive a command from the head of the radiological group. We also know that the Radiological group only uses a *radiological group device* (RGD, a specific device with mixed radio-phone communication).

Figure 58 shows the number of people that receive the command “Establish Emergency Level 0” when we simulate different probabilities of failure of the *RGD*. We use a box plot in which the triangle represents the mean, the horizontal line the median and the circles the outliers.

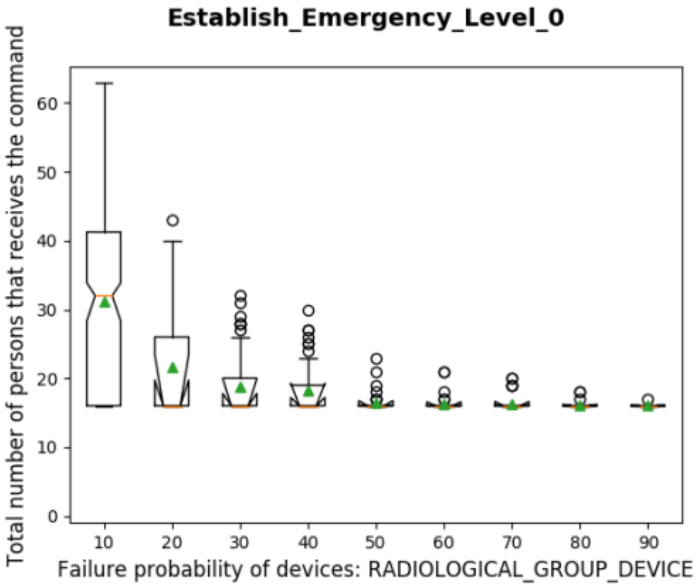


Figure 58 RGD failures

We can see that, regardless the failure probability, some people always receive the command. This number remains constant since they are part of the NEP leadership and they do not use the RGD to communicate. However, even with a 10% failure probability, in 75% of the cases, less than 40 people receive the command and the median is around 30. If the failure probability increases to 20%, the median is drastically reduced to less than 20 people. This value remains constant when the failure probability increases over 20%. Based on this analysis and taking into account the definitions of the NEP, we can conclude that we cannot afford a failure rate of only 10% in the RGDs because in more than 75% of the cases less than 40 people out of 63 receive the command.

Figure 59 shows how many times each device is used based on the failure probability of the RGD. We use these results to validate our model.

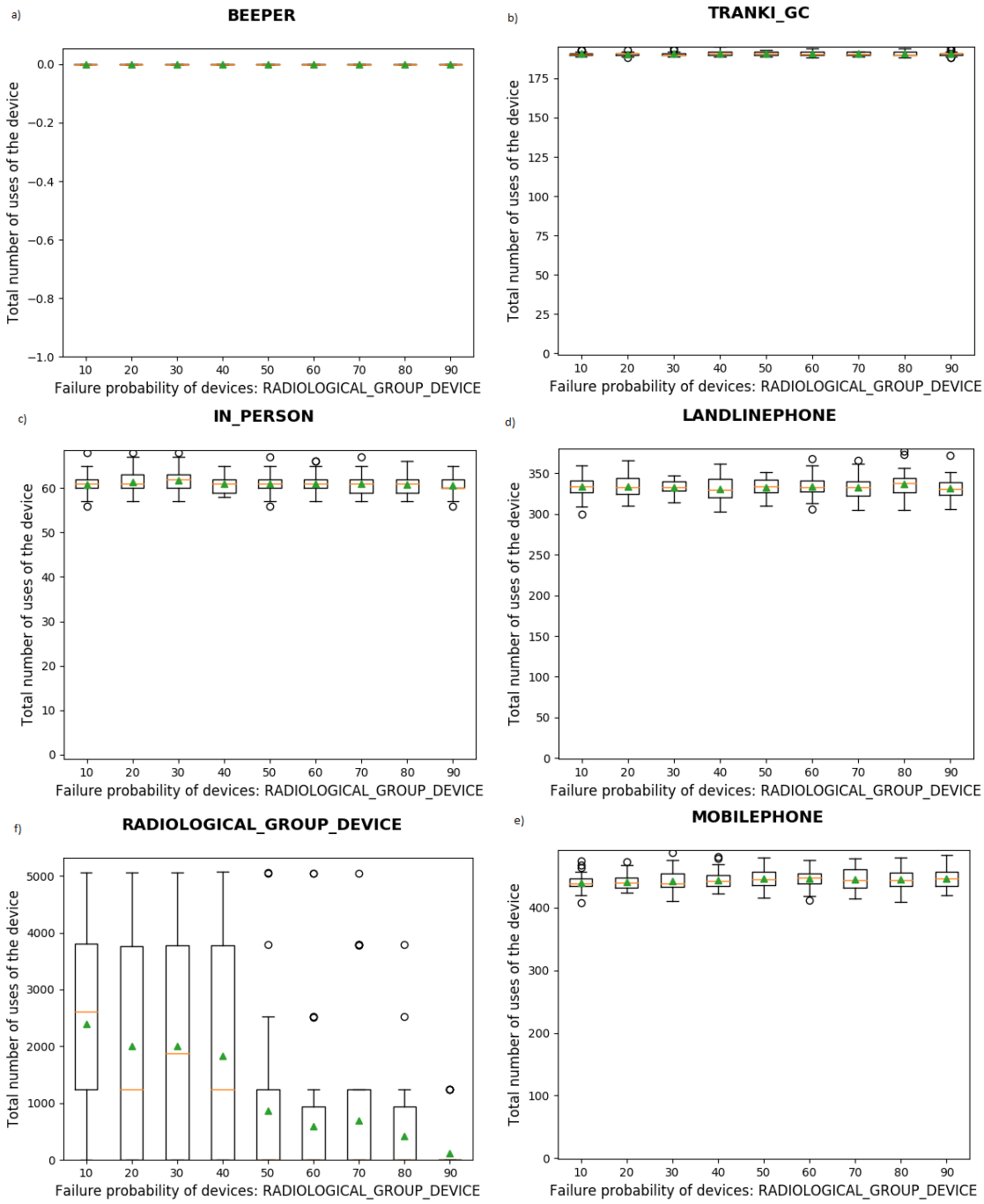


Figure 59 Number of activations of the different devices when the RGD fails with different probabilities.

In Figure 59 a), we can see that the beeper is not used (the mean, medium and quartiles are all zero). Although only the beeper is shown, we obtain the same results for fax, e-mail, private landline phone, two radio channels - REMAR and REMER -, satellite phone, and TrankiE - a phone-radio used by the police -. This result is correct, as the specification document says that none of these devices should be used by the radiological group.

In Figure 59 (b-e), we show that the data distribution is uniform when we simulate failures in the RGD. The number of attempts to establish the communication causes variability in the different simulations. These results validate the model based on the NEP specifications, which says that the Radiological Group only use the RGD. This restriction justifies why the plots in Figure 59 (a-e) are uniform for the different failure probabilities.

Figure 59 f) shows two different trends. When the failure probability is low (less than 50%), the number of activations of the RGD is high. The variability for each failure probability is also high (i.e. wide interquartile range). When the failure probability is 50% or greater, the number of activations is significantly reduced and the variability is lower. The mean of the number of RGD activations shows a decreasing trend. When the failure probability is low, there are many devices working. If a device does not fail, the owner keeps trying to communicate (i.e. the total number of activations of the device is high). But if they see that their device is not working, they stop using it. Therefore, when a device fails and the owner has something to send the information transmission process is blocked. In those cases, the number of activations for the devices is lower. An increase in the failure probability is translated in an increase of the number of devices broken. Then, the probability to block the information transmission increases. This explains why the mean decreases. Additionally, Figure 59 f) shows, that regardless the failure probability, there are cases (i.e. simulations) where the RGD is activated just one time. These results show that there is a critical person in the process, and they can block the whole information transmission if their device is broken (this has been confirmed analyzing the simulation logs and NEP specifications), which confirms that if the device of the Radiological Group head is broken, the whole process is blocked.

Based on these results, we can see we need to review the communications within the Radiological group. The simulation results allowed us to come with following questions that affect the organization: why the people within the radiological group cannot use their mobile phone or e-mail? Is there any security issue (e.g. authentication, encryption, etc.)? The discussion of these questions with decision makers will bring new scenarios to analyze to test different solutions. Then, to simulate a new scenarios they just update the model parameters in the XML files, such as the devices for each person or the communication devices that they can use with other people. Then, they run the program that automatically instantiates the DAM and generates the computerized model and runs the simulation.

10.4. NEP DAM: Health Group

In this section, we focus on the study of the Health Group. To conduct this analysis, we instantiate the NEP DAM with 107 individuals and their communications devices. These individuals include the Head of the NEP and the whole Health Group. As we have explained along the thesis, to generate the results presented in this section, we do not modify our model. We just run our program with different configuration parameters.

We have focused on studying what happens when the command “Establish Emergency Level 0” is decreed by the NEP Director and after a while he decide to establish the other emergency level (i.e. 1, 2 and 3). We have study a scenario where the mobile phone fails with different probabilities (i.e. 10%, 20%, etc.). As in the previous section, the simulations represent a 95% Confidence Interval for the mean of people that receive the command. The confidence interval is represented as notches in the plot (Figure 60).

Based on the NEP specifications, we know that 49 people should receive a command inside this group. The NEP establishes that inside the Health group both mobile and landline phone communications can be used. However, first responders only have access to the mobile. Although the NEP also allows the use of beeper inside the group, it does not specify who has access to this device. Therefore, in our analysis we will assume that anybody has beeper in order to be in the conservative side.

Figure 60 shows the number of people that receive the command "Establish Emergency Level 0" when we simulate different probabilities of failure of the mobile phone and Figure 61 for "Establish Emergency Level 1". The plots for "Establish Emergency Level 2" (and Level 3) are the same as the one in Figure 61.

Since the results are automatically generated, we use the same type of box plot as in the previous section (i.e. the triangle represents the mean, the horizontal line the median and the circles the outliers).

In Figure 60, we can see that, regardless the failure probability, some people always receive the command. This people are part of the NEP leadership and they have alternative communications means such as landline phone. However, even with a 10% failure probability, there are people that do not receive the command and the mean and median are around 35. As the failure probability increases, both the mean and the median decreases. When the failure probability is over 50%, we get uniform results with some variability. In these cases, both the mean and median are under 10 people.

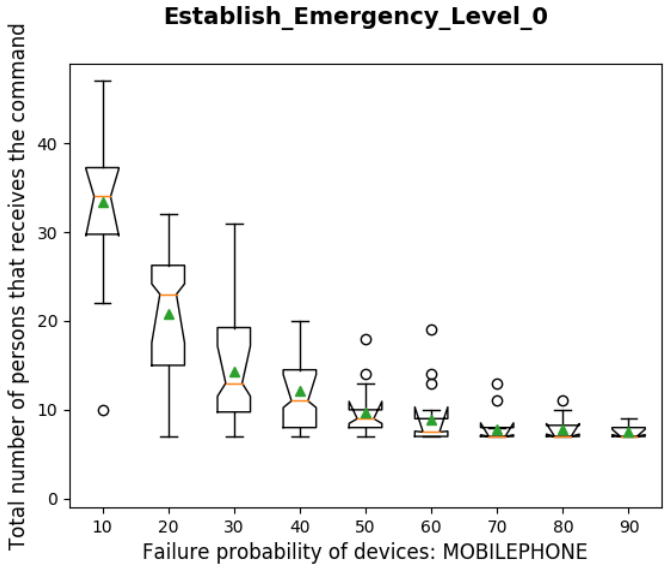


Figure 60 "Establish Emergency Level 0". Mobile phone failures

In Figure 61, we can see that in most of the cases, even with low failure probability the people do not receive the other commands. These results are explained taking into account the behavior of the people. In this scenario, the individuals send the messages in a FIFO order and they do not have a limit on the times they try to transmit the command. Therefore, once the device of a person who only has one communication mechanism fails, the information transmission process can be blocked.

