Acknowledgments

First of all, I would like to thank my thesis advisor, Mr. Donat Roland , for almost eight months of advice and his guidance in software engineering and programming principles which made much of this work possible. In particular, I appreciate the opportunity that Mr. Donat provided to me as well as a wide latitude to explore my thesis topic. I would like to thank Professor Zio Enrico for his fundamental support and his insights into the progression and the results of this work. I would also like to thank Mr. Philippe Nonclercq, Head of the T-51 team at Industrial Risks Management department of EDF R&D, for his kindness and for the help he gave me in making important decisions such as the doctorate program at EDF. I also thank all collegues for their collaboration on my thesis and for making me feel so well placed at the department. Thanks to my parents, Dario and Marina, and to my two younger brothers, Alessio and Simone, for all of their love, suggestions and support throughout my life. Special thanks to my sweet girlfriend Vera, whose guidance, love, and support has made everything in these long months worthwhile.

Abstract

Probabilistic Risk Assessment (PRA) is an ongoing topic of research at the department of Industrial Risk Management (IRM) of Electricité de France (EDF) R&D. For many years EDF researchers support, in terms of methods and tools, engineering and production divisions of the French company to assess the safety and performance of the complex systems installed in their power production units. To accomplish its mission, the IRM department has developed tools, collected in a unique software called KB3 (KnowledgeBase version 3), covering the whole process from the system description to their quantification. These tools are based on the modeling language FIGARO, also developed at IRM department. It is a declarative object oriented language used to describe the behavior of a system by specifying its rules of operation. Figaro platform provides an environment suited to the representation of static systems as well as the dynamic systems based on discrete events. In recent years, the IRM department is heavily used by internal clients to achieve dynamic probabilistic safety studies relating to electrical and hydraulic systems which have a role in the power generation units of EDF. These applications are currently processed using the generic formalism BDMP (Boolean Driven Markov Processes). It is a generic modeling formalism based on principles similar to those of fault trees with more the possibility to represent some dynamic dependences between system components (passive redundancy, reconfiguration). However, when the system complexity increases, this formalism suffers from some limitations, partly due to its generic character : difficult to maintain and regain BDMP existing model; difficult to take account of common cause

failures (CCF) of order greater than or equal to three, restriction to the hypothesis of constant failure rate. This work aims to contribute to the development of a new knowledge base KB3 devoted specifically to the dynamic rappresentation of the sources of the electrical system of the EPR. Ultimately it is a tool that allows to carry out dynamic PSA studies from a modeling faithful to that of the real system of the plant considered. The studies carried out in this work merely serve to support the EDF engineering division CNEN (Centre National d'équipement nucléaire) engaged in the realization of the EPR (UK EPR project) whose construction on the site of Hinkley Point C is going to be approved by the Office for Nuclear Regulation (ONR), the regulator for the civil nuclear industry in the United Kingdom.

Sommario

Lo sviluppo di metodi dinamici di valutazione probabilistica della sicurezza (PSA) è un tema costante di ricerca presso il dipartimento di Industrial Risk Management (IRM) di Electricité de France (EDF) R&D. Per molti anni i ricercatori di EDF sostongono, in termini di metodi e strumenti, le divisioni di ingegneria e produzione della società francese per valutare la sicurezza e le prestazioni dei complessi sistemi insallati nelle loro unità di potenza. Per compiere la sua missione, il dipartimento IRM ha sviluppato strumenti, raccolti in un unico programma chiamato KB3 (KnowledgeBase v.3), che coprono l'intero processo, dalla descrizione del sistema alla loro quantificazione. Questi strumenti si basano sul linguaggio di modellazione FIGARO, anche questo sviluppato presso il dipartimento IRM. È un linguaggio dichiarativo orientato agli oggetti utilizzato per descrivere il comportamento di un sistema specificando le modalità di funzionamento. La piattaforma Figaro fornisce un ambiente adatto alla rappresentazione di sistemi statici così come di sistemi dinamici basati su eventi discreti. Negli ultimi anni, il dipartimento IRM é stato ingaggiato da clienti interni alla società francese per ottenere studi dinamici probabilistici di sicurezza in materia di impianti elettrici e idraulici che hanno un ruolo nelle unità di generazione di potenza di EDF. Queste applicazioni sono attualmente trattate con il generico formalismo BDMP (Boolean Logic Driven Markov Processes). È un formalismo di modellazione basata su principi simili a quelle degli alberi di guasto (fault trees) con in più la possibilità di rappresentare alcuni dipendenze dinamiche tra i componenti del sistema (ridondanza passiva, riconfigurazione). Tuttavia, quando il

sistema aumenta la propria complessità, questo formalismo soffre di alcune limitazioni, in parte a causa del suo carattere generico: difficoltà nel mantenere e recuperare il modello pre-esistente BDMP , difficoltà nel tenere conto dei guasti di causa comune (CCF) di ordine superiore o uguale a tre, limitazione alle ipotesi di tasso di guasto costante. Questo lavoro allora si propone di contribuire allo sviluppo di una nuova base di conoscenze KB3 specificamente dedicata alla rappresentazione dinamica delle sorgenti del sistema elettrico del reattore EPR. In definitiva, si è cercato di sviluppare uno strumento che consente di effettuare studi dinamici PSA a partire da una modellazione fedele a quella del sistema reale dell'impianto considerato. Gli studi condotti in questo lavoro non fanno che supportare la divisione d'ingegneria di EDF CNEN (Centre Nationale d'Équipement Nucléaire) , impegnata nella realizzazione del nuovo EPR (progetto UK EPR) la cui costruzione sul sito di Hinkley Point C sta per essere approvata da parte dell' Office for Nuclear Regulation (ONR), l'ente regolatore per l'industria nucleare civile nel Regno Unito.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Description

Since late 1970s, Probabilistic Safety Assessment (PSA) has been commonly used to quantify the risk associated with the operation of nuclear power plants. In the late 1980s, the NUREG-1150 study was commissioned by the U.S. Nuclear Regulatory Commission (NRC) to examine the risk of five U.S. nuclear power plants utilizing the best available PSA methods of the day. The analysis presented in NUREG-1150 represented a major contribution to the state of the art in PSA with regards to nuclear plant systems and phenomenology. However, the methodology used suffers from certain drawbacks :

- since the timing of events is not explicitly modeled, the competition between risk significant and non-risk significant outcomes cannot be explicitly represented;
- the ordering of events is preset by the analyst although this order may change if uncertainty in system and phenomenological modeling is considered;
- the modeling of complex severe accident phenomenology is driven by expert- judgment and not always treated in a phenomenologically consistent manner.

For these reasons, there is currently an increasing interest in dynamic PSA methodologies since they can be used to address deficiencies of conventional methods listed above. In this context, EDF R&D has worked for many years in order to develop its own methods and tools to improve the PSA approaches. This work is an example of such attempts.

The goal is the development of a new modeling tool specifically devoted to represent the dynamic behavior of electrical systems. Ultimately, this tool aims to carry out dynamic PSA studies from a modeling faithful to that of the real system of the plant considered. This modeling tool has in fact, been to support the EDF engineering division CNEN (Centre National d'Équipement Nucléaire) to carry out specific safety demonstration studies to answer questions from both the British and French Safety Agency.

1.2 Objective

The purpose of this work is to report on the development and results of applying a dynamic modeling method based on the so-called smart components technique. Specifically, there are four distinct objectives that have been attempted to pursue:

- to develop a knowledge base in which the dynamic behavior of generic electrical components has been modeled;
- to perform a PSA study focusing on the electrical sources of the French EPR, distinguishing the Flamanville 3 and Hinkley Point C cases;
- to estimate by Monte-Carlo simulation the probability of occurrence of some indicators referring to the Loss of Off-site Power initiating event;
- to analyse the main accident event sequences leading to top events considered in the study.

1.3 EDF

1.3.1 EDF R&D

The EDF R&D is structured around seven sites, three located in France, near Paris. Completed on the site of Chatou, after the Second War World, the first infrastructure of

EDF R&D hosted the National Hydraulics Laboratory (NHL) whose works has enabled the development of hydropower in a context of high energy shortages. With the evolution of the means of production and needs, research on thermal, nuclear, networks, environment and numerical simulation, two others centers areas, Clamart and Renardières , were added to complete the area of expertise in R&D. In a context of profound change, the ambition of the R&D comes as three major areas :

- Consolidate a carbon-energy mix
- Anticipate the electrical system of the future taking into account the new a challenges of decentralized generation and smart grids
- Develop a flexible energy demand and low carbon.