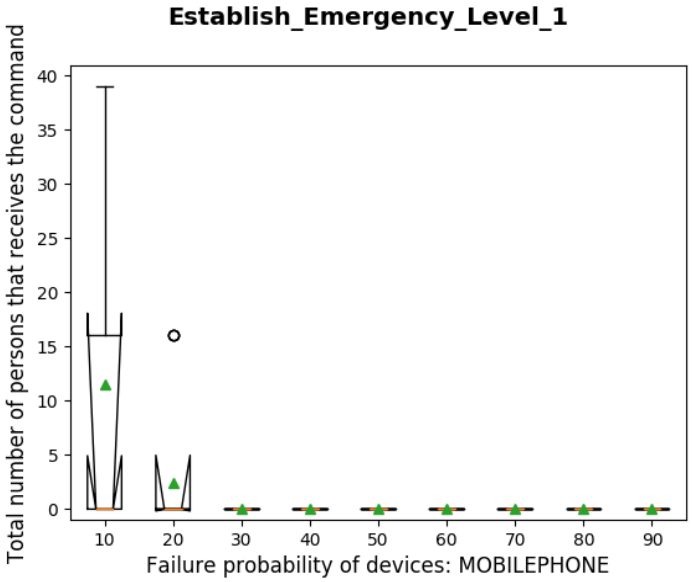


Figure 61 "Establish Emergency Level 1". Mobile phone failures

Based on this analysis and taking into account the NEP, we can conclude that we cannot afford a failure rate of only 10% in the mobile phone because in more than 75% we have people that do not receive the command. Moreover, the behavior we have studied is not efficient since the information transmission process is blocked easily.

Recalling the assumptions we made in section 9.1 to make studies using the network model, these results justify assumption 5 (i.e. defining the communication in mobile and landline phone using a single type of link). As we can see in Figure 61, when the mobile phone fails (see 90% failure probability), even the landline phone is working, most of the people do not receive the command.

These results also validates the ones presented in section 9.2, were we identify the isolation of the members of the Health group when a downfall in the phone communication channel happens. These results also show that when the mobile fails, there are many isolated individuals.

Figure 62 shows how many times each device is used based on the failure probability of the mobile.

In Figure 62 a), we can see that the beeper is not used (the mean, medium and quartiles are all zero). Although only the beeper is shown, we obtain the same results for fax, e-mail, private landline phone, two radio channels - REMAR and REMER -, satellite phone, and TrankiE - a phone-radio used by the police -. This result is correct, as we assume that anybody inside the Health group has access to the beeper and the specification document says that the other devices are not use by the Health group.

In Figure 62 (b and e), we show that the data distribution is uniform when we simulate failures in the mobile. The number of attempts to establish the communication causes variability in the different simulations. These results validate the model based on the NEP specifications, which says that the Health group do not use these two devices. These restrictions in the use of devices justifies why the plots in Figure 62 (a, b and e) are uniform for the different failure probabilities.

Figure 62 c) shows that the medium for in-situ communications is the same for all failure probabilities. However, when the failure probability of the mobile phone is low (10 and 20%), there is higher variability (although the medium remains constant). These results are justified taken into account that only in these cases the other commands (i.e. Establish emergency level 1, 2, 3) are transmitted.

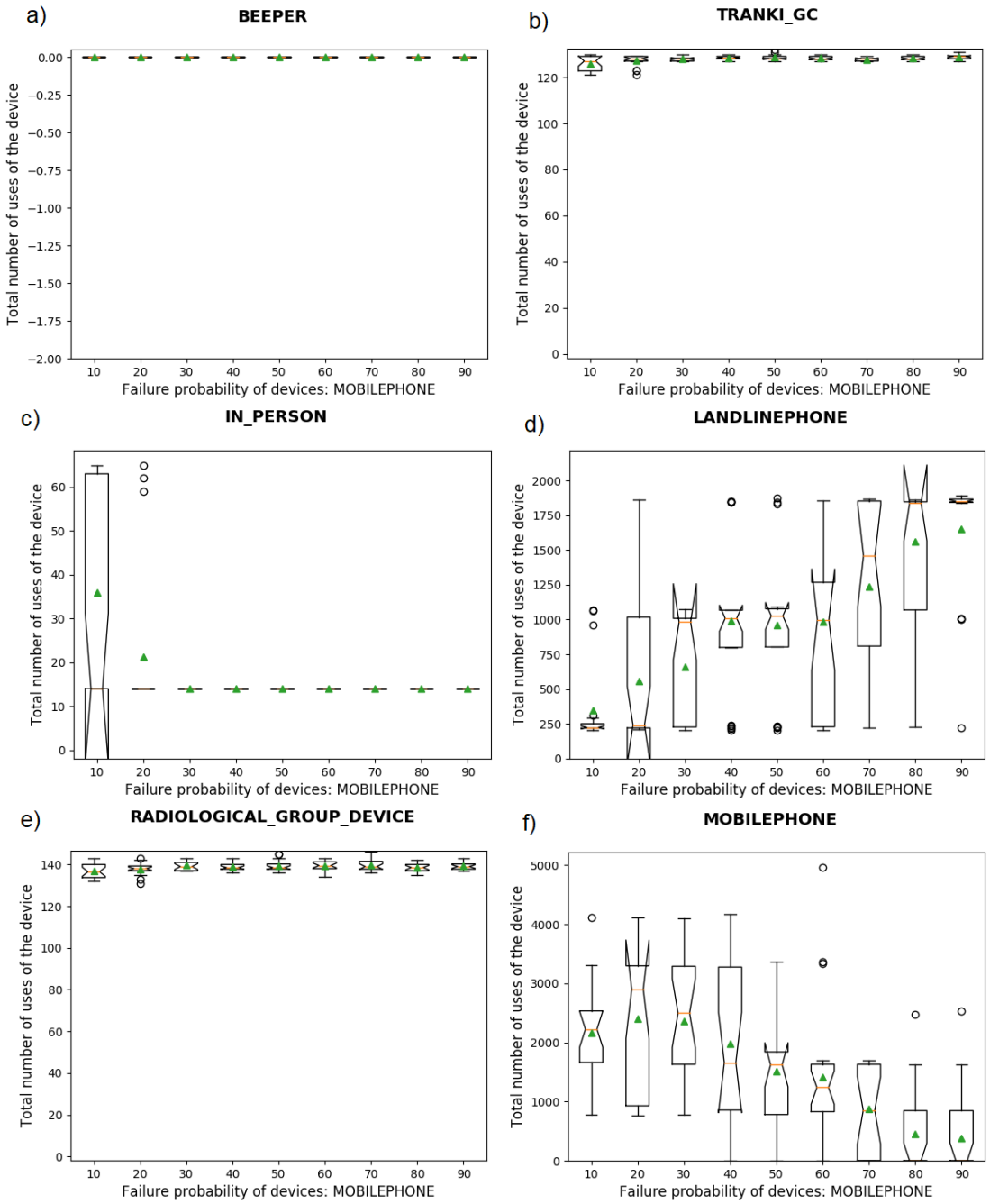


Figure 62 Number of activations of the different devices when the mobile phone fails with different probabilities

Figure 62 d) shows an increasing trend in the mean and medium in the use of landline phone and Figure 62 f) a decreasing trend for the mobile. When the mobile phone fails, the indivuduals who has access to landline phone use it more and stop using the mobile. Moreover, since there are people that

only have access to mobile, the probability to establish the communication is lower. As people do not know why there is not response, they keep trying to communicate.

Based on these results, we can see that we need to review the communications within the Health group. We have also found that the behavior of the individuals is also critical in the information transmission process. With these simulations, we have identified a key parameter to study: the number of attempts before we can accept a communication is lost (N_a). We couldn't have reached this conclusion about the behavior just analyzing the network model.

Since this parameter (i.e. N_a) is not included in the behavior defined in the Agent-Based model, to make these studies we will need to modify the model. One advantage of the proposed architecture is that we can enrich the model adding new features in an incremental way. In this case, we will need to update the XML files where the behavior of the agents is defined and just update one component of the DAM: the Behavior Rules (see Figure 33). These analyses are included as future research lines of this thesis (see section 11.2).

Chapter 11. Conclusions and Future Work

In this chapter, we present the conclusions of this thesis. We discuss the applicability of the proposed framework to design mitigation and disaster management plans and for policy decision making. We also present the limitations of this work and state the future research lines.

11.1. Conclusions

In this thesis, we have introduced a Framework to design resilient organizations. The framework is based on the methodology introduced by Perez Rios (2010) to design viable organizations and an architecture to simulate diffusion processes in multiplex networks. The Framework is used to identify the pathologies of an organization and to study the communication inside an organization (a key element to be resilient). We have focused on the application of this Framework to design resilient emergency plans.

To build the Framework, we first carried a literature review about organizational resilience. The main contributions of this review are a conceptualization of organizational resilience, a four-level Maturity Model for Organizational Resilience (MMOR) based on the development of the abilities or capacities the organization has to deal with disruptive events and the identification of two streams to measure organizational resilience. The first one is focused on providing an estimate of organizational resilience potential based on the characteristics, abilities or capabilities of the organization before a disruptive event occurs. The second stream will measure organizational resilience once a disruptive event has occurred. Measuring resilience after a disruptive event has occurred will help to provide better estimates of the resilience potential studying the correction between the two measures.

There are different works (see Chapter 2) that focus on the measurement organizational resilience. However, we did not identify a measure that allows us to quantify the resilience of the organization. Therefore, we can conclude that we cannot yet compare two organizations and conclude which one is more resilient.

Based on the review of organizational resilience and taking into account the definition of viable organizations, we have established that resilient organizations should be viable. Both viable and resilient organizations should be designed to survive to changes in the environment. Therefore, we conclude that resilient organizations fit the principles of the VSM.

Since resilient organizations fits the principles of the VSM, we have proposed the application of the VSM and the methodology introduced by Pérez Ríos (2010) to the design of resilient organizations. Using this methodology we can identify the pathologies that the organization suffers. Once we have identified the pathologies, we can make changes in the organization to overcome them. This approach allows improving the organization tackling the pathologies identified.

Although the organization may suffer different pathologies, we have focused on the study of one of them: the communications inside the organization. We chose this pathology because this thesis started as collaboration with the Civil Protection Agency to design resilient emergency plans and several reviews of emergencies highlighted the importance of the communications to be resilient.

The communications inside an organization can be modeled as an information diffusion process in multiplex networks. We reviewed existing architectures to study information diffusion process and we found that the one proposed by Bouanan et al. (2016) could be applicable to study the resilience of communications. We test its applicability to our case study (a Spanish NEP) and we identify some drawback when including the behavior of the people involved in the plan and when simulating different scenarios (see section 5.5). The main drawbacks were based on the use of MySQL to define the behavior of the agents and the use of DS-DEVS to simulate dynamic networks.

Based on this architecture, in this thesis we have proposed a new one and a development process based on a formal modeling and simulation methodology to simulate diffusion processes in multiplex networks (see Chapter 6). We used Agent Based Modeling techniques to identify the agents involved in the diffusion process and their behavior, and Network Theory to define the relations between these agents. Both the Network and Agent Based models were used to develop the Diffusion Abstract Model. We used DEVS to define the Diffusion Abstract Model because many other formalisms can be transformed into DEVS models and the advantages presented in Chapter 3 (e.g. we can use a well-defined simulation algorithm, we can develop hierarchical models in a modular way, etc.).

To obtain the architecture and development process we followed a bottom up approach. We generalized the model and the steps we used to study the communications in the NEP. In the thesis, we introduced the general architecture first to explain the application of development process to build a model from scratch.

In Chapters 7 – 10, we explained how to apply the architecture proposed in Chapter 6 to study the communications inside an organization. We have used as a case study a real NEP from Spain because they are complex organizations and the communications are a key element for the management of the emergency. Moreover, the literature remarks the importance of improving emergency plans and we have access to data of a real NEP provided by the Civil Protection Agency.