1.3.2 MRI : Industrial Risks Management

The focus of study of the Industrial Risks Management Department (*MRI, Management des Risques Industriels*) , where I carried out this work during an 8-month long internship, is an hazard-prone socio-technical system operating within the EDF Group, such as nuclear and thermal power plants, hydraulic facilities and the power transmission network. This study includes various dimensions :

- the component,
- the technical system,
- the human and organizational factors,
- environment (natural, technological, the organizational, regulatory, etc.).

In close contact with EDF's operational units , the MRI Department develops models and tools which help to improve the control of risks with respect to safety, performance (availability, cost, etc.) and life cycle. The Department also provides support to other R&D

departments through its skills in operating safety, statistics and uncertainty propagation in particular. Its activities also spread to the EDF Group, since the Department has a long-standing history of collaboration with EDF Energy (UK). The MRI Department develops six key skills which enable it to cover all industrial risk issues:

- Probabilistic Safety Analyses,
- Systems Risks Analyses,
- Human and Organizational Factors,
- Probabilistic and statistic Approaches of Physical Phenomena,
- Decision Support & Performance of Assets,
- Modeling & Numerical Simulation of Processes, Robotics and M Experimental Approaches.

1.3.3 CNEN — Centre National d'Équipement Nucléaire

The CNEN is one of the six entities of the EDF Nuclear Engineering Division (DIN, *Division Ingénierie Nucléaire*) that is especially responsible for the design and implementation of EPR (European Pressurized Reactor, third generation plus). It provides detailed design of the nuclear island and control system. It supervised nuclear projects in France (Flamanville 3 and Penly 3) and abroad (United Kingdom, China, United States, etc.).. Dedicated teams are present in these countries. Moreover, the CNEN ensures the development and monitoring of engineering tools, especially with regard to computer-aided design.

1.4 Scope

This study is part of an EDF R&D project called DOPAMINE, closely linked to the PSA activities at the EDF Nuclear Engineering and Production Division . The project goal is

to respond to issues raised in DIN units, by expanding the scope of EPS applications. On the one hand, the project provides solutions to industrialize PSA methods and secondly, it investigates the possibility of developing new tools to meet the more relevant needs of DIN units . Among these emerging needs, there is an always more growing demand for dynamic PSA studies. In particular, following a request from the Office for Nuclear Regulation (ONR), the regulator for the civil nuclear industry in the United Kingdom, the CNEN asks for advanced PSA analyses on long-term scenarios of Loss of Off-site Power for the EPR in the context of the UK EPR project in order to highlight some of the modeling conservatisms associated to the classical EPR PSA study. It's that the goal we attempted to pursue in this work.

1.5 Dissertation overview

Chapter 2 of this work provides a background of the PSA and historical context for the specific case of application. Chapter 3 will also give an overview of the Hinckley Point C PSA with most regarding for the initiating events under consideration. Chapters 4-6 focuses on the tools used in the dynamic analyses and the modeling of the system under consideration. Chapter 4 describes the modeling tool used for the dynamic PSA analysis. Chapter 5 describes the structure and the key elements of the developed modeling tools. Chapter 6 gives an overview of the quantification techniques applied to compute indicators in the dynamic PSA models. Chapter 7 focuses on the specific case of study, or on the real object of interest, the EPR electrical sources, presents a discussion on the results that came out of the dynamic analysis and a comparison between the two cases of FLA and HPC. Finally, chapter 8 will discuss the conclusions and make suggestions about future work.

2 Probabilistic Safety Assessment

2.1 Hystorical Remarks

This chapter gives a background on the development of both conventional and dynamic Probabilistic Safety Assessment (PSA) methods. Section 2.2 discusses the development of conventional PSA. Subsections 2.2.1, 2.2.2 et 2.2.3 give a definition on the three levels of the classical PSA analysis. Subsection 2.2.4provides a basic overview of conventional PSA methodology as well as examples of different methods. Section 2.3 presents a brief background on dynamic PSA, and subsection 2.3.1 discusses dynamic PSA approach in a general way. Subsection 2.3.2 presents the more suitable methodology used today as reference for dynamic PSA. Subsection 2.3.3 provides an example of a dynamic method developed at EDF R&Dcalled Boolean Logic Driven Markov Process (BDMP). Finally, Subsection 2.3.4 introduces the concept of "smart components" which is the modeling approachadopted in this work.

2.2 Conventional PSA

Prior to the release in 1975 of the Reactor Safety Study (RSS)WASH1400(10) [7], the methods of licensing power plants for construction and operation were solely based on a deterministic approach. In a deterministic approach, heavy conservatisms are integrated in plant design in order to make up take into for the uncertainties in phenomenology and operation. The reliance on conservatism stems from the fact thatthere was limited

modeling capability for nuclear reactors analysis at the time. . The RSS introduced Probabilistic Safety Assessment (PSA) to the U.S. nuclear power industry and provided a detailed safety assessment of two plants : a pressurized water reactor (PWR) and a boiling water reactor (BWR). The RSS represented a large leap forward with regards to safety assessment in the nuclear industry. However, it was limited in its treatment of severe accident behavior, human reliability, common cause failure, external events, and uncertainty analysis *[20]*. In the late 1980s the NRC commissioned a study of five U.S. nuclear reactors. This study culminated in the NRC report NUREG1150 [20] which has set the standard in nuclear power plant risk assessment for almost two decades. As a result of the NUREG1150 analyses, the NRC required that all U.S. nuclear power plants submitted Individual Plant Examinations (IPEs). IPEs consist of a comprehensive study of plant systems to determine estimates for the core damage frequency (CDF) and large early release frequency (LERF) for a variety of potential initiating events. Plant specific PSAs that evolved from the IPEs are maintained today by power plants and updated as plant systems may change. These PSAs are used for both power plant operators and regulators in assessing modifications or updates to plant design *[REF]*. The results of these PSAs give to plant operators a clear definition of what are the most risk-significant systems in a plant and provide a powerful tool for decision making.

There are three levels of PSA performed for nuclear power plants :

- Level 1 PSA quantifies the CDF;
- Level 2 PSA starts from the conclusions of the Level 1 analysis and thenexamines the mode and timing of containment failure and the release of radioactivity material to the environment;
- Level 3 PSA starts from the conclusions of Level 2 analysis and then quantifies the risk in terms of offsite adverse health effects.

Level 1 PSA covers the period defined as “Accident Frequency Analysis” which analyzes

the various plant states which could potentially lead to core damage. Level 2 period covers what is referred to as "Accident Progression Analysis" which estimates the timing and mode of containment failure and "Source Term Analysis" which estimates the size of the radioactive release resulting from various scenarios. Finally, Level 3 analysis covers the "Consequences Analysis" phase of PSA which attempts to estimate the offsite consequences from the Level 2 results. The next three subsections give an overview of the analysis used for each level of PSA. The discussion presented in next three sections will only provide a broad overview of the various levels of PSA. In Section 2.2.4, a more detailed discussion of the classical PSA methodology is presented.

2.2.1 Level 1 PSA

Level 1 analysis is based on event tree / fault tree methodology *[REF]* to determine the set of events which lead to core damage along with their corresponding frequencies given a list of considered initiating events. For each major system or subsystem in the plant that may be called upon during the course of an accident, a fault tree is constructed to explain the failure of the underlying mission from the failure of the basic components involved . . This may include the Emergency Core Cooling System (ECCS), pressurizer relief values, steam generator valves, etc. *Figure 2.1* shows an example of a simple Level 1 Event Tree.