We have focused on the study of the resilience of the communication network analyzing scenarios with different failure probabilities for the communication devices. We have shown how these scenarios and results are generated automatically using the program explained in Chapter 8. The analysis of these results (i.e. including the human in the loop as shown in Figure 21) have provided useful information to design new policies (e.g. including new communication mechanism in some groups, establish the number of attempts to transmit a command, etc.) to make the system more resilient (i.e. less susceptible to the failures in the communications mechanism). These new policies can be tested using the model presented in this thesis as test suite.

We identified that the behavior of the individuals involved in the NEP (i.e. the nodes in the Network model) also affects the resilience of the network. For example, in section 10.4, we identify that the number of attempts before we can accept that a communication is lost (N_a) may be critical in to improve the information transmission process. Having a person that does not transmit the information as expected can stop the whole process if it is a critical node. The lack of these studies is a limitation of this thesis, and as we mention in section 11.2, it will be a future research line.

The model we developed has the advantage that allows us to easily modify the behavior of the agents. We can study different behavior of the individuals just modifying the Agent's parameters defined in the XML file. If we want to develop more advance models for the behavior of the agent (i.e. including more parameters), we need to update the XML structure and therefore the DAM. Having the

DAM formally defined and implemented in DEVS has one advantage: it is modular. We just need to update the definition of the Behavior Rules model inside the Person model (see Figure 33). Therefore, we reduce developing time since we just need to redefine, implement, verify and validate one component of the model. This advantage was maintained in the extrapolation process when we made the generalization to obtain the architecture and development process.

The case study shown in this thesis (i.e. the study of the communications inside the NEP) shows the applicably of this architecture for policy making. The models developed under the architecture can be used as a test suite for decision makers before implementing their policies in the real world. Having a test suite will allow policymakers to have a tool for testing new policies before implementing them in the real world, thus saving time, costs, and reducing the effects of inadequate policies.

Although we have shown the applicability to study the communications inside organizations, it can also be applied in other field such as design policies to control the diffusion of diseases, design policies to control the spread of fake news over the social media, etc.

Our case study also demonstrates the applicability of whole framework as a kernel to design the resilient mitigation and disaster management plans. It is really significant for emergency agencies such as the Federal Emergency Management Agency (FEMA), whose primary objective is to coordinate the response to a disaster in the United States when the local and state authorities do not have enough resources.

11.2. Future work

The future research lines of this thesis will aim to tackle the limitations of this research and to improve the framework we have defined.

In this thesis, we build a model of the NEP. All the models are abstraction of the reality that can be improved. Analyzing the Health group, we identify an important parameter to include in our model: the number of attempts before we can accept a communication is lost (N_a) (see section 10.4). We will work on the modification of the model to improve it and make the behavior of the people more real. With this new model, we will study more scenarios where we can analyze the combined effect of modifications in the behavior of the people and the reliability of the communication mechanism.

We will also apply this Framework to study the communications in other emergency plans and other organizations. The application of the Framework to study other emergency plans and organizations will provide evidence of its applicability and the opportunity to check the architecture.

Having different models to study the communications inside organizations we can find similarities. Based on the similarities, we will define and implement metamodels to study the communications in different organizations. We will design a tool with a graphical interface to instantiate the metamodels and automatically generate different components of the architecture: the Network Model, the Agent Based Model based, the Diffusion Abstract Model and its computerized version.

These improvements will be oriented to demonstrate the applicability of our framework to aid decision makers in policy driving and the design of mitigation and disaster management plans using our architecture as a test suite.

Another limitation of the thesis is that we have just study one the pathologies introduced by Pérez Ríos (2010): the one related to communications. Moreover, we have focus on the communications between people, but gathering data from sensors and early warning systems can also be critical. Future research lines will aim to study this other type of communications and other pathologies to improve the framework we have proposed.

Other research line will aim to advance in the topic of resilience. We will focus on providing a quantitative measure to evaluate the resilience of the organizations. With this measure, we will have a double objective. First, we want to be able to compare two organizations in terms of its resilience. Second, we want to be able to provide feedback to the organization about its weaknesses and how to improve its level of resilience.

Finally, we will aim to apply the proposed architecture to study other diffusion processes in multiplex networks in different domains such as the spread of diseases, rumors over populations, etc.

.

Bibliography

Acquaah, M., Amoako-Gyampah, K. & Jayaram, J., 2011. Resilience in family and nonfamily firms: An examination of the relationships between manufacturing strategy, competitive strategy and firm performance. *International Journal of Production Research*, 49(18), pp.5527–5544.

Afgan, N.H., 2010. Resilience of company management system. In PICMET '10 - Portland International Center for Management of Engineering and Technology, Proceedings - Technology Management for Global Economic Growth. pp. 1–8.

Aguirre, B.E., Dynes, R.R., Kendra, J. & Connell, R., 2005. Institutional resiliance and disaster planning for new hazards: Insights from hospitals. *Journal of Homeland Security and Emergency Management*, 2.

Akbar, M., Aliabadi, S., Patel, R. & Watts, M., 2013. A fully automated and integrated multi-scale forecasting scheme for emergency preparedness. *Environmental Modelling and Software*, 39, pp.24–38.

Alblas, A. & Jayaram, J., 2015. Design resilience in the fuzzy front end (FFE) context: An empirical examination. *International Journal of Production Research*, 53(22), pp.6820–6838.

Aleksic, A., Stefanović, M., Arsovski, S. & Tadić, D., 2013. An assessment of organizational resilience potential in SMEs of the process industry, a fuzzy approach. *Journal of Loss Prevention in the Process Industries*, 26(6), pp.1238–1245.

Alexiou, A., 2014. Taming the waves of adversity: Exploring the multidimensional construct of organizational resilience. *Managing Emerging Technologies for Socio-Economic Impact*, pp.340–353.

Alonso, A. & Bressan, A., 2015. Resilience in the context of italian micro and small wineries: An empirical study. *International Journal of Wine Business Research*, 27(1), pp.40–60.

Ambulkar, S., Blackhurst, J. & Grawe, S., 2015. Firm's resilience to supply chain disruptions: Scale development and empirical examination. *Journal of Operations Management*, 33–34, pp.111–122.

Angourakis, A., Santos, J.I., Galán, J.M. & Balbo, A.L., 2015. Food for all: An agent-based model to explore the emergence and implications of cooperation for food storage. *Environmental Archaeology*, 20(4), pp.349–363.

Annarelli, A. & Nonino, F., 2016. Strategic and operational management of organizational resilience: Current state of research and future directions. *Omega*, 62, pp.1–18.

Apneseth, K., Wahl, A.M. & Hollnagel, E., 2013. Measuring resilience in integrated planning. *Oil and Gas, Technology and Humans: Assessing the Human Factors of Technological Change*, pp.129–145.

Araúzo, J.A., Pajares, J. & Lopez-Paredes, A., 2010. Simulating the dynamic scheduling of project portfolios. *Simulation Modelling Practice and Theory*, 18(10), pp.1428–1441.

Asgary, A., Kong, A. & Levy, J., 2009. Fuzzy-Jess expert system for indexing business resiliency. In *TIC-STH'09: 2009 IEEE Toronto International Conference - Science and Technology for Humanity*. pp. 153–158.

Ates, A. & Bititci, U., 2011. Change process: A key enabler for building resilient SMEs. *International Journal of Production Research*, 49(18), pp.5601–5618.

Bañuls, V.A., Turoff, M. & Hiltz, S.R., 2013. Collaborative scenario modeling in emergency management through cross-impact. *Technological Forecasting and Social Change*, 80(9), pp.1756–1774.

Barros, F.J., 1997. Modeling formalisms for dynamic structure systems. *ACM Transactions on Modeling and Computer Simulation*, 7(4), pp.501–515.

Bastian, M., Heymann, S. & Jacomy, M., 2009. Gephi: An open source software for exploring and manipulating networks. *ICWSM*, 8, pp.361–362.

Battiston, F., Nicosia, V. & Latora, V., 2017. The new challenges of multiplex networks: Measures and models. *European Physical Journal: Special Topics*, 226(3), pp.401–416.

Bauernhansl, T., Mandel, J. & Diermann, S., 2012. Evaluating changeability corridors for sustainable business resilience. In *Procedia CIRP*. pp. 364–369.

Beer, S., 1981. Brain of the firm: the managerial cybernetics of organization, J. Wiley New York.

Beer, S., 1985. Diagnosing the System for Organizations, Chichester: Wiley.

Beer, S., 1979. *The Heart of Enterprise*, Chichester: Wiley.

Beer, S., 1989. The viable system model: its provenance, development, methodology and pathology". In R. Espejo & R. Harnden, eds. *The Viable System Model, Interpretations and Applications of Stafford Beer's VSM*. Chichester: Wiley.

Beermann, M., 2011. Linking corporate climate adaptation strategies with resilience thinking. *Journal of Cleaner Production*, 19(8), pp.836–842.

Bell, M., 2002. The five principles of organizational resilience. *Gartner Research.[Online]*, pp.2–4.

Bergström, J., van Winsen, R. & Henriqson, E., 2015. On the rationale of resilience in the domain of safety: A literature review. *Reliability Engineering & System Safety*, 141, pp.131–141.

Berman, E., 2009. Small Business Resilience. *Industrial Management*, 51(1), p.6.

Bhamidipaty, A., Lotlikar, R. & Banavar, G., 2007. RMI: a framework for modeling and evaluating the resiliency maturity of IT service organizations. In *IEEE International Conference on Services Computing (SCC 2007)*. Salt Lake City, UT, pp. 300–307.

Bhamra, R., Burnard, K. & Dani, S., 2015. Resilence. The Concept, a Literature Review and Future Directions. In R. Bhamra, ed. *Organizational Resilience. Concepts, Integration and Practice*. CRC Press, pp. 3–29.

Bhamra, R., Dani, S. & Burnard, K., 2011. Resilience: the concept, a literature review and future directions. *International Journal of Production Research*, 49(18), pp.5375–5393.

Biggs, D., Hall, C. & Stoeckl, N., 2012. The resilience of formal and informal tourism enterprises to disasters: Reef tourism in Phuket, Thailand. *Journal of Sustainable Tourism*, 20(5), pp.645–665.

Bouanan, Y., 2016. Contribution à une architecture de modélisation et de simulation à événements discrets : application à la propagation d'information dans les réseaux sociaux. Bordeaux.

Bouanan, Y., Zacharewicz, G., Vallespir, B., Ribault, J. & Diallo, S.Y., 2016. DEVS based Network : Modeling and Simulation of Propagation Processes in a Multi-Layers Network. In *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016*. Pasadena, CA, USA.

Boza, A. & Poler, R., 2013. Enhancing enterprise resilience through enterprise collaboration. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*. pp. 688–693.

Braes, B. & Brooks, D., 2010. Organisational resilience: a propositional study to understand and identify the essential concepts. *Proceedings 3rd Australian Security and Intelligence Conference*, (November), pp.14–22.

Braes, B. & Brooks, D., 2011. Organisational Resilience: Understanding and identifying the essential concepts. In *Safety and Security Engineering IV*. WIT Press, p. 117.

van Breda, A.D., 2016. Building Resilient Human Service Organizations. *Human Service Organizations Management, Leadership and Governance*, 40(1), pp.62–73.

Brewton, K., Danes, S., Stafford, K. & Haynes, G., 2010. Determinants of rural and urban family firm resilience. *Journal of Family Business Strategy*, 1(3), pp.155–166.

Brinkmeier, M., Fischer, M., Grau, S., Schäfer, G. & Strufe, T., 2009. Methods for Improving Resilience in Communication Networks and P2P Overlays. *PIK Praxis der Informationsverarbeitung und Kommunikation*, 32(1), pp.64–78.