Figure 2.2.1: Example of Fault Tree/Event Tree methodology

Each pathway on the event tree is considered to be one possible scenario of accident evolution. They represent the sequence of system actuations which would occur for a given transient as well as fault-trees that relate the specified undesirable consequences (Top Events) to the failure of basic components. A sequence of successes or failures of the components gives one scenario which may occur in a particular transient. Using the probabilities of failure for each of the systems involved, the probability of each scenario

can be computed. Each of these scenarios leads to what is known as an endstate. Given the knowledge of what systems must actuate in order to prevent unacceptable consequences within a certain period of time, the endstates can be labeled as either leading to core damage or to coresafe states. Since the number of event tree pathways in a typical event tree can number in the hundreds of thousands, it can be necessary to group the endstates into bins with other endstates that result from similar event sequences. In a Level 1 analysis, these binned results are referred to as Plant Damage States (PDSs). The PDSs generated are determined by analysts to be those which most significantly define system behavior. Grouping all of the event tree end states by these sets of characteristics can dramatically reduce the number of scenarios which must be considered in the Level 2 analysis. For each PDS the total probability is the sum of the probabilities of each of the endstates which fall into that category. The CDF of the Core Damage Frequency is then the sum of the probabilities of the Minimal Cut Sets (MCSs) .. A single quantification of the plant event trees gives an estimate for the CDF, given the failure probabilities assumed for each plant component and the assumed frequencies of the initiating events. The likelihood of failure of a particular component or frequency of occurrence of an initiating event can be described by a probability distribution rather than a point value (if sufficient information is available to construct such a distribution). Given the distribution, it is necessary to quantify the event tree multiple times using different values from these distributions. In NUREG1150, the effect of such uncertainty on the CDF was quantified using the LatinHypercube Sampling (LHS) technique. LHS is a method of stratified Monte Carlo Sampling which provides more controlled coverage in sampling a distribution than regular Monte Carlo analysis. Using LHS, values of each of the uncertain parameters are sampled and the event tree quantified multiple times. This results in a distribution on the estimate of the core damage frequency rather than just a pointvalue. The construction and quantification of event trees and fault trees is typically accomplished with software programs.

2.2.2 Level 2 PSA

Level 2 analysis begins with the PDSs which are computed in the Level 1 analysis and are used as initial conditions for the Level 2 accident progression event trees (APET). A Level 2 APET is similar in form to the Level 1 events trees with the exception that the Top Events on the APET do not necessarily correspond to the success or failure of plant systems. Instead, APET Top Events question the occurrence of certain severe accident phenomena, i.e. "What fraction of the core participates in a highpressure melt ejection?" . Typically, APETs are too large to be displayed in the form of an event tree and are typically displayed as a list of APET questions (the Top Events) in the event tree). The APETs used in Level 2 PSA identify, sequentially order, and probabilistically quantify the important events in the progression of a severe accident. The development of an APET consists of :

- identifying potentially important parameters to the accident progression and associated containment building structural response,
- determining possible values of each parameter (including dependencies on outcomes of previous parameters in the event tree),
- ordering the events chronologically,
- defining the information needed to determine each parameter.

The quantification of an APET is primarily based on sensitivity studies performed with accident simulation computer codes and expert judgments that are validated against experimental data. Prior to quantification of the APET, a number of calculations are performed with the accident simulation code to estimate the branching probabilities. These calculations include a range of code parameter variations that provide insights to the analyst on the impact of uncertainties on the probability of alternative branches on the tree. Distributions for each of these questions are constructed which is typically ac-

complished via expert elicitation backed up by experimental results and some mechanistic code calculations *[20]*. Quantification of the Level 2 APET is performed in a manner similar to that of the Level 1 event tree. In this case, LHS is more heavily utilized. For the Level 1 event tree, probability distributions are assigned only to a few Top Events, however in Level 2 analysis the majority of the APET questions have an associated distribution. LHS is used to quantify the event tree multiple times picking random values from the distribution associated with each APET question on each trial. In the end, all pathways on the APET are quantified. Like Level 1 analysis, the number of APET endstates is too large to analyze directly, so binning is performed. This binning focuses primarily on characteristics of the accident evolution which affect the integrity of the containment and potential scrubbing of radioactive aerosols which may be present in the containment. Each endstate is classified according to certain characteristics to build the set of Accident Progression Bins (APBs) which results from each PDS. For each APB, a source term is calculated using a simple parametric model which takes into account the timing and type of containment failure in its estimate of the release. The source term analysis calculates a containment release for each accident progression bin. The code does not model the transport of radioactive material within and out of the containment but is simply a parametric tool used to combine the results from more detailed analyses *[8]*. Note that the source term analysis does not track individual nuclides but rather classes of nuclides. Once releases have been calculated for each APB, the results are even further binned according to similarities in the source term characteristics. The refined binning performed on the source term analysis is then used as an entry point into the consequence analysis or Level 3 PSA.

2.2.3 Level 3 PSA

Level 3 PSA includes coverage of the period after the release of radionuclides to the environment through the estimation of offsite consequences. The Level 3 PSA analysis

takes as input a set of refined bins from the Level 2 analysis in order to estimate offsite consequences. These refined bins are based upon offsite release calculations which are derived from the various APBs which result from the Level 2 APET analysis. The Level 3 analysis utilizes these estimates of the offsite release combined with meteorological and demographic data to estimate the potential offsite consequences. The Level 3 results are typically focused on quantifying the number of potential early fatalities as well as latent cancer fatalities which result from a potential accident scenario.

2.2.4 Conventional PSA Methodology Overview

Conventional PSA usually answers three basic questions *[20, 9]*:

1. What can go wrong with the studied technological entity, or what are the initiators or initiating events (undesirable starting events) that lead to adverse consequence(s)?

2. What and how severe are the potential detriments, or the adverse consequences that the technological entity may be eventually subjected to as a result of the occurrence of the initiator?

3. How likely to occur are these undesirable consequences, or what are their probabilities or frequencies?

The answer to the first question requires technical knowledge of the possible causes leading to detrimental outcomes of a given activity or action. In order to focus on the most important initiators, logic tools like Master Logic Diagrams (MLD) or Failure Modes and Effects Analyses (FMEA) *[20]* have been successfully used until today. In particular, the answers to the second and third questions are obtained by developing and quantifying accident scenarios. The answer to the second question is obtained from deterministic analyses (e.g., thermal, fluid, structural or other engineering analyses) that describe the phenomena that could occur along the path of the accident scenario when the initiator

and the other subsequent events (through the detrimental consequences) take place. The methods used for these deterministic evaluations depend on the specifics of the technology involved. The answer to the third question is obtained by using Boolean Logic methods for model development and by probabilistic or statistical methods for the quantification portion of the model analysis. Boolean logic tools include inductive logic methods like event tree analysis (ETA) or event sequence diagrams (ESD) analysis and deductive methods like fault tree analysis (FTA). In cases when the probability of an event is well known from past experience, statistical actuarial data can be used if the uncertainty in these data are acceptably low. For rare events (e.g., system failures), for which there is no past failure experience at all or the data are very sparse, probabilistic failure models have bee developed with deductive logic tools like fault trees, or inductive logic tools like reliability block diagrams (RBD) and FMEAs. The final result of a conventional PSA is given in the form of a risk curve and the associated uncertainties. The risk curve is generally the plot of the frequency of exceeding a consequence value (the ordinate) as a function of the consequence values (the abscissa). If the risk assessment is qualitative, the result can be represented as a two-dimensional matrix showing probability categories versus consequence categories. In addition to the above model development and quantification, PSA studies require special but often very important analysis tools like human reliability analysis (HRA) and dependent-failure or common-cause failures analysis (CCF). HRA deals with methods for modeling human error, while CCF deals with methods for evaluating the effect of inter-system and inter-component dependencies which tend to cause significant increases in overall system or facility risk. PSA studies can be performed for internal initiating events as well as for external initiating events. Internal initiating events are here defined to be hardware or system failures or operator errors in situations arising from the normal mode of operation of the facility. External initiating events are those encountered outside the domain of the normal operation of a facility. Initiating events associated with the occurrence of natural phenomena (e.g.,

earthquakes, lightning, tornadoes, fires and floods) are typical examples of external initiators.

2.3 Dynamic PSA

2.3.1 DynamicPSAMethodologiesOverview

Despite the significant advancements in PSA that have been introduced into the nuclear industry over the past twenty years, the conventional methods of PSA (as introduced by NUREG1150) contain certain drawbacks that do not always allow for an appropriate modeling of system risk. One of the most pivotal drawbacks of conventional PSA methodology is that time is not explicitly accounted for. During the course of an accident, the exact timing of events could be important in scenario evolution especially when operator action and certain severe accident processes are considered. In addition, the methods of classical PSA does not always provide for mechanistic modeling of all systems and processes in a physically consistent manner. For these reasons (among others), dynamic methods have been developed over the past 25 years to address the deficiencies in conventional PSA modeling. The term dynamic PSA can have several possible meanings depending on the context of the situation, such meanings are:

(1) "Living PSA", a conventional PSA that is updated to reflect plant changes,

(2) PSA model for which component aging is directly considered,

(3) PSA that can be used during plant operation to help operators assess current plant risk,

(4) PSA that couples the stochastic and phenomenological models of the plant to account for possible dependencies between events in which the need for and timing of branching is determined by conditions of the analysis rather than predetermined.

For this work, when discussing dynamic PSA, the last definition listed above is the more meaningful, especially in the sense of discrete events description of system evolution. In this context, dynamic PSA refers to the integration of time dependent phenomenological modeling coupled with an appropriate probabilistic model to explore potential scenario pathways as a function of time. Many other models may also be incorporated into the analysis, including the time dependent behavior of an operator or crew or the time dependent response of components and systems . A true dynamic PSA incorporates models of all possible time dependent elements which have the potential to impact plant safety. Many dynamic methodologies have been developed which attempt to create PSA models that incorporate all of these elements. Dynamic analogs of both event trees and fault trees have been developed for a variety of purposes. Dynamic fault trees are used to model the behavior of components whose failure or recovery rate may be explicitly or implicitly dependent on time. A wide variety of dynamic event tree (DET) methodologies have also been developed. DETs are similar in form to their classical analogs with the exception that the branching process occurs in time and the ordering of events is not preset by the analyst. The DET methodologies that have been developed fall into two basic categories: 1) discrete dynamic event tree (DDET) methodologies and 2) continuous dynamic event tree (CDET) methodologies. In DDET methodologies, branching occurs at fixed points in the system state space. A majority of DDET methodologies branch on fixed time. Namely, at fixed time intervals, the occurrence of certain events is questioned, and there is the potential for branching to occur. Branching can only occur at these fixed points in time. Examples of DDETs methodologies of this type include DYLAM *[20]* (Dynamic Logical Analytical Methodology), developed at the Joint European Center at Ispra, Italy in late 1980s; DETAM *[12]* (Dynamic Event Tree Analysis Method), developed in 1992 by Acosta and Siu; DENDROS (Dynamic Event Network Distributed Risk Oriented Scheduler), developed in 1999 by Munoz and Minguez; and ADS [11] (Accident Dynamics Simulator), developed by Hsueh and Mosleh in 1993. The aforementioned DDET

methodologies have been primarily focused on Level 1 analysis and have primarily considered branching conditions based on component/hardware failures and human action. Recently, the ADAPT (Analysis of Dynamic Accident Progression Trees) methodology has been developed by Hakobyan et. al *[8]*. ADAPT is also a DDET methodology, except that the branching in ADAPT occurs at discrete points in system state space instead of discrete time. The ADAPT methodology has been designed to model the probabilistic behavior of active components, passive components, severe accident phenomena, and has also been applied to human reliability analysis. The ADAPT tool was primarily developed to model the stochastic behavior of passive components and severe accident phenomena, but it has been extended to work outside this scope. An overview of the ADAPT methodology is provided in section 2.3.2. CDETs are different from DDETs in the fact that branching in CDETs occurs continuously (i.e. can occur at any time and at any point in state space). Practically, branching in CDETs is performed via Monte Carlo analysis. Many scenarios are generated where randomly generated branching conditions are injected into a process simulator. After enough simulations, average behavior can be determined from the results. Once such example of a methodology of this type is the MCDET *[REF]* (Monte Carlo DET) methodology developed by Hofer, Kloos and others in 2002 developed at Gesellschaft für Anlagen und Reaktorsicherheit (GRS), in Germany. MCDET has been primarily developed for Level 2 applications and for practical purposes has been linked to the MELCOR severe accident analysis code. There are also other types of dynamic methodologies which do not necessarily extend from classical PRA constructs. For example, Maseguerra, Ricotti, and Zio have developed dynamic PRA methods based on neural networks *[REF]*. Another example is the Dynamic System Doctor (DSD) methodology developed by Wang, Chen, and Aldemir *[13]*, which utilizes the CelltoCell Mapping Technique for modelbased fault diagnosis in dynamic systems. The review of methods provided here is not meant to be exhaustive, but merely to give the reader a sampling of the different types of dynamic methodologies that exist.

2.3.2 The ADAPT Methodology

The ADAPT (Analysis of Dynamic Accident Progression Trees) methodology is a discrete dynamic event tree methodology developed by Hakobyan et al. *[8]* and it is the more suitable methodology used today as reference for dynamic PSA. In the ADAPT methodology, a simulation tool is utilized to explore many pathways of scenario evolution in a severe accident using an analyst specified set of branching conditions. ADAPT itself provides an overlying probabilistic model to the simulation of a physical process. Many discrete DET methodologies branch on fixed time . However, in the ADAPT methodology, in addition to user specified time points, branching conditions may depend on the system history as well as system locations in the state space. The ADAPT methodology can be used to propagate uncertainty from a wide variety of sources to determine not only the distribution on system output uncertainty, but can also be used to discover event sequences which may be important to system risk and may not be discoverable by classical methods. The ADAPT methodology can consider branching conditions on a wide range of stochastic events including the success or failure of active systems or the occurrence of various severe accident phenomena. In addition, since a consistent model is being used to track system history for all scenarios (a simulator of user choice), uncertainty in code modeling can also be propagated by examining how variations in code models affect output data and event sequences. One unique aspect of the ADAPT methodology is the manner in which passive components and phenomenology are modeled.

2.3.3 A tool developed by EDF R&D : Boolean Logic Driven Markov Process (BDMP)

Fault-trees are undoubtedly the easiest and most often used technique in complex systems dependability assessment. Many people have refined this technique which has been applied to various industries, including aerospace, medical, and nuclear. Thanks to the state of the art fault tree algorithms based on Binary Decision Diagrams (BDDs), it

has become possible to compute the exact value of availability, a good approximation of reliability, and various importance measures for a very large repairable system (typically: with several hundreds of components) modeled by a fault tree in a few seconds on a PC *[15]*. However, conventional fault-trees (in the following called "static" fault-trees) are not at all suited to modeling systems in which there are strong dependencies between components. The assumption of components independence is precisely what makes fault-trees so powerful, but this assumption is extremely restrictive, and may prove to be totally unrealistic and lead to grossly erroneous results for some kinds of systems. In order to be able to model component dependencies, one has to recur to dynamic models. The most popular of these are Markov processes, because of their numerous nice mathematical properties [15, 5]. In practice, the direct use of Markov processes has virtually been given up, to be replaced by some higher level formalisms that enable the automatic generation of a (potentially huge) Markov chain. The problem with these higher level representations, like stochastic Petri nets , is that they are much too general. By 'much too general', we mean that it is impossible to infer any interesting property of the Markov graph, that could be used to simplify its processing, from the model input by the user. Therefore it appears that some kind of trade-off must be chosen between static fault trees, which have a low modeling power, but are extremely easy to process, and general dynamic models such as Petri nets, which enable the construction of much more accurate, but unfortunately intractable models. This is in fact the purpose of the new concept of 'Boolean logic Driven Markov Processes'.

2.3.3.1 From dynamic fault-trees to Boolean logic driven Markov processes (BDMP)

In the light of the observations made in the introduction, its seems obvious that two possibilities arise:

(1) try to impose some constraints on very general dynamic formalisms so as to make

it possible to infer some properties from the models built by the user, in order to make the processing more tractable,

(2) try to extend static fault trees by adding some dynamic features.

Unsurprisingly, both approaches have been developed. The first possibility has been explored mainly in the field of computer systems performance analysis [15], because in this field, models usually present a high degree of symmetry, which allows efficient state lumping in Markov processes. For example, Plateau and Stewart *[1]* have developed the concept of 'Stochastic Automata Networks' ; another important research stream is around the concept of so-called "well-formed (Petri) nets". Starting from the observation that failures which result from the ordering of specific events cannot be modeled using static fault trees, some of these dependencies, which can be modeled using Markov models, have been represented with special gates that complement the existing static fault tree gates. Among the sequence dependencies that can be modeled using dynamic fault trees are functional dependencies and sparing. Functional dependencies are considered to be the occurrence of some event, called the trigger event, which causes a set of dependent components to become unusable. Sparing involves the sequencing of events associated with the replacement of a failed component with either a hot, warm or cold spare. Although BDMP may seem similar to dynamic fault trees, they are in fact quite different. Instead of adding new kinds of gates, they assign a new semantics to the traditional graphical representation of faulttrees, augmented only by a new kind of links (these links are called 'triggers' and are represented by dotted arrows). They enable the analyst to combine conventional fault trees and Markov models in a brand new way. In fact, they offer much more modeling power, than a simple juxtaposition of these formalisms, as can be seen in the examples we provide in this article. Moreover, BDMP have very interesting mathematical properties, which allows a dramatic reduction of combinatorial problems in operational applications, especially when they are processed using a method based on sequences exploration. This method not only is able to process BDMP equivalent

to Markov processes with huge state spaces, but also gives very interesting qualitative results: the most probable sequences that lead to an undesirable state.