Bruneau, M., Chang, S.E., Eguchi, R.T., Lee, G.C., O'Rourke, T.D., Reinhorn, A.M., Shinozuka, M., Tierney, K., Wallace, W.A. & von Winterfeldt, D., 2003. A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthquake spectra*, 19(4), pp.733–752.

Bullmore, E. & Sporns, O., 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3), pp.186–198.

Burnard, K. & Bhamra, R., 2011. Organisational resilience: development of a conceptual framework for organisational responses. *International Journal of Production Research*, 49(18), pp.5581–5599.

Caralli, R.A., Curtis, P.D., Allen, J.H., White, D.W. & Young, L.R., 2010. Improving operational resilience processes: The CERT® resilience management model. *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust*, pp.1165–1170.

Carroll, J.S., 1998. Organizational Learning Activities in High-Hazard Industries: the Logics Underlying Self-Analysis. *Journal of Management Studies*, 35(6), pp.699–717.

Chami, G.F., Molyneux, D.H., Kontoleon, A.A. & Dunne, D.W., 2013. Exploring network theory for mass drug administration. *Trends in Parasitology*, 29(8), pp.370–379.

Chand, A.M. & Loosemore, M., 2016. Hospital learning from extreme weather events: using causal loop diagrams. *Building Research and Information*, pp.1–14.

Chang, N.-B., Ning, S.-K. & Chen, J.-C., 2006. Multicriteria relocation analysis of an off-site radioactive monitoring network for a nuclear power plant. *Environmental management*, 38(2), pp.197–217.

Chen, X., Meaker, J.W. & Zhan, F.B., 2006. Agent-Based Modeling and Analysis of Hurricane Evacuation Procedures for the Florida Keys. *Natural Hazards*, 38(3), pp.321–338.

Chewning, L., Lai, C. & Doerfel, M., 2012. Organizational Resilience and Using Information and Communication Technologies to Rebuild Communication Structures. *Management Communication Quarterly*, 27(2), pp.237–263.

Chopra, S.S. & Khanna, V., 2014. Understanding resilience in industrial symbiosis networks: Insights from network analysis. *Journal of Environmental Management*, 141(April 2014), pp.86–94.

Cook, A., Blom, H.A.P., Lillo, F., Mantegna, R.N., Micciche, S., Rivas, D., Vazquez, R. & Zanin, M., 2015. Applying complexity science to air traffic management. *Journal of Air Transport Management*, 42, pp.149–158.

Coutu, D.L., 2002. How resilience works. *Harvard business review*, 80(5), pp.46–56.

Cozzo, E., Baños, R.A., Meloni, S. & Moreno, Y., 2013. Contact-based Social Contagion in Multiplex Networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 88(5), pp.1–5.

Crichton, M.T., Ramsay, C. & Kelly, T., 2009. Enhancing organizational resilience through emergency planning: Learnings from cross-sectoral lessons. *Journal of Contingencies and Crisis Management*, 17(1), pp.24–37.

Csardi, G. & Nepusz, T., 2006. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), pp.1–9.

Dalziell, E.P. & Mcmanus, S.T., 2004. Resilience, Vulnerability, and Adaptive Capacity: Implications for System Performance. *International Forum for Engineering Decision Making*, p.17.

Danes, S., Lee, J., Amarapurkar, S., Stafford, K., Haynes, G. & Brewton, K., 2009. Determinants of family business resilience after a natural disaster by gender of business owner. *Journal of Developmental Entrepreneurship*, 14(4), pp.333–354.

Demmer, W., Vickery, S. & Calantone, R., 2011. Engendering resilience in small-and medium-sized enterprises (SMEs): A case study of Demmer Corporation. *International Journal of Production Research*, 49(18), pp.5395–5413.

Deng, Y., Li, Q. & Lu, Y., 2015. A research on subway physical vulnerability based on network theory and FMECA. *Safety Science*, 80, pp.127–134.

Dervitsiotis, K.N., 2004. Navigating in turbulent environmental conditions for sustainable business excellence. *Total Quality Management & Business Excellence*, 15(5–6), pp.807–827.

Dewald, J. & Bowen, F., 2010. Storm clouds and silver linings: Responding to disruptive innovations through cognitive resilience. *Entrepreneurship: Theory and Practice*, 34(1), pp.197–218.

Doe, P.J., 1994. Creating a resilient organization. *Canadian Business Review*, 21, p.22.

De Domenico, M., Porter, M.A. & Arenas, A., 2014. MuxViz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, p.cnu038.

Dunn, S. & Wilkinson, S.M., 2015. Increasing the resilience of air traffic networks using a network graph theory approach. *Transportation Research Part E: Logistics and Transportation Review*, In press.

Edmonds, B., 2001. The Use of Models - making MABS more informative An Analysis of Modelling Understanding Multi-Actor Systems Through Modelling With MAS Abstraction Analysis of target system.

Erol, O., Henry, D. & Sauser, B., 2010. Exploring resilience measurement methodologies. In *20th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2010*. pp. 302–322.

Erol, O., Henry, D., Sauser, B. & Mansouri, M., 2010. Perspectives on measuring enterprise resilience. In *2010 IEEE International Systems Conference Proceedings, SysCon 2010*. pp. 587–592.

Erol, O., Mansouri, M. & Sauser, B., 2009. A framework for enterprise resilience using service oriented architecture approach. In *2009 IEEE International Systems Conference Proceedings*. pp. 127–132.

Erol, O., Sauser, B.J. & Mansouri, M., 2010. A framework for investigation into extended enterprise resilience. *Enterprise Information Systems*, 4(2), pp.111–136.

Espinosa-Paredes, G., Nuñez-Carrera, A., Laureano-Cruces, A., Vázquez-Rodríguez, A. & Espinosa-Martinez, E., 2008. Emergency management for a nuclear power plant using fuzzy cognitive maps. *Annals of Nuclear Energy*, 35(12), pp.2387–2396.

Estrada, E. & Gómez-Gardeñes, J., 2014. Communicability reveals a transition to coordinated behavior in multiplex networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 89(4), pp.1–5.

European Commission, 2016. Building Resilience : The EU's approach. Available at: http://ec.europa.eu/echo/files/aid/countries/factsheets/thematic/resilience_en.pdf [Accessed July 12, 2017].

European Commission, 2017. Resilience as a strategic priority of the external action of the EU. Roadmap - Ares(2017)1137007. Available at: https://ec.europa.eu/info/law/better-regulation/initiatives/ares-2017-1137007_en [Accessed July 12, 2017].

Eusgeld, I., Kroger, W., Sansavini, G., Schlapfer, M. & Zio, E., 2009. The role of network theory and object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures. *Reliability Engineering and System Safety*, 94(5), pp.954–963.

Fang, C., Marle, F., Zio, E. & Bocquet, J.C., 2012. Network theory-based analysis of risk interactions in large engineering projects. *Reliability Engineering and System Safety*, 106, pp.1–10.

Fiksel, J., 2003. Designing Resilient, Sustainable Systems. *Environmental Science and Technology*, 37(23), pp.5330–5339.

Fiksel, J., 2006. Sustainability and resilience: toward a systems approach. *Sustainability: Science, Practice, & Policy*, 2(2), pp.14–21.

Folke, C., Hahn, T., Olsson, P. & Norberg, J., 2005. Adaptive Governance of Social-Ecological Systems. *Annual Review of Environment and Resources*, 30(1), pp.441–473.

Freeman, S.F., Hirschhorn, L. & Triad, M.M., 2003. Moral purpose and organizational resilience: Sandler O'Neill & Partners, LP in the aftermath of September 11, 2001. In *Academy of Management Proceedings*. pp. B1--B6.

Galán, J.M., Izquierdo, L., Izquierdo, S., Santos, J.I., Olmo, R. del, Lopez-Paredes, A. & Edmonds, B., 2009. Errors and Artefacts in Agent-Based Modelling. *Journal of Artificial Societies and Social Simulation*, 12(1).

Galán, J.M., Lopez-Paredes, A. & del Olmo, R., 2009. An agent-based model for domestic water management in Valladolid metropolitan area. *Water Resources Research*, 45(5), pp.1–17.

Gallos, L.K., Song, C., Havlin, S. & Makse, H.A., 2007. Scaling theory of transport in complex biological networks. *Proceedings of the National Academy of Sciences*, 104(19), pp.7746–7751.

Garnett, J.L. & Kouzmin, A., 2007. Communicating throughout Katrina: Competing and complementary conceptual lenses on crisis communication. *Public Administration Review*, 67, pp.171–188.

Gibson, C. & Tarrant, M., 2010. A "Conceptual Models" approach to organisational resilience. *The Australian Journal of Emergency Management*, 25(2), pp.8–14.

Gilbert, N., 2007. *Agent Based Models*, London: Sage.

Gilly, J., Kechidi, M. & Talbot, D., 2014. Resilience of organisations and territories: The role of pivot firms. *European Management Journal*, 32(4), pp.596–602.

Gomes, J.O., Borges, M.R.S., Huber, G.J. & Carvalho, P.V.R., 2014. Analysis of the resilience of team performance during a nuclear emergency response exercise. *Applied Ergonomics*, 45(3), pp.780–788.

Gómez, S., Díaz-Guilera, A., Gómez-Gardeñes, J., Pérez-Vicente, C.J., Moreno, Y. & Arenas, A., 2013. Diffusion Dynamics on Multiplex Networks. *Physical Review Letters*, 110(2), p.28701.

Goverment of Canada, 2014. Canada in a Changing Climate: Sector Perspectives on Impacts and Adaptation F. J. Warren & D. S. Lemmen, eds., Ottawa, ON.

Government of Canada, Adaptation and climate resilience. Available at: https://www.canada.ca/en/services/environment/weather/climatechange/pan-canadian-framework/adaptation-climate-resilience.html [Accessed September 26, 2017].

Grande, O. & Trucco, P., 2008. Resilience analysis of civil defence organization: A fuzzy cognitive Map based approach. In *9th International Conference on Probabilistic Safety Assessment and Management 2008, PSAM 2008*. pp. 1542–1549.

Granell, C., Gomez, S. & Arenas, A., 2013. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical Review Letters*, 111(12), pp.1–10.

Grøtan, T.O. & Asbjørnslett, B.E., 2007. ICT in resilient global logistics. In *Proceedings of ESREL 2007*. pp. 2349–2356.

Grøtan, T.O., Størseth, F., Rø, M.H. & Skjerve, A.B., 2008. Resilience, Adaptation and Improvisation – increasing resilience by organising for successful improvisation. *3rd Symposium on Resiience Engineering*, pp.1–7.

Gunasekaran, A., Rai, B. & Griffin, M., 2011. Resilience and competitiveness of small and medium size enterprises: An empirical research. *International Journal of Production Research*, 49(18), pp.5489–5509.

Hamel, G. & Valikangas, L., 2003. The quest for resilience. *Harvard business review*, 81(9), pp.52–65.

Hammond, G.D. & Bier, V.M., 2015. Alternative evacuation strategies for nuclear power accidents. *Reliability Engineering and System Safety*, 135, pp.9–14.

Hardy, T.L., 2014. Resilience: A holistic safety approach. In *Proceedings - Annual Reliability and Maintainability Symposium*. pp. 1–6.

Hearnshaw, E.J.S. & Wilson, M.M.J., 2013. A complex network approach to supply chain network theory Edward. *International Journal of Operations & Production Management*, 33(4), pp.442–469.

Heinicke, M., 2014. Implementation of resilient production systems by production control. In *Procedia CIRP*. pp. 105–110.

Henry, D. & Ramirez-Marquez, J.E., 2010. A generic quantitative approach to resilience: A proposal. In *20th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2010*. pp. 291–301.

Hilton, J., Wright, C. & Kiparoglou, V., 2012. Building resilience into systems. In *SysCon 2012 - 2012 IEEE International Systems Conference, Proceedings*. pp. 638–645.