2.3.3.2 From petri nets to BDMP

The general idea of BDMP, as suggested by their name, is to associate a Markov process (which represents the behavior of a component or a subsystem) to each leaf of a fault-tree. This fault-tree is the structure function of the system. What is really new with BDMP is that: the basic Markov processes have two 'modes', corresponding to the fact that the components/subsystems that they model are required or are in standby (of course, they can also have only one mode, and the meaning of the modes may be different in some cases). At each time, the choice of the mode of one of the Markov processes (unless it is independent) depends on the value of a Boolean function of other processes. An extreme case is when the processes are independent. Then it will be seen that this corresponds to a fault-tree, the leaves of which are associated to independent Markov processes. From a theoretical point of view, Boolean functions and Markov processes are all what is needed to define BDMP. The predefined processes are sufficient to model a large variety of systems, some of them showing a very complex dynamic behavior. However, the possibility to recur to Petri nets gives us the assurance that BDMP are very general.

2.3.4 Smarts (Intelligent) Generic Components Approach

Dynamic PSA methodologies, in general, do not possess a generic model based method for the system or the scenarios being represented. Since most of the applications have been problem specific, the computer codes were written merely to test a concept. So, the system descriptions were "hard coded" into the program or provided to the computation engine by means of a text based input format, specific to the particular version of the code or programminglanguage. Not much effort had been made to determine a qualita-

tive representation technique that would facilitate usage of the dynamic computational methodology. This lack of user friendliness is one of the important reasons for the limited usage of dynamic methodologies. Such a user friendly representation tool could be used to perform a qualitative analysis of the system. This is fundamental to modeling systems which are based on similar design concepts, but having different configurations. In recent years some effort has been made in this direction for capturing the dynamic system behavior and one these is the so called smarts (intelligent) components approach.

This is a component based scheme of modeling. The concept of object oriented programming is well known in the software engineering field *[4]*. In recent years, the principle of object oriented modeling has been used in other areas, like physical modeling and risk analysis. This involves defining each component in the system by means of an object and various instantiations of these objects can be created. These objects have their attributes and functioning mode. Attributes can be further classified into state and characteristics. A vector of discrete or continuous variables describes the state of the object and characteristics are used to individualize an object; they remain unchanged during simulation. Functioning modes are an essential feature of object oriented modeling and these are used to describe the behavior of the object. Various components, e.g. busbars, breakers, pumps etc. can be defined, along with their attributes, e.g. state transitions, and functioning modes , e.g. functions of the components. These objects can be stored in a database to create a library of components. Using this library of components, one can create particular instances of the same type of component for the system in question. The concept of inheritance makes it possible to extend the definition of these classes by incorporating additional attributes and methods. Once the object model is defined, all components and their possible state transitions are defined. One can then interface this library with an appropriate simulation code that generates all possible scenarios. The basic theme behind probabilistic dynamics would be something like this?. The construction of the transition matrix can also be automated from the transition descriptions of

the various components . This approach frees the user from defining scenarios (ESD) and directly let him to generating state graphs that contain all possible system states during its life cycle. One of the prime advantages of using a library of objects representing components is that this library can be created even during the design stage of the system. Once an object is defined in the library, it can be reused in several applications, similar to the use of generic failure data in nuclear PSA. With smart generic components, it is not only possible to reuse the failure data but the complete definition of the component. Another important advantage of this method is that it can be used in "living PSA". As systems grows older and new data on certain components are available or replaced , the system model can be very easily updated, because this would simply imply a change to an attribute of an object or creation of a new object in the system library. Although the creation of the library does not possess challenges, the direct simulation of system evolution could be computational demanding and so the use of advanced Monte Carlo techniques in combination with this representation method is to be studied. With advancement in computers and development in automatic Monte Carlo biasing techniques, it may be possible to use "smart" components for performing dynamic reliability analysis in the future. This work is an attempt to demonstrate that some efforts have been already done in this sense at EDF R&D department, and in chapters 4-7 the tool developed as well as the results obtained through Monte Carlo simulation are shown.

3 Overview of EPR Reference PSA Analysis

3.1 Introduction

The purpose of this chapter is to give a definition of internal initiating events considered in the EPR Level 1 PSA showing the methodology and the results, the latter presented in terms of Core Damage Frequency (CDF) per reactor per year. Thanks to the cooperation between EDF and Areva probabilistic studies were carried out during the design process of the EPR to support and optimize the design of systems and processes. In PSA classical fault trees were used to estimate the probability of failure of the mission system, while event trees have been commonly used to estimate the CDF due to each initiating event. The risk quantification was performed using RiskSpectrum ® Professional [14], one of the most advanced risk and reliability analysis software throughout the world, which includes tools for fault tree and event tree modeling and analysis, documentation, monitoring risk assessment of human reliability and failure mode and effect analysis *[REF]*.

The PSA level 1 analysis aims to address all potential accidents related to the reactor core that could lead to radioactive releases into the nuclear power plant. The scope of the EPR PSA is defined as below:

- All reactor operational modes are covered, from operation at full power to shutdown for refuelling with at least one fuel element in the reactor vessel.

- The study is limited to internal events.
- The internal initiating events considered are presented and justified, then internal and external threats are addressed.
- Thermohydraulic and neutronic parameters are based on specific study relying on dedicated deterministic computation codes.
- Component availabilities are based on scheduled preventive maintenance.

3.2 Methodology

The methodology used in the level 1 PSA to model the EPR [14]considers the following events:

- Random individual component failures.
- Components which fail as a result of the initiating fault (cascading failures [1]).
- Common cause failures (involving both components and signals).
- Preaccidental human errors and human errors occurring during the course of fault sequences.
- Potential dependencies between separate human activities.
- External events that have the potential for initiating a plant transient (fires,floods, earthquakes, loss of offsite power, ecc)

Equipment unavailabilities due to repair and preventative maintenance activities at power and shutdown states are included in the base case PSA model. Uncertainty analyses using a MonteCarlo methodology are performed to derive confidence levels for the PSA

[1] These are multiple failures initiateed by the failure of one component in the system, as a sort of domino effect

results. The analyses take into account uncertainties in reliability data and initiating event frequencies by inputting these parameters as probability distributions. In the following sections we will briefly look at some specific aspects of the modeling of the system in the EPR level 1 PSA.

3.2.1 System Mission Time

The mission time is defined as the time that has elapsed following the initiator and during which the possible failures that affect the mission of PSA may occur. In this case, the EPR Level 1 PSA considers sequences durations up to the time to reach a state of failure. Thermohydraulic complementary studies confirm that such a state is reached in a period of time much shorter than 24 hours [2]. Thus, the EPR PSA model mainly uses once the mission of 24 hours, in line with international practice, regardless of the actual duration of the various missions required. Mission times of less than 24 hours are only used for the components required to maintain the power supply (batteries, diesel generators) in the frame of LOOP events in the short term, which are limited to 2 hours.

3.2.2 Plant Operating States

The EPR plant passes through multiple configurations during a cycle of operation and a series of "standard states of the reactor" are defined in correspondence with these different configurations. In order to make the number of states of the reactor tractable, the states are grouped by means of a qualitative assessment. Practical guidelines to be pursued in the assignements for each state of components groups, based on key attributes which can affect components interdependencies, may be the following:

- similarity of the parameters.
- similarity of available systems and components.

[2] The extension of the mission time up to 24 hours has been strongly demanded by international control entities, as a result of the Fukushima accident occurred in March 2011

- • similarity of potential initiating events.
- similarity of component external environmental conditions

In some cases, the conditions of the plant such as pressure, temperature, system availability, decay heat level etc. can change with time within a state of the reactor. In such cases, conservative assumptions were generally made in the PSA .

3.2.2.1 Power State (A, B)

In most cases, the EPR PSA studies for power states address reactor states A and B together, because similar functional analyses apply. The exceptions are the following:

- Boron dilution events which are studied separately for states A and B.
- Loss of main feedwater, loss of condenser, turbine trip, anticipated transient without scram and the reactor trip, which are studied only for State A.
- Loss of the Startup and Shutdown System which are studied only in State B.

In states A and B, the plant is assumed to be at full power with all systems available, all controls in operation, and the core thermal power being removed via the steam generators. . The average time spent in standard reactor states A and B represents 94% of the cycle duration. Preventative maintenance is technically possible during power operation and is permitted on some safety systems *[REF]*.

3.2.2.2 Shutdown States (C, D, E, F)

For these states the plant condition considered in the PSA model represents plant shutdown rather than plant startup. For example during plant startup in State Ca, four reactor coolant pumps are in operation to heat up the reactor coolant. The PSA models the corresponding shutdown configuration since the core thermal power is much greater during plant shutdown than during plant startup.

3.2.3 Reliability Data

Reliability data are derived mainly from operational experience feedback from France and Germany *[14]*, supplemented by data from the EG&G generic reliability database. Reliability data used for instrumentation and control systems are defined. Failure modes and related reliability data used for equipment other than instrumentation and control systems are detailed in the UK EPR data report for the Hinkley Point C EPR on the one hand, and in the FLA3 EPR data report for the Flamanville 3 EPR on the other hand. For each component type, the component boundary is presented according to the definition given in the relevant source. The EPR PSA fault tree modelling in the system analyses is consistent with these component boundaries. . Broadly, the methodology adopted in the UK EPR PSA uses parameters taken from the EDF database.

3.2.4 Preventative Maintenance

Unavailability due to preventative maintenance has been included in the Level 1 PSA model base case. Additionally, the increase of risk caused by maintenance activities is considered via a sensitivity analysis which evaluates the maximum impact on risk of unavailabilities due to preventative maintenance. The results of this sensitivity analysis demonstrate the robustness of the EPR design and show that the design meets the probabilistic safety objectives. The maintenance scenario considers the following preventative maintenance on certain groups of systems that were determined by a functional analysis.

3.2.5 Common Cause Failures

Common cause failures (CCF) are failures on demand or during a system mission period that could simultaneously affect several components, where the failures are due to the same cause. Common cause failures include failures of equipment due to errors in design, manufacture, installation or operation. CCF applies to groups of redundant equipment items operating under similar conditions. The same CCF model is used for different

types of component: pumps, valves, diesels, high and medium voltage circuit breakers, sensors etc.

No account is taken of a CCF of equipment items in the following cases:

- When items of equipment are not required to change state during an accident (e.g. switchboards, piping etc). For example simultaneous leakage due to corrosion of pipework in four redundant trains, is not considered as a CCF. It is considered that such leakages could occur at any time, and hence it is likely that the damage would be detected during normal operation, leading to a program of repair and prevention on the redundant equipment in the trains.

- When several components, such as contactors and emergency switchgear, are operating under similar conditions before the initiator occurrence. In this case the failures modes would be likely to be detected by observation, allowing corrective measures to be carried out.

CCF of components is considered when the components belong to the same system and have the same function. For example, CCFs of the low voltage circuit breakers are considered in the modelling of the EPR subsystem. CCF of the same component belonging to different systems is not considered because such components would have different functions and be subject to different test and maintenance regimes. Therefore, for example, CCF of the low voltage circuit breakers of the pumps and similar equipment on other systems is not considered.

3.2.6 Instrumentation and Control

The important role played by the instrumentation and control system (I&C) is modelled in the PSA by using a specific reliability model called the 'Compact Failure Model'. PSA modelling of the I&C systems is implemented in two stages:

- modelling of the I&C control channels with the 'Compact Failure Model'.

- global integration of the I&C functions into the PSA model.

The method of modelling I&C is referred to as the 'Compact Failure Model' (CM), which is a simplified functional representation of the I&C digital systems in the PSA. The use of CM is assumed to be appropriate to use for the EPR PSA. The CM is based on splitting the I&C digital system into elementary I&C functions, also called channels, each one being represented by a specific fault tree in the PSA. According to the CM, each single I&C function is broken down into three main parts, as shown in the following symbolic representation: an instrumentation part, a specific and non,specific processing part, and an actuator part. Symbolic representation of an automatic I&C function for final integration into the PSA event trees, these symbolic representations of failures are converted into fault trees.

In the fault trees, fixed numerical values are used for the overall unavailabilities of the instrumentation and processing parts. These values depend on the classification and are directly used in the PSA Boolean modelling. It will now be given a brief description of the I&C representation model used in the EPR level 1 PSA.

Acquisition Part The acquisition part corresponds to the sensors used as input to the I&C functions. The term "sensors" includes the measuring cell module, the electronic converter and the transmission connector technology. Modelling of the instrumentation part does not exactly conform to the CM principle. The CM principle recommends modelling the instrumentation part by using groups of redundant sensors. However, for a given I&C function, all the sensors required for the elaboration of the signal are separately modelled. When redundant sensors exist, a logic gate is used in order to represent the voting logic between the sensors. Various types of redundancy and voting logic are modelled in the EPR PSA (for example 2 out of 4, 2 out of 3 or 1 out of 2). Some exceptions exist where a single basic event is used to represent several sensors (for example, rod position sensors or Self Powered Neutron Detectors).

Processing Part The processing part corresponds to the processing functions implemented in the following computerised I&C systems: the Protection System, the Safety Automation System, the Process Automation System, the Reactor Control Surveillance and Limitation System and the Severe Accident I&C. These functions receive a signal from the instrumentation part. According to the CM rules, the processing part of a given I&C system is divided into two parts: the specific processing part and the non specific (also called "common") processing part. The "specific logic" processing part relates to a given safety function and its processing logic. It extends from the acquisition of the parameters (downstream of the sensors) to the generation of the partial instructions (before voting). It includes all the redundant printed circuit boards (hardware and software) used by the associated safety function and required for partial instructions. The "common" processing part takes into account all the components used for voting processing. Moreover It includesall the elements, systems and common protocols necessary for data transmission (e.g. the data buses, the exchange protocols). This part also includes the representation of common equipment points as well as Common Cause Failures that may be introduced by use of common technology. The Reactor Protection System is divided into two subsystems, A and B: all the signals processed in a given subsystem are affected by the failure of this subsystem.

Actuator Part The actuator part corresponds to the elements that support the action on the process subfunction. It represents set of actuators (pumps motor, valves drive) and includes their associated electrical interface (switchgear) and the I&C part supporting the basic actuator control subfunctions. The actuators themselves are not included in the actuator part. The number of actuator trains depends on the degree of redundancy of the mechanical or electrical system supporting the safety function. At present, the actuator part is not modelled in the PSA model. Since modelling of the actuator part is not dealt with in the CM, this actuator part will be considered during modelling of the

related specific systems (and not in the I&C modelling), when the detailed allocation of acquisition controls is specified. It is assumed that not modelling the actuator part does not significantly underestimate risk.

3.3 Loss of Offsite Power (LOOP)

The availability of power supply is essential for safe operation and accident recovery at the most recent nuclear power plants. Normally, electric power is supplied by offsite sources via the electrical grid. Loss of this offsite power can have a major negative impact on the plant's ability to achieve and maintain safe shutdown conditions. Risk analyses performed for the EPR indicate that the loss of all electric power contributes over 70% of the overall risk at some plants. Clearly, loss of offsite power (also referred to LOOP) and subsequent restoration of offsite power are important inputs to plant PSA.

3.3.1 Initianting Events Analysis

The assessment of the LOOP initiating event is performed for two types of LOOP event considering short ($<$ 2 hours) and long term (up to 24 hours) duration LOOP. Loss Of Offsite Power is defined as loss of both the main and auxiliary grid connections. For the UK EPR, automatic switchover to house load operation is conservatively assumed to fail with a probability of 1[3] . The analysis covers also the Station Black Out (SBO) situations which are defined as Loss Of Offsite Power with the occurrence of low voltage on the four 10kV safety busbars LHA, LHB, LHC, LHD, each one backed up by an Emergency Diesel Generators (EDG). The following initiating events are considered in the LOOP group for state A and B:

- Short term LOOP in at power states A andB (LOOPS AB). This includes the LOOP event caused by a reactor trip (consequential LOOPS AB).

[3]It is considered to be successfull in the case of FLA 3

- Long term LOOP in power states andB (LOOPL AB). This includes the LOOP event caused by a reactor trip (consequential LOOPL AB).

A consequential Loss Of Offsite Power is defined as a loss of main and auxiliary grid connections due to the reactor trip following non LOOP initiating events such as spurious reactor trip, turbine trip, Loss of Condenser or Loss of Main Feedwater.