Holling, C.S., 1973. Resilience and Stability of Ecological Systems. *Annual review of ecology and systematics*, pp.1–23.

Hollnagel, E., 2010. How Resilient Is Your Organisation? An Introduction to the Resilience Analysis Grid (RAG). *Sustainable Transformation: Building a Resilient Organization*.

Horne III, J.F., 1997. The coming age of organizational resilience. *Business Forum*, 22(2/3), pp.24–29.

Horne III, J.F. & Orr, J.E., 1998. Assessing behaviors that create resilient organizations. *Employment Relations Today*, 24(4), pp.29–39.

Hosseini, S., Barker, K. & Ramirez-Marquez, J.E., 2016. A review of definitions and measures of system resilience. *Reliability Engineering & System Safety*, 145, pp.47–61.

Hu, Y., Li, J. & Holloway, L.E., 2009. A modeling and aggregation approach for analyzing resilience of manufacturing enterprises. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. pp. 692–697.

Hu, Y., Li, J. & Holloway, L.E., 2010. Achieving resilience for a class of serial production networks. In *Proceedings of the 2010 American Control Conference, ACC 2010*. pp. 5326–5331.

Hu, Y., Li, J. & Holloway, L.E., 2008. Towards modeling of resilience dynamics in manufacturing enterprises: Literature review and problem formulation. In *4th IEEE Conference on Automation Science and Engineering, CASE 2008*. pp. 279–284.

Ibrahim, G.M., Rutka, J.T. & Snead, O.C., 2013. Network analysis reveals patterns of antiepileptic drug use in children with medically intractable epilepsy. *Epilepsy & behavior : E&B*, 28(1), pp.22–5.

Ihaka, R. & Gentleman, R., 1996. R : A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3), pp.299–314.

Ismail, H.S., Poolton, J. & Sharifi, H., 2011. The role of agile strategic capabilities in achieving resilience in manufacturing-based small companies. *International Journal of Production Research*, 49(18), pp.5469–5487.

Jaaron, A. & Backhouse, C., 2014. Service organisations resilience through the application of the vanguard method of systems thinking: A case study approach. *International Journal of Production Research*, 52(7), pp.2026–2041.

Jackson, D., Firtko, A. & Edenborough, M., 2007. Personal resilience as a strategy for surviving and thriving in the face of workplace adversity: a literature review. *Journal of advanced nursing*, 60(1), pp.1–9.

Jackson, S., 2007. A multidisciplinary framework for resilence to disasters and disruptions. *Journal of Integrated Design and Process Science*, 11(2), pp.91–108.

Janssen, M.A., Bodin, O., Anderies, J.M., Elmqvist, T., Ernstson, H., McAllister, R.R.J., Olson, P. & Ryan, P., 2006. Toward a Network Perspective of the Study of Resilience in Social-Ecological Systems. *Ecology and Society*, 11(1), p.15.

Jennings, N.R., Sycara, K. & Wooldridge, M., 1998. A Roadmap of Agent Research and Development. *Autonomous agents and multi-agent systems*, 38, pp.7–38.

Jiang, Y. & Jiang, J.C., 2015. Diffusion in Social Networks: A Multiagent Perspective. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2), pp.198–213.

Johnsen, S. & Veen, M., 2012. Risk assessment and improvement of resilience of critical communication infrastructure. In *Advances in Safety, Reliability and Risk Management - Proceedings of the European Safety and Reliability Conference, ESREL 2011*. pp. 2739–2747.

Kachali, H., Stevenson, J.R., Whitman, Z.R., Seville, E., Vargo, J. & Wilson, T.M., 2012. Organisational resilience and recovery for Canterbury organisations after the 4 September 2010 earthquake. *Australasian Journal of Disaster and Trauma Studies*, 1(1), pp.11–19.

Kamalahmadi, M. & Parast, M.M., 2016. A review of the literature on the principles of enterprise and supply chain resilience: Major findings and directions for future research. *International Journal of Production Economics*, 171, pp.116–133.

Kao, T.C. & Porter, M.A., 2017. Layer Communities in Multiplex Networks. *Journal of Statistical Physics*, pp.1–17.

Karagiannis, G.-M., Piatyszek, E. & Flaus, J.-M., 2010. Industrial emergency planning modeling: a first step toward a robustness analysis tool. *Journal of hazardous materials*, 181(1–3), pp.324–34.

Kendra, J. & Wachtendorf, T., 2002. Elements of Community Resilience in the World Trade Center Attack,

Kendra, J. & Wachtendorf, T., 2003. Elements of resilience after the World Trade Center disaster: reconstituting New York City's Emergency Operations Centre. *Disasters*, 27(1), pp.37–53.

Khakzad, N. & Reniers, G., 2015. Using graph theory to analyze the vulnerability of process plants in the context of cascading effects. *Reliability Engineering and System Safety*, 143, pp.63–73.

Khelil, A., Becker, C., Tian, J. & Rothermel, K., 2002. An epidemic model for information diffusion in MANETs. Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems - MSWiM '02, p.54.

Kohno, Y., Masuda, Y., Nagahashi, H., Tanaka, K. & Tashiro, K., 2012. Form development for self-rating an organization's vulnerability and resilience to disruption. *Journal of Disaster Research*, 7(4), pp.392–407.

Lalonde, C., 2007. Crisis Management and Organizational Development : Towards the Conception of a Learning Model in Crisis Management. *Organization Development Journal*, 25(1), pp.507–517.

Lambertini, L. & Marattin, L., 2016. To adjust or not to adjust after a cost-push shock? A simple duopoly model with (and without) resilience. *Economics of Innovation and New Technology*, 25(2), pp.172–181.

Langlois, L., 2013. IAEA Action Plan on nuclear safety. *Energy Strategy Reviews*, 1(4), pp.302–306.

Lee, A. V, Vargo, J. & Seville, E., 2013. Developing a Tool to Measure and Compare Organizations ' Resilience. *Natural Hazards Review*, 14, pp.29–41.

Lengnick-Hall, C.A. & Beck, T.E., 2005. Adaptive fit versus robust transformation: How organizations respond to environmental change. *Journal of Management*, 31(5), pp.738–757.

Lengnick-Hall, C.A., Beck, T.E. & Lengnick-Hall, M.L., 2011. Developing a capacity for organizational resilience through strategic human resource management. *Human Resource Management Review*, 21(3), pp.243–255.

Linnenluecke, M.K., 2017. Resilience in Business and Management Research: A Review of Influential Publications and a Research Agenda. *International Journal of Management Reviews*, 19(1), pp.4–30.

Linnenluecke, M.K. & Griffiths, A., 2010. Beyond adaptation: Resilience for business in light of climate change and weather extremes. *Business and Society*, 49(3), pp.477–511.

Linnenluecke, M.K., Griffiths, A. & Winn, M., 2012. Extreme weather events and the critical importance of anticipatory adaptation and organizational resilience in responding to impacts. *Business Strategy and the Environment*, 21(1), pp.17–32.

Liu, N., An, H., Gao, X., Li, H. & Hao, X., 2016. Breaking news dissemination in the media via propagation behavior based on complex network theory. *Physica A: Statistical Mechanics and its Applications*, 453, pp.44–54.

Longstaff, P.H. & Yang, S., 2008. Communication Management and Trust : Their Role in Building Resilience to " Surprises " Such As Natural Disasters , Pandemic Flu , and Terrorism. *Ecology And Society*, 13(1), p.3.

Lopez-Paredes, A., 2001. Análisis e Ingeniería de las Instituciones Económicas. Una metodología basada en agentes. Universidad del País Vasco (Spain).

Lopez-Paredes, A., Hernández-Iglesias, C. & Gutiérrez, J.P., 2002. Towards a new experimental socio-economics. *The Journal of Socio-Economics*, 31(4), pp.423–429.

Lopez-Paredes, A. & Hernández, C., 2008. *Agent Based Modelling in Natural Resource Management*, INSISOC, Social Systems Engineering Centre.

Louisot, J.-P., 2015. Risk and/or Resilience Management. *RISK GOVERNANCE & CONTROL: Financial markets and institutions*, 5(2), pp.84–91.

Lv, Y., Huang, G.H.H., Guo, L., Li, Y.P.P., Dai, C., Wang, X.W.W. & Sun, W., 2013. A scenario-based modeling approach for emergency evacuation management and risk analysis under multiple uncertainties. *Journal of hazardous materials*, 246–247, pp.234–44.

MacKenzie, C., Santos, J.R. & Barker, K., 2012. Measuring changes in international production from a disruption: Case study of the Japanese earthquake and tsunami. *International Journal of Production Economics*, 138(2), pp.293–302.

Macuzić, I., Tadić, D., Aleksic, A. & Stefanović, M., 2016. A two step fuzzy model for the assessment and ranking of organizational resilience factors in the process industry. *Journal of Loss Prevention in the Process Industries*, 40, pp.122–130.

Mafabi, S., Munene, J. & Ahiauzu, A., 2015. Creative climate and organisational resilience: the mediating role of innovation. *International Journal of Organizational Analysis*, 23(4), pp.564–587.

De Maio, C., Fenza, G., Gaeta, M., Loia, V. & Orciuoli, F., 2011. A knowledge-based framework for emergency DSS. *Knowledge-Based Systems*, 24, pp.1372–1379.

Mallak, L.A., 1997. How to build a resilient organization. In *Proceedings of the Industrial Engineering Solutions 1997 Conference*. Miami, pp. 170–177.

Mallak, L.A., 1998. Measuring resilience in health care provider organizations. *Health manpower management*, 24(4), pp.148–152.

Mamouni Limnios, E., 2011. Resilient organizations : Offense versus Defense. In *25th Annual ANZAM Conference*. pp. 7–9.

Mamouni Limnios, E., Mazzarol, T., Ghadouani, A. & Schilizzi, S.G., 2014. The resilience architecture framework: Four organizational archetypes. *European Management Journal*, 32(1), pp.104–116.

Manyena, S.B., 2006. The Concept of Resilience Revisited. *Disasters*, 30(4), pp.433–450.

Markman, G. & Venzin, M., 2014. Resilience: Lessons from banks that have braved the economic crisis-And from those that have not. *International Business Review*, 23(6), pp.1096–1107.

McManus, S., Seville, E., Vargo, J. & Brunsdon, D., 2008. Facilitated Process for Improving Organizational Resilience. *Natural Hazards Review*, 9(2), pp.81–90.

McManus, S., Seville, E., Vargo, J. & Brunsdon, D., 2007. Resilience Management: A Framework for Assessing and Improving the Resilience of Organisations. *Resilient Organisations Research Report 2007/01*.

Megele, C., 2014. Resilient organizations turning challenges into opportunities: HR occupies a central place in preparing companies for change. *Human Resource Management International Digest*, 22(5), pp.1–4.

Mendonça, D., Beroggi, G.E.G., van Gent, D. & Wallace, W., 2006. Designing gaming simulations for the assessment of group decision support systems in emergency response. *Safety Science*, 44(6), pp.523–535.

Mendonça, D. & Wallace, W.A., 2015. Factors underlying organizational resilience: The case of electric power restoration in New York City after 11 September 2001. *Reliability Engineering & System Safety*, 141(0), pp.83–91.

Microsoft, 2015. Power BI. Available at: https://powerbi.microsoft.com/es-es/.

Milanzi, D. & Weeks, R., 2014. Understanding servitization: A resilience perspective. In PICMET 2014 - Portland International Center for Management of Engineering and Technology, Proceedings: Infrastructure and Service Integration. pp. 2332–2342.

Negeri, E., Kuipers, F. & Baken, N., 2015. Designing reliable and resilient smart low-voltage grids. *International Journal of Critical Infrastructure Protection*, 9, pp.24–37.

Newman, M., 2003. The structure and function of complex networks. *SIAM review*, 45(2), pp.167–256.

Newman, M., Barabasi, A.-L. & Watts, D.J., 2006. *The Structure and Dynamics of Networks*, Princeton University Press.

Nikolai, C. & Madey, G., 2009. Tools of the Trade : A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 12(22).

Nooy, W. De, Mrvar, A. & Batagelj, V., 2005. *Exploratory social network analysis with Pajek*, Cambridge University Press.

North, M.J., Collier, N.T. & Vos, J.R., 2006. Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. , 16(1), pp.1–25.

Omoto, A., 2013. The accident at TEPCO's Fukushima-Daiichi Nuclear Power Station: What went wrong and what lessons are universal? *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 731, pp.3–7.

Orchiston, C., Prayag, G. & Brown, C., 2016. Organizational resilience in the tourism sector. *Annals of Tourism Research*, 56, pp.145–148.

Ortiz-de-Mandojana, N. & Bansal, P., 2015. The long-term benefits of organizational resilience through sustainable business practices. *Strategic Management Journal*.

Ouyang, M., 2014. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety*, 121, pp.43–60.

Pal, R., Torstensson, H. & Mattila, H., 2014. Antecedents of organizational resilience in economic crises - An empirical study of Swedish textile and clothing SMEs. *International Journal of Production Economics*, 147(PART B), pp.410–428.

Park, J. & Jung, W., 2007. A study on the development of a task complexity measure for emergency operating procedures of nuclear power plants. *Reliability Engineering & System Safety*, 92(8), pp.1102–1116.

Pavón, J., Arroyo, M., Hassan, S. & Sansores, C., 2008. Agent-based modelling and simulation for the analysis of social patterns. *Pattern Recognition Letters*, 29(8), pp.1039–1048.

Peng, H., Lu, S., Zhao, D., Zhang, A. & Li, J., 2012. An anti-attack model based on complex network theory in P2P networks. *Physica A: Statistical Mechanics and its Applications*, 391(8), pp.2788–2793.

Pérez Ríos, J., 2012. Design and Diagnosis for Sustainable Organizations: The Viable System Method, Springer Heidelberg, New York, Dordrecht, London.

Pérez Ríos, J., 2010. Models of organizational cybernetics for diagnosis and design. *Kybernetes*, 39(9/10), pp.1529–1550.

Poland, G.A., Kennedy, R.B., McKinney, B.A., Ovsyannikova, I.G., Lambert, N.D., Jacobson, R.M. & Oberg, A.L., 2013. Vaccinomics, adversomics, and the immune response network theory: Individualized vaccinology in the 21st century. *Seminars in Immunology*, 25(2), pp.89–103.

Posada, M. & Lopez-Paredes, A., 2008. How to Choose the Bidding Strategy in Continuous Double Auctions: Imitation Versus Take-The-Best Heuristics. *Journal of Artificial Societies and Social Simulation*, 11(1), p.6.

Powley, E.H., 2009. Reclaiming resilience and safety: Resilience activation in the critical period of crisis. *Human Relations*, 62(9), pp.1289–1326.

Poza, D., Santos, J.I., Galan, J.M. & Lopez-Paredes, A., 2011. Mesoscopic effects in an agent-based bargaining model in regular lattices. *PLoS ONE*, 6(3), p.e17661.

Proper, J.W. & Pienaar, W.J., 2011. Resilience as an imperative in public transport organisations. *Corporate Ownership and Control*, 8(4 D), pp.373–388.

Quesnel, G., Duboz, R. & Ramat, É., 2009. The Virtual Laboratory Environment - An operational framework for multi-modelling, simulation and analysis of complex dynamical systems. *Simulation Modelling Practice and Theory*, 17(4), pp.641–653.

Raj, A., Kuceyeski, A. & Weiner, M., 2012. A Network Diffusion Model of Disease Progression in Dementia. *Neuron*, 73(6), pp.1204–1215.

Read, D., 2005. Some Observations on Resilience and Robustness in Human Systems. *Cybernetics & Systems*, 36, pp.773–802.

Reinmoeller, P. & Van Baardwijk, N., 2005. The link between diversity and resilience. *MIT Sloan Management Review*, 46(4), p.61.

Rerup, C., 2001. "Houston, we have a problem": Anticipation and improvisation as sources of organizational resilience. *Comportamiento Organizacional y Gestão*, 7(1), pp.27–44.

Rigaud, E., Neveu, C., Duvenci-Langa, S., Obrist, M. & Rigaud, S., 2013. Proposition of an organisational resilience assessment framework dedicated to railway traffic management. In *Rail Human Factors: Supporting Reliability, Safety and Cost Reduction*. pp. 727–732.

Righi, A.W., Saurin, T.A. & Wachs, P., 2015. A systematic literature review of resilience engineering: Research areas and a research agenda proposal. *Reliability Engineering & System Safety*, 141, pp.142–152.

Riolli, L. & Savicki, V., 2003. Information system organizational resilience. *Omega*, 31(1), pp.227–233.

Robb, D., 2000. Building Resilient Organizations. *OD PRACTITIONER*, 32(3), pp.27–32.

Rocco S., C.M. & Ramirez-Marquez, J.E., 2011. Vulnerability metrics and analysis for communities in complex networks. *Reliability Engineering and System Safety*, 96(10), pp.1360–1366.

Ruiz-Martin, C., 2013. Modelo Organizational para la Gestión de Emergencias. Universidad de Valladolid.

Ruiz-Martin, C., Bouanan, Y., Wainer, G., Zacharewicz, G. & Lopez-Paredes, A., 2016. A hybrid approach to study communication in emergency plans. In T. M. K. Roeder et al., eds. *Proceedings of the 2016 Winter Simulation Conference*. Arlington, Virginia, USA, pp. 1376–1387.

Ruiz-Martin, C., Lopez-Paredes, A. & Wainer, G., 2015. Applying complex network theory to the assessment of organizational resilience. *IFAC-PapersOnLine*, 48(3), pp.1224–1229.

Ruiz-Martin, C., Lopez-Paredes, A. & Wainer, G., 2018. What we know and do not know about organizational resilience. *International Journal of Production and Management Engineering*, InPress.

Ruiz-Martin, C. & Poza, D., 2015. Project configuration by means of network theory. *International Journal of Project Management*, 33(8), pp.1755–1767.

Ruiz-Martin, C., Ramírez Ferrero, M., Gonzalez-Alvarez, J.L. & Lopez-Paredes, A., 2015. Modelling of a Nuclear Emergency Plan: Communication Management. *Human and Ecological Risk Assessment: An International Journal*, 21(5), pp.1152–1168.

Saito, K., Kimura, M., Ohara, K. & Motoda, H., 2009. Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis. *Asian Conference on Machine Learning*, pp.322–337.

Sanchis, R. & Poler, R., 2013. Definition of a framework to support strategic decisions to improve Enterprise Resilience. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*. pp. 700–705.

Santos, J.I., Poza, D.J., Galán, J.M. & López-Paredes, A., 2012. Evolution of Equity Norms in Small-World Networks. *Discrete Dynamics in Nature and Society*, pp.1–18.

Sarjoughian, H.S. & Zeigler, B.P., 1998. DEVSJAVA : Basis for a DEVS-based Collaborative M & S Environment. In *Proceedings of SCS International Conference on Web-Based Modeling and Simulation*. San Diego.

Sbayou, M., Bouanan, Y., Zacharewicz, G., Ribault, J. & François, J., 2017. DEVS modelling and simulation for healthcare process application for hospital emergency department. *Simulation Series*, 49(1).

Schelling, T.C., 1971. Dynamic Models of Segregation. *Journal of Mathematical Sociology*, 1, pp.143–186.

Schwaninger, M. & Pérez Ríos, J., 2008. System dynamics and cybernetics: a synergetic pair. *System Dynamics Review*, 24(2), pp.145–174.

Seville, E., 2009. Resilience: Great Concept... But What Does it Mean for Organizations? *Tephra*, July, pp.9–14.

Sheffi, Y., 2007. Building a Resilient Organization. *The Bridge - National Academy of Engineering*, 37(1), pp.30–36.

Sheffi, Y. & Rice Jr., J.B., 2005. A supply chain view of the resilient enterprise. *MIT Sloan Management Review*, 47(1), p.41–48+94.

Simonovic, S.P. & Ahmad, S., 2005. Computer-based model for flood evacuation emergency planning. *Natural Hazards*, 34(1), pp.25–51.

Sole-Ribalta, A., Domenico, M. De, Gómez, S. & Arenas, A., 2014. Centrality Rankings in Multiplex Networks. In *In Proceedings of the 2014 ACM conference on Web science*. pp. 149–155.

Somers, S., 2009. Measuring resilience potential: An adaptive strategy for organizational crisis planning. *Journal of Contingencies and Crisis Management*, 17(1), pp.12–23.

Spanish Goverment, 2015. Cobertura de Banda Ancha en España en el Primer Trimestre de 2015. , p.87. Available at: http://www.minetur.gob.es/telecomunicaciones/banda-ancha/cobertura/Documents/cobertura-BA-1trimestre2015.pdf [Accessed October 17, 2016].

Starr, R., Newfrock, J. & Delurey, M., 2003. Enterprise Resilience: Managing Risk in the Networked Economy. *Strategy and Business*, (30), pp.70–79.

Stewart, J. & O'Donnell, M., 2007. Implementing change in a public agency. *International Journal of Public Sector Management*, 20(3), pp.239–251.

Strogatz, S.H., 2001. Exploring complex networks. *Nature*, 410(6825), pp.268–276.

Suddaby, R., 2010. Editor's comments: Construct clarity in theories of management and organization. *Academy of Management Review*, 35(3), pp.346–357.

Sutcliffe, K.M. & Vogus, T.J., 2003. Organizing for resilience. In K. Cameron, J. E. Dutton, & R. E. Quinn, eds. *Positive Organizational Scholarship*. San Francisco, pp. 94–110.

Tadić, D., Aleksic, A., Stefanović, M. & Arsovski, S., 2014. Evaluation and Ranking of Organizational Resilience Factors by Using a Two-Step Fuzzy AHP and Fuzzy TOPSIS. *Mathematical Problems in Engineering*, pp.1–13.

Taleb, N.N., 2012. *Antifragile: Things that Gain from Disorder*, New York: Random House.

Taleb, N.N. & Douady, R., 2013. Mathematical definition, mapping, and detection of (anti)fragility. *Quantitative Finance*, 13(11), pp.1677–1689.

Thomas, A., Byard, P., Francis, M., Fisher, R. & White, G., 2016. Profiling the resiliency and sustainability of UK manufacturing companies. *Journal of Manufacturing Technology Management*, 27(1), pp.82–99.

Tierney, K.J., 2003. Conceptualizing and measuring organizational and community resilience: Lessons from the emergency response following the September 11, 2001 attack on the World Trade Center. *Disaster Research Center Preliminary Papers*.

Tillement, S., Cholez, C. & Reverdy, T., 2009. Assessing organizational resilience: an interactionist approach. *Management*, 12(4), pp.230–264.

Tompkins, J.A., 2007. 4 Steps to business resilience. *Industrial Management (Norcross, Georgia)*, 49(4), pp.14–18.

Van Trijp, J., Ulieru, M. & Van Gelder, P., 2012a. Quantitative approach of organizational resilience for a Dutch Emergency Response Safety Region. In *Advances in Safety, Reliability and Risk Management - Proceedings of the European Safety and Reliability Conference, ESREL 2011*. pp. 173–180.

Van Trijp, J., Ulieru, M. & Van Gelder, P., 2012b. Quantitative modeling of organizational resilience for Dutch emergency response safety regions. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 226(6), pp.666–676.

Tse, T., Couturier, J. & Roux, L., 2012. Resilience of a business: The case of Yellow Pages Directories. *International Journal of Management Practice*, 5(2), pp.149–169.

Tukamuhabwa, B.R., Stevenson, M., Busby, J. & Zorzini, M., 2015. Supply chain resilience: Definition, review and theoretical foundations for further study. *International Journal of Production Research*, 53(18), pp.5592–5623.