3.3.2 Dominant Accident Sequence Analysis

The following apply for each initiating event of the LOOP group:

- The Emergency Diesel Generators are automatically started, while the Station Blackout Diesels are usually started by the operator from the Main Control Room (MCR) after the loss of all Emergency Diesel Generators. Both starts require the availability of the batteries, the 220V uninterrupted power supply.

- The reloading sequence, which follows the start of the Emergency Diesel Generators, maintains the power supply to the safety trains and their support systems. The following systems are supported by this action: the Emergency Feedwater System , the Component Cooling Water System , and the Essential Service Water System , the Chemical and Volume Control System, the Safety Injection System and the Containment Heat Removal System.

- If the batteries fail, the Station Blackout Diesel Generators can be started manually by local action. This backup is only considered when the batteries fail in operation after a long time window.

The following sections cover Short term Loss Of Offsite Power and Long term Loss Of Offsite Power at power states A and B, being these situations dealt with in the analysis of this work.

3.3.2.1 Short Loss of Offsite Power (LOOPS)

The short term LOOP is defined as the maximum duration which can be survived without any electrical supply from the diesels or the grid. This period is limited to 2 hours by:

- The water inventory of the steam generators which will provide approximately 1 hour 30 minutes of steaming through the Main Steam Safety Valves or Main Steam Relief Train without feeding, followed by
- RCP heatup with cycling of the Pressuriser Safety Valves for about 30 minutes.

Consequently, the short term LOOP event does not explicitly require the Emergency Diesel Generators (EDG) to start. The 2 hour batteries are required for the 2 hour LOOP to supply the LV busbars for Instrumentation and Control (I&C) and for operation of the Main Steam Relief Trains. Unavailability of those busbars is mainly caused by failure of the 2 hour batteries and the failure of the operator to start the SBO diesels manually to supply the busbars (following the failure in operation of the batteries). However, as soon as the power is recovered, the core recovery is only possible with the manual actuation of Feed and Bleed. Indeed, the steam generators are dry on the secondary side and refilling of hot and dry steam generators is usually not foreseen in the emergency operating procedures.

The safety functions which are challenged by the short term LOOP event in at power states A and B are :

- Reactivity Control
- Removal of core decay heat and stored heat.
- Reactor Coolant System integrity
- Reactor Coolant System inventory control.

3.3.2.2 Long Loss of Offsite Power (LOOPL)

The long term LOOP is is also defined as the the short term LOOP but assumed to last for 24 hours, according with the most recent international safety requirements. The functional safety requirements which are challenged by the Long term LOOP event in at power states A and B are :

- Reactivity Control
- Removal of core decay heat and stored heat
- Reactor Coolant System integrity
- Reactor Coolant System inventory control.

3.3.3 Contribution of the LOOP family

The contribution of the LOOP group to the Internal Event Core Damage Frequency is 3.1E07/r.y, which represents 44.2% of the Internal Event CDF. The partitioning of the CDF between power and shutdown is 96% for at power states (A, B), including consequential LOOP and 4% for shutdown states (C and D). The relative contribution of each LOOP initiating event within the group is given below:

The Core Damage Frequency of the short LOOP family is decreased by about 32%, for at power states and for shutdown states. That is directly linked to the decrease of the initiating event frequency from 6E 02/r.y. to 4E 02/r.y [19, 14]. The Core Damage Frequency of the long LOOP family is increased by about a factor 3, for at power states and for shutdown states. That is directly linked to the increase of the initiating event frequency from a generic one of 1E 03/r.y. to a Hinkley Point site specific of 5E 03/r.y. Therefore, the Core Damage Frequency of the overall Loss Of Offsite Power family has been increased by 109% and is now significant.

The LOOP events show the relative importance of the electrical supply.

Long Loss of Offsite Power: The sequences arising from the Long term LOOP in at power states (A,B) represent 84% of the CDF for this group. The major part is related to the LOOP as an initiating event. During shutdown states, the long term LOOP contributes about 4% of the CDF for the group. For long term LOOP, the diesels are the most important components. Each function contributing to the residual heat removal requires the operation of at least one of the four Emergency Diesel Generators, or one of the two SBO Diesel Generators.

Short Term Loss of Offsite Power: The sequences arising from the Short term LOOP in at power states represent 13% of the CDF for this group. For the short term LOOP in at power states, the Reactor Coolant Pump Seals and the DEA are particularly important because their failure causes a small LOCA [4]. Two diverse sets of batteries (2 hours batteries and 12 hours batteries) have been modelled in the UK EPR PSA; the common cause failure of the four batteries is considered in the analysis. Failure of the two sets of batteries results in the unavailability of the electrical supply to the I&C and the actuators. The PSA assumes that the core damage would occur in this situation. During shutdown states, the short term LOOP is negligible.

[4]The Loss Of Coolant Accident are those accedents that result in a loss of reactor coolant at a rate in excess of the capability of the reactor makeup system from breaks in the reactor coolant pressure boundary, up to and including a break equivalent in size to the double-ended rupture of the largest pipe of the reactor coolant system

4 The FIGARO Platform

4.1 KB3 And The FIGARO Modeling Language

In order to improve the quality, rapidity, and accessibility of dependability studies, EDF developed the KB3 program *[5]*. KB3 automatically builds reliability models of structural types (fault trees or systems of Boolean equations) or behavioral types (Markov graphs, Monte Carlo simulation models, etc.) for studying a system on the basis of its graphical description. . To carry out dependability studies with KB3, a knowledge base *[17]* adapted to the problems involved in the studies to be carried out is needed. Such a knowledge base must contain a generic description of the different kinds of components that might be encountered in the studies (description of possible component failure modes and of their consequences on the system). This single generic description is independent of the topology of a given system and can therefore be used for all system studies involving the problems dealt with. The KB3 knowledge bases are written in Figaro, an original language developed by EDF. For 20 years now, KB3 has been used in numerous operational applications, and has consequently participated in the development of a very wide range of knowledge bases to meet the requirements of those applications. The following sections provide a brief overview of the FIGARO language and the key elements of modeling using it.

4.2 General Presentation

The FIGARO language is a hybrid language in the sense that it is both an object-oriented language and an artificial intelligence language in which objects' behaviour is described by rules. It is a general modelling language with the following objectives:

- provide an appropriate formalism for developing knowledge bases (with generic descriptions of components)
- be more general than all the usual reliability models
- find the best trade off between modelling power (or generality) and possibilities for the processing of models
- be as legible as possible
- be easily associated with graphic representations.

Basically, the FIGARO allows to compactly describe very complex stochastic automaton without explicitly expressing all the possible state-space of target systems. . the smart-components modelling approach is based on this fundamental characteristic.

4.3 Brief Introduction Of FIGARO

This section is intended to give the minimum necessary to introduce the two derivation of the FIGARO language [17]: FIGARO 0 and FIGARO 1 along with the relations between them.. Apart from some global information , a knowledge base contains generic models (introduced by the keyword CLASS) whose instances can be used the describe a whole system. Each class contains the following set of fields, all of which except for the class declaration are optional:

CLASS t KIND_OF t1 t2 ;

INTERFACE i1 KIND t1 CARDINAL 1 TO INFINITY ;

i2 KIND t2 ;

CONSTANT c1 DEFAULT { constant expression (Boolean, numeric, character string) };

DIST_PARAMETER p1 DEFAULT {constant numeric expression};

FAILURE p1 LABEL "first failure mode of %OBJECT" ;

p2 ;

EFFECT e1 LABEL "first effect of %OBJECT";

e2 ;

ATTRIBUTE a1 DOMAIN BOOLEAN DEFAULT FALSE;

a2 DOMAIN 'value1' 'value2' 'value3' DEFAULT 'value1';

OCCURRENCE { occurrence rules } INTERACTION { interaction rules }

A class basically consists of two parts:

(1) Static and declarative part:

- Seclaration of the name of the class and of the class(es) whose characteristics it has inherited.
- Seclaration of interfaces, namely otherclasses that interact with the considered class with possibly some constraints on the cardinality of objects in each interface.
- Declaration of constants.
- Declaration of class attributes, namely the state variables of the considered class- with their initial value.

(2) Dynamic part:

Two kind of rules are available in the FIGARO language to describe the behavior of an object : the occurrence rules and the interaction rules. The occurrence rules describe the conditions governing the occurrence of the stochastic transitions an object goes through. On the other hand, the purpose of the interaction rules is to propagate the deterministic consequences following the occurrence of a system transition *[REF]*.