Turoff, M., Chumer, M., Van de Walle, B. & Yao, X., 2004. The design of a dynamic emergency response management information system (DERMIS). *Journal of Information Techonology Theory and Application*, 5(4), pp.1–35.

US Department of Homeland Security, Resilience. Available at: https://www.dhs.gov/topic/resilience [Accessed July 12, 2017].

Vangheluwe, H.L.M., 2000. DEVS as a common denominator for multi-formalism hybrid systems modelling. In *CACSD. Conference Proceedings. IEEE International Symposium on Computer-Aided Control System Design (Cat. No.00TH8537)*. IEEE, pp. 129–134.

Vicino, D., 2015. Improved Time Representation in Discrete-Event Simulation Modeling and Simulation. Université Nice Sophia Antipolis; Carleton University.

Vicino, D., Niyonkuru, D., Wainer, G. & Dalle, O., 2015. Sequential PDEVS Architecture. In *DEVS '15 Proceedings of the Symp on Theory of M&S: DEVS Integrative M&S Symposium*. pp. 165–172.

Vogus, T.J. & Sutcliffe, K.M., 2007. Organizational resilience: Towards a theory and research agenda. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp.3418–3422.

Wainer, G., 2002. CD++: a toolkit to develop DEVS models. *Software: Practice and Experience*, 32(13), pp.1261–1306.

Wainer, G., 2009. Discrete-Event Modeling and Simulation: A Practitioner's Approach, CRC Press.

Wang, W., Liu, Q.-H., Cai, S.-M., Tang, M., Braunstein, L.A. & Stanley, H.E., 2016. Suppressing disease spreading by using information diffusion on multiplex networks. *Scientific Reports*, 6(7600), p.29259.

Watanabe, C., Kishioka, M. & Nagamatsu, A., 2004. Resilience as a source of survival stategy for high-technology firms experiencing megacompetition. *Technovation*, 24(2), pp.139–152.

Weick, K.E., 1993. The collapse of sensemaking in organizations: The Mann Gulch disaster. *Administrative science quarterly*, pp.628–652.

Whitehorn, G., 2010. Building Organisational Resilience in the Public Sector. In *Comcover Insurance and Risk Management Conference*.

Whitman, Z., Kachali, H., Roger, D., Vargo, J. & Seville, E., 2013. Short-form version of the Benchmark Resilience Tool (BRT-53). *Measuring Business Excellence*, 17(3), pp.3–14.

Whitman, Z., Stevenson, J., Kachali, H., Seville, E., Vargo, J. & Wilson, T., 2014. Organisational resilience following the Darfield earthquake of 2010. *Disasters*, 38(1), pp.148–177.

Wicker, P., Filo, K. & Cuskelly, G., 2013. Organizational resilience of community sport clubs impacted by natural disasters. *Journal of Sport Management*, 27(6), pp.510–525.

Wilensky, U., 1999. NetLogo. *Center for Connected Learning and Computer-Based Modeling, Northwestern University*. Available at: http://ccl.northwestern.edu/netlogo/.

Winston, A., 2014. Resilience in a hotter world. *Harvard business review*, 92(4), p.56–64,132.

Woods, D.D., 2015. Four concepts for resilience and the implications for the future of resilience engineering. *Reliability Engineering and System Safety*, 141, pp.5–9.

Wright, C., Kiparoglou, V., Williams, M. & Hilton, J., 2012. A framework for resilience thinking. *Procedia Computer Science*, 8, pp.45–52.

Xiong, H., Puqing, W. & Bobashev, G. V, 2015. Multiple Peer Effects in the Diffusion of Innovations on Social Networks: A Simulation Study. *SSRN Electronic Journal*.

Yağan, O. & Gligor, V., 2012. Analysis of complex contagions in random multiplex networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 86(3), pp.1–11.

Yazdani, A., Otoo, R.A. & Jeffrey, P., 2011. Resilience enhancing expansion strategies for water distribution systems: A network theory approach. *Environmental Modelling and Software*, 26(12), pp.1574–1582.

Zeigler, B.P., Praehofer, H. & Kim, T.G., 2000. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems, Academic press.

Zhang, W.J. & Van Luttervelt, C.A., 2011. Toward a resilient manufacturing system. *CIRP Annals - Manufacturing Technology*, 60(1), pp.469–472.

Zhou, Q., Huang, W. & Zhang, Y., 2011. Identifying critical success factors in emergency management using a fuzzy DEMATEL method. *Safety Science*, 49(2), pp.243–252.

Zhu, L. & Luo, J., 2016. The Evolution Analysis of Guangzhou Subway Network by Complex Network Theory. *Procedia Engineering*, 137, pp.186–195.

Appendix A.Diffusion Abstract Model. Formal definition using DEVS

The DAM, presented in figure 2, is formally defined using DEVS as follows:

$$DAM = < X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC >$$

Where

$X = \emptyset$

$Y = \emptyset$

$$D = \left\{ \begin{array}{c} Node_1, Node_2, \dots, Node_n, \\ IndirectLink_1, IndirectLink_2, \dots, IndirectLink_n, \\ DirectLink, LinkConnectors, DiffusionElementGenerator, \\ NodeUpdater, IndirectLinkUpdater, DirectLinkUpdater, \\ LinkConnectorsUpdater \end{array} \right\}$$

$$M = \left\{ \begin{array}{c} M_{Node1}, M_{Node2}, \dots, M_{Noden} \\ M_{IndirectLink1}, M_{IndirectLink2}, \dots, M_{IndirectLinkn} \\ M_{DirectLink}, M_{LinkConnectors}, M_{DiffusionElementGenerator} \\ M_{NodeUpdater}, M_{IndirectLinkUpdater}, M_{DirectLinkUpdater}, \\ M_{LinkConnectorsUpdater} \end{array} \right\}$$

$EIC = \emptyset$

$EOC = \emptyset$

$$IC = \left\{ \begin{array}{c} \big((DiffusionElementGenerator, Out), (Node_1, InitialDiffusionElement_{In})\big), \\ \big((DiffusionElementGenerator, Out), (Node_2, InitialDiffusionElement_{In})\big), \\ \dots \\ \big((DiffusionElementGenerator, Out), (Node_n, InitialDiffusionElement_{In})\big), \\ \big((NodeUpdater, Out), (Node_1, PropertyUpdate_{In})\big), \\ \big((NodeUpdater, Out), (Node_2, PropertyUpdate_{In})\big), \\ \dots \\ \big((NodeUpdater, Out), (Node_n, PropertyUpdate_{In})\big), \\ \big((IndirectLinkUpdater, Out), (IndirectLink_1, PropertyUpdate_{In})\big), \\ \big((IndirectLinkUpdater, Out), (IndirectLink_2, PropertyUpdate_{In})\big), \\ \dots \\ \big((IndirectLinkUpdater, Out), (IndirectLink_2, PropertyUpdate_{In})\big), \\ \big((DirectLinkUpdater, Out), (DirectLink, PropertyUpdate)\big), \\ \big((LinkConnectorsUpdater, Out), (LinkConnectors, PropertyUpdate)\big), \\ \big((Node_1, PropertyUpdate_{Out}), (NodeUpdater, In)\big), \\ \big((Node_2, PropertyUpdate_{Out}), (NodeUpdater, In)\big), \\ \dots \\ \big((Node_n, PropertyUpdate_{Out}), (NodeUpdater, In)\big), \\ \big((Node_1, DiffusionElementDirect_{Out}), (DirectLink, DiffusionElement_{In})\big), \\ \big((Node_2, DiffusionElementDirect_{Out}), (DirectLink, DiffusionElement_{In})\big), \\ \dots \\ \big((Node_n, DiffusionElementDirect_{Out}), (DirectLink, DiffusionElement_{In})\big), \\ \big((Node_1, DiffusionElementIndirect_{Out}), (IndirectLink_1, NodeDiffusionElement_{In})\big), \\ \big((Node_2, DiffusionElementIndirect_{Out}), (IndirectLink_2, NodeDiffusionElement_{In})\big), \\ \dots \\ \big((Node_n, DiffusionElementIndirect_{Out}), (IndirectLink_n, NodeDiffusionElement_{In})\big), \\ \big((DirectLink, DiffusionElement_{Out}), (Node_1, DiffusionElementDirect_{In})\big), \\ \big((DirectLink, DiffusionElement_{Out}), (Node_2, DiffusionElementDirect_{In})\big), \\ \dots \\ \big((DirectLink, DiffusionElement_{Out}), (Node_n, DiffusionElementDirect_{In})\big), \\ \big((IndirectLink_1, NodeDiffusionElement_{Out}), (Node_1, DiffusionElementIndirect_{In})\big), \\ \big((IndirectLink_2, NodeDiffusionElement_{Out}), (Node_2, DiffusionElementIndirect_{In})\big), \\ \dots \\ \big((IndirectLink_n, NodeDiffusionElement_{Out}), (Node_n, DiffusionElementIndirect_{In})\big), \\ \big((IndirectLink_1, ConnectorDiffusionElement_{Out}), (LinkConnectors, DiffusionElement_{In})\big), \\ \big((IndirectLink_2, ConnectorDiffusionElement_{Out}), (LinkConnectors, DiffusionElement_{In})\big), \\ \dots \\ \big((IndirectLink_n, ConnectorDiffusionElement_{Out}), (LinkConnectors, DiffusionElement_{In})\big), \\ \big((LinkConnectors, DiffusionElement_{Out}), (IndirectLink_1, ConnectorDiffusionElement_{In})\big), \\ \big((LinkConnectors, DiffusionElement_{Out}), (IndirectLink_2, ConnectorDiffusionElement_{In})\big), \\ \dots \\ \big((LinkConnectors, DiffusionElement_{Out}), (IndirectLink_n, ConnectorDiffusionElement_{In})\big) \end{array} \right\}$$

$$n = \#nodes\ in\ the\ Network\ Model$$

The rest of the coupled models inside the DAM are defined following the same formalism and reasoning.

Appendix B. Generator Filter. Formal definition using DEVS

The formal definition of Generator Filter atomic model is as follows:

$$GeneratorFilter(Id) = < X, Y, S, ta, \delta_{ext}, \delta_{int}, \delta_{con}, \lambda >$$

Where

$X = \{(\text{"In"}, diffusionElement) \mid diffusionElement \in DiffusionElements\}$

$DiffusionElements \in \forall structure\ with\ a\ field\ called\ "destinatary"\ \}$

$$Y = \left\{ \begin{matrix} (\text{"}Out\text{"}, diffusionElement \cup \emptyset) \mid \\ diffusionElement \in (DiffusionElements | destinatary = Id) \end{matrix} \right\}$$

$$S = \left\{ \begin{matrix} messagesPassingFilter \mid (messagesPassingFiter = \emptyset \cup \\ messagesPassingFilter = \{diffusionElement \mid destinatary = Id\} \end{matrix} \right\}$$

$$ta(s) = \left\{ \begin{matrix} messagesPassingFiter = \emptyset \rightarrow \infty \\ messagesPassingFilter = \{diffusionElement \mid destinatary = Id\} \rightarrow 1ms \end{matrix} \right\}$$

$\delta_{ext}(S, e, X) = \{(X | destinatary = Id) \rightarrow messagesPassingFilter += X\}$

$\delta_{int}(S) = \{messagesPassingFilter = \emptyset\}$

$\delta_{con}(S, e, X) = \delta_{int}(S) + \delta_{ext}(S, e, X)$

$\lambda(S) = \{messagesPassingFilter\}$

Appendix C.Indirect Links. Formal definition using DEVS

The formal definition of Indirect Links coupled model is as follows:

$$Indirect\ Links = < X, Y, D, \{M_d | d \in D\}, EIC, EOC, IC >$$

Where

$$X = \left\{ \begin{matrix} (\text{"NodeIn"}, diffusionElement) U \\ (\text{"LinkConnectorsIn"}, diffusionElement) U \\ (\text{"UpdaterIn"}, stateUpdate) \end{matrix} \right\}$$

$diffusionElement \in DiffusionElements$

$stateUpdate \in StateUpdates$

$DiffusionElements \in \forall structure\ with\ field\ \text{"}Destinatary\text{"}$

StateUpdates $\in \forall structure\ with\ field\ \text{"}LinkType\text{"}$

$$Y = \left\{ \begin{matrix} (\text{"}NodeOut\text{"}, diffusionElement) U \\ (\text{"}LinkConnectorsOut\text{"}, diffusionElement) \end{matrix} \right\}$$

$$D = \left\{ \begin{matrix} FilterNode, FiterUpdater, FilterLinkConnectors, \\ LinkType_1, LinkType_2, \dots, LinkType_n, \\ LinkSink \end{matrix} \right\}$$

$$M = \left\{ \begin{matrix} M_{FilterNode}, M_{FiterUpdater}, \dots, M_{FilterLinkConnectors}, \\ M_{LinkType1}, M_{LinkType2}, \dots, M_{LinkTypen}, \\ M_{LinkSink} \end{matrix} \right\}$$

$$EIC = \left\{ \begin{matrix} \left((Self, Self_{NodeIn}), (FilterNode, FilterNode_{In})\right), \\ \left(\left(Self, Self_{UpdaterIn}\right), (FilterUpdater, FilterUpdater_{In})\right), \\ \left((Self, Self_{LinkConnectorsIn}), (FilterLinkConnectors, FilterLinkConnectors_{In})\right) \end{matrix} \right\}$$

$$EOC = \left\{ \begin{matrix} \left((LinkType1, LinkType1_{NodeOut}), (Self, Self_{NodeOut})\right), \\ \dots \\ \left((LinkTypen, LinkTypen_{NodeOut}), (Self, Self_{NodeOut})\right) \\ \left((LinkType1, LinkType1_{LinkConnectorsOut}), (Self, Self_{LinkConnectorsOut})\right), \\ \dots \\ \left((LinkTypen, LinkTypen_{LinkConnectorsOut}), (Self, Self_{LinkConnectorsOut})\right) \end{matrix} \right\}$$

$$IC = \left\{ \begin{array}{c} \big((FilterNode, FilterNode_{LT1Out}), (LinkType1, LinkType1_{NodeIn})\big), \\ \dots \\ \big((FilterNode, FilterNode_{LTnOut}), (LinkTypen, LinkTypen_{NodeIn})\big), \\ \big((FilterNode, FilterNode_{LTn+1Out}), (LinkSink, LinkSink_{NodeIn})\big), \\ \dots \\ \big((FilterNode, FilterNode_{LTmOut}), (LinkSink, LinkSink_{NodeIn})\big), \\ \Big((FilterUpdater, FilterUpdater_{LT1Out}), \big(LinkType1, LinkType1_{UpdaterIn}\big)\Big), \\ \dots \\ \Big((FilterUpdater, FilterUpdater_{LTnOut}), \big(LinkTypen, LinkTypen_{UpdaterIn}\big)\Big), \\ \Big((FilterUpdater, FilterUpdater_{LTn+1Out}), \big(LinkSink, LinkSink_{UpdaterIn}\big)\Big), \\ \dots \\ \Big((FilterUpdater, FilterUpdater_{LTmOut}), \big(LinkSink, LinkSink_{UpdaterIn}\big)\Big), \\ \big((FilterLinkConnectors, FilterLinkConnectors_{LT1Out}), (LinkType1, LinkType1_{ConnectorsIn})\big), \\ \dots \\ \big((FilterLinkConnectors, FilterLinkConnectors_{LTnOut}), (LinkTypen, LinkTypen_{ConnectorsIn})\big), \\ \big((FilterLinkConnectors, FilterLinkConnectors_{LTn+1Out}), (LinkSink, LinkSink_{ConnectorsIn})\big), \\ \dots \\ \big((FilterLinkConnectors, FilterLinkConnectors_{LTmOut}), (LinkSink, LinkSink_{ConnectorsIn})\big) \end{array} \right\}$$

$n = \#LinkTypes\ in\ Indirect\ Link$

$$m = Total\ \#LinkTypes\ in\ the\ model$$

Appendix D.Generator Filter. Implementation in CDBoost

```
1 //Declaration of the ports in the atomic
2 struct genertatorFilter_defs{
3   struct out : public out_port<DiffusionElement> {};
4   struct in : public in_port< DiffusionElement > {};
5 };
6 //Atomic model definition
7 template<typename TIME>
8 class genertatorFilter {
9   using defs= genertatorFilter_defs;
10   public:
11   using input_ports=tuple<typename defs::in>; //Input ports definition
12   using output_ports=tuple<typename defs::out>; //Output ports definition
13   string id; //Model parameter
14   struct state_type{ //Model state declaration
15     vector<DiffusionElement> messagesPassingFilter;
16   };
17   state_type state; //Model state definition
18   genertatorFilter (string Id) noexcept { //Constructor & state initialization
19     id=Id;
20     state. messagesPassingFilter.clear();
21   }
22   void internal_transition() { //Internal transition
23     state.messagesPassingFilter.clear();
24   }
25   //External transition
26   void external_transition(TIME e,typename make_message_bags<input_ports>::type mbs){
27     for (const auto &x : get_messages<typename defs::in>(mbs)){
28        if(x.destinatary == id) state.messagesPassingFilter.emplace_back(x);
29     }
30   }
31   //Confluence transition
32   void confluence_transition(TIME e,typename make_message_bags<input_ports>::type mbs){
33      internal_transition();
34      external_transition(TIME(), move(mbs));
35   }
36   typename make_message_bags<output_ports>::type output() const { //Output function
37     typename make_message_bags<output_ports>::type bags;
38     for (int i = 0; i < (state. messagesPassingFilter.size()); i++){
39       get_messages<typename defs::out>(bags).push_back(state.messagesPassingFilter[i]);
40     }
41     return bags;
42   }
43   //Time advance function
44   TIME time_advance() const {
45              return   (state.messagesPassingFilter.empty()   ?   numeric_limits<TIME>::infinity()   :
46                                                  TIME("00:00:00:001"));
47   }
48 }
```

The *Generator Filter* model filters the messages in the *in* port based on the model Id. When the massages pass the filter criteria, they are sent through the out port. We start by defining the input and output ports of the model (lines 1-5). Then, we implement the DEVS functions: the internal transition (lines 22-24), the external transition (lines 25-30), confluence (lines 31-35) the output (lines 36-42) and time advance functions (lines 43-47). To do so, we define the DEVS function for the filter. The

internal transition function clears the *msgPassingFilter* variable. The external transition function stores the messages received through the input port in msgPassingFilter variable if the field “to” of the message matches the Id of the model. The output function sends the messages stored in the msgPassingFilter variable through the output port. Finally, the time advance function passivates the model if there is nothing to send, and set the time advance in 1ms is there is something to send

Appendix E. Diffusion Abstract Model. Implementation in CDBoost

```
1 //Input ports
2 using iports_DAM = tuple<>;
3 //Output ports
4 using oports_DAM = tuple<>;
5 //Components
6 using submodels_DAM = models_tuple<
7     Node1, Node2, ... , Noden,
8     IndirectLink1, IndirectLink2, ...,IndirectLinkn,
9     DirectLink, LinkConnectors, DiffusionElementGenerator,
10    NodeUpdater, IndirectLinkUpdater, DirectLinkUpdater, LinkConnectorsUpdater
11  >;
12 //External Input Couplings
13 using eics_DAM = tuple< >;
14 //External Output Couplings
15 using eocs_DAM = tuple< >;
16 //Internal Couplings
17 using ics_DAM = tuple<
18   IC<DiffusionElementGenerator, DiffusionElementGenerator::Out, Node1,
19           Node1::InitialDiffusionElementIn>,
20   ...
21   IC<DiffusionElementGenerator, DiffusionElementGenerator::Out, Noden,
22           Noden::InitialDiffusionElementIn>,
23   IC<NodeUpdater, NodeUpdater::Out, Node1, Node1::PropertyUpdateIn>,
24     ...
25   IC<NodeUpdater, NodeUpdater::Out, Noden, Noden::PropertyUpdateIn>,
26   IC<IndirectLinkUpdater, IndirectLinkUpdater::Out, IndirectLink1,
27           IndirectLink1::PropertyUpdateIn>
28     ...
29   IC<IndirectLinkUpdater, IndirectLinkUpdater::Out, IndirectLinkn,
30           IndirectLinkn::PropertyUpdateIn>
31   IC<DirectLinkUpdater, DirectLinkUpdater::Out, DirectLink,
32           DirectLink::PropertyUpdateIn>
33   IC<LinkConnectorsUpdater, LinkConnectorsUpdater::Out, LinkConnectors,
34           LinkConnectors::PropertyUpdateIn>
35   IC<Node1, Node1::PropertyUpdateOut, NodeUpdater, NodeUpdater::In>,
36   ...
37   IC<Noden, Noden::PropertyUpdateOut, NodeUpdater, NodeUpdater::In>,
38   IC<Node1, Node1:: DiffusionElementDirectOut, DirectLink,
39           DirectLink::DiffusionElementIn>,
40   ...
41   IC<Noden, Noden:: DiffusionElementDirectOut, DirectLink,
42           DirectLink::DiffusionElementIn>,
43   IC<Node1, Node1:: DiffusionElementIndirectOut, IndirectLink1,
44           IndirectLink1::DiffusionElementIn>,
45   ...
46   IC<Noden, Noden:: DiffusionElementIndirectOut, IndirectLinkn,
47           IndirectLinkn::DiffusionElementIn>,
48   IC<DirectLink, DirectLink::DiffusionElementOut, Node1,
49             Node1::DiffusionElementDirectIn>,
50   ...
51   IC<DirectLink, DirectLink::DiffusionElementOut, Noden,
52           Noden::DiffusionElementDirectIn>,
53   IC<IndirectLink1, IndirectLink1::DiffusionElementOut, Node1,
54           Node1::DiffusionElementIndirectIn, >,
55   ...
56   IC<IndirectLinkn, IndirectLinkn::DiffusionElementOut, Noden,
57           Noden::DiffusionElementIndirectIn,>,
```

```
58   IC<IndirectLink1, IndirectLink1::ConnectorDiffusionElementOut,LinkConnectors,
59           LinkConnectors::DiffusionElementIn>
60   ...
61   IC<IndirectLinkn, IndirectLinkn::ConnectorDiffusionElementOut, LinkConnectors,
62           LinkConnectors::DiffusionElementIn>
60   IC<LinkConnectors, LinkConnectors::DiffusionElementOut, IndirectLink1,
63           IndirectLink1::ConnectorDiffusionElementIn>
64   ...
65   IC<LinkConnectors, LinkConnectors::DiffusionElementOut, IndirectLinkn,
66           IndirectLinkn::ConnectorDiffusionElementIn >
67   >;
68 //Coupled model
69 template<typename TIME>
70 struct DAM : public coupled_model<
71       TIME, iports_DAM, oports_DAM, submodels_DAM, eics_DAM, eocs_DAM, ics_DAM>{};
```

To implement the DAM, we translate every component in the formal definition (see Appendix A) to a specific notation that CDBoost understands. We use the services and notation explained in section 2. In lines 1-4, we define the input and output ports of the model as a tuple. For the DAM, they are an empty tuple. In lines 5-11, we define the subcomponents of the models using the keyword *models_tuple*. This tuple includes the name of all the components of the DAM (both atomics and coupled models) defined in M. In lines 12-15, we define the external input and output couplings (EIC & EOC) as tuples. In the DAM, they are an empty set. In lines 16-67, we define the internal couplings (IC). The IC is a tuple that includes the IC specified in the formal definition. Finally, we define the DAM as a coupled model (lines 66-71). The coupled model is defined as a tuple (i.e. coupled model) that contains all the elements previously implemented and a TIME type parameter.