These rules often make use of quantifiers in order to be valid irrespective of the content of sets of objects defined by the interfaces; some examples are given below. This set composed by the knowledge base and a system description given as a list of objects linked by their interfaces consists in a complete FIGARO 1 model associated to the considered system . To prepare this model to the quantification step, a conversion in FIGARO 0 is performed which basically corresponds to the following operations:

- application of inheritance and overwriting rules to every object in the system,
- elimination of the quantifiers in the rules. This is made possible by the fact that quantifiers concern sets that are known by the list of their elements: the interfaces of objects. The rules obtained will generally be simpler (and in some cases they are simply eliminated), but also in a larger number since they will be repeated as many times as there are objects of a given class.

5 Modeling technique

5.1 Presentation Of CONCERTO

In the context of the increasing need for research tools and methods for dynamic modeling of systems, the development of a new formalism has been initiated for this work. This formalism is called CONCERTO (the name comes from an anagram of its broader definition, *COmplex mechaNiCal systEm Representation TOolbox*), a knowledge base written with the FIGARO language completely dedicated to the dynamic modeling of some generic electrical and hydraulic components though only the electrical part is used . By developing CONCERTO, the aim is to provide a library of pre-existing electrical components and to allow :

- the representation of power sources of a generic industrial plant, and then
- let the analyst to easily manage it to perform a dynamic PSA.

As mentioned in the Chapter \ref{chap1}, the principle of modeling by which we proceeded in the realization of CONCERTO is very close to that of the method of smart (intelligent) components [20]. In Section 5.2 we discuss the general approach adopted in the construction of the knowledge base; in section 5.3 it is shown in detail how the dynamic behavior of the generic components of interest is modeled for the representation of a power source. The last 3 sections present the dynamic modeling of preventive maintenance, common cause failures (CCF) and I&C system, respectively.

5.2 General approach

In FIGARO, a CLASS definition, i.e. a generic object, is performed in the manner presented in section 4.3. In this specific case, the generic object is an electrical component whose functioning mode and characteristics in terms of failure are appropriately described starting from the real physical principle on which they relay. The general approach that we adopted was therefore to simplify as much as possible the behavior of each component, trying to define its fundamental properties. At the beginning we focused on the single type of component, trying to answer the following questions :

- What is its proper function?
- How does it relate to other system components?
- Which are its mechanisms of failure?

After having roughly defined the various components, it has been adopted a reverse logic of representation. The objective was that of categorizing them on the basis of certain criteria. The criteria that have been taken into account for this grouping process are :

- the *failure mode* (e.g. failure to run, failure on demand etc.),
- its proper *function* in electric power propagation (e.g. consumer, source etc.),
- and, finally, the possibility of being *controlled* (e.g. by an I&C system, an operator).

Initially, the focus has been given to the role played by the component in electrical distribution. For example, it is obvious that a switch has a function within an electrical system that is completely different from that covered by a transformer. It was gradually discovered the importance and effectiveness of defining classes of components in response to these criteria. The structure of the knowledge base responds to this need. To achieve this goal, instead of specific components abstract levels were built for each of the three

identified criteria . The latter of them was considered in a second time, given its considerable complexity either at the level of understanding of the phenomenon and from the point of view of its modeling. Taking advantage of the cascade process of construction based on the inheritance properties of FIGARO, it was possible to build the single real component starting from the abstract classes. Later, when the definition of real components reached a good degree of detail, we passed to a deeper level of modeling. Subsection 5.2.1 presents how the electrical power propagation has been modeled; subsection 5.2.2 describes the structure of CONCERTO.

5.2.1 Electric Power Propagation Modeling

From a theoretical point of view, electricity can be thought as a fluid flowing through the components that populate an electrical system. The basic idea of representing this phenomenon is that all components has a specified role in the flow distribution. However, in many cases the operation mode itself of the component is dependent on the received input flow: in other words, for some components it is essential to be powered by the electrical flow. From this reasoning, two key concepts arise to model an electrical system. With reference to the generic component, they are :

- input power,
- output current.

These two characteristics are expressed in CONCERTO by means of the *boolean attributes* `fed` and `flowed` *respectively* . At each transition of the system, the attributes fed and flowed are updated for all the components part of the considered system according the basic electrical propagation implemented in CONCERTO. A second objective was to allocate the components within these *categories* : source, consumer, link and node. Depending on the category to which the component belongs, the electrical propagation through it is consequently modeled. As stated earlier, modeling is done starting from

abstract levels of representation and it is at these levels that the electrical propagation mechanisms are defined. Since it is a dynamic modeling, internal rules (interactions) verify the necessary conditions allowing the electric propagation and these conditions vary with the type of component. For example, at our level of modeling, an alternator is considered as a perfect source, , so the attribute *fed* is always set to the value true for these class of components..

5.2.2 Knowledge base structure

CONCERTO has been designed starting from the principles of inheritance and categorization of components that have been mentioned before. Different classes at different levels of modeling have been realized to be combined and finally represent real components. Generally, the behavior of a component is defined from an abstract level of modeling, which defines the general properties of a class of components and not of a specific one.. For example, the *abstract* level defines components for the following classes: *ComponentAbstract, FailureAbstract , ControlledCompAbstract, StartingCompAbstract, OperatedCompAbstract, SourceAbstract, SourceControlledAbstract, ConsumerAbstract, NodeAbstract.* In fact, the knowledge base structure has been designed so that a component could be modeled using generic level classes, which specify its global behavior when combined. Ultimately, a real component is an entity that inherits all the properties defined in the abstract classes upstream. This type of structure allows CONCERTO user to manipulate a component at its different levels : abstract description, specific functioning mode, specific failures, common cause failures etc. If a concept of the modeling has not been well thought out and must be changed, it is possible to act at a general level so that the correction impacts on all components that share that characteristic behavior.

5.3 Components

As we mentioned in subsection 2.3.4, CONCERTO has been essentially designed to provide a library of objects aiming to represent electrical and hydraulic components (even if most of the development involved the electrical aspects in this study). Each specific component is described by rules and attributes that inherit from upstream generic classes. There are different levels of description of the behavior of a component. Section 5.3.1 presents the generic levels. Section 5.3.2 goes into detail of real components.

5.3.1 Generic Components

In the knowledge base the generic description is made following these three levels:

1. abstract components;
2. generic electric components;
3. failures modes.

An abstract component has properties and attributes that may represent different types of component. By using the attributes *fed* and *flowed* , the electrical propagation mode is specified for the various types of component. Then, it is assumed that the generic component can be maintained and so the possibility to be selected within a group of maintenance is defined. Since all components can potentially fail, then components have some special attributes indicating the types of faults which may be encountered during their life cycle. Typically a component may fail in two ways :

- in the course of its operation, i.e. a failure to run;
- on request of control system, and in this case, two possibilities can occur : failure to start and failure to operate.

Abstract Components

Type name	Description	Parent type
ComponentAbstract	All components inherit and therefore share the attributes of this class	-
ControlledCompAbstract	Components required by the control system to operate	ComponentAbstract
StartingCompAbstract	Components that must be turned on to run	ControlledCompAbstract
OperatedCompAbstract	Components with an opening/closing mechanism	ControlledCompAbstract
SourceAbstract	Generic source (electric or hydraulic)	ComponentAbstract
SourceControlledAbstract	Sources that are required by the control system to operate	SOurceAbstract, StartingCompAbstract
ConsumerAbstract	Generic component which consumes and propagates the flow	ComponentAbstract
NodeAbstract	Generic node with a k-out-of-n logic	ComponentAbstract

Table 5.1: Abstract electric components.

Generic Electric Components

Type name	Description	Parent type
ElecComp	Generic electric component	ComponentAbstract, FailureToRun
ElecLink	This class put other components on interface	-
ElecSource	This class represents a self-sustaining source	ElecComp , SourceAbstract
ElecSourceControlled	This class represents a controlled generic electric source	ElecComp, SourceControlledAbstract, FailureToStart
ElecConsumer	Generic electric consumer	ElecComp, ConsumerAbstract
ElecNode	Generic electric node	NodeAbstract, ElecConsumer

Table 5.2: Generic components.

Falure modes

Type Name	Parameters
`FailureToRun`	λ_{FR} : failure to run rate $MTTR_{\mathrm{FR}}$: Mean time to repair a failure to run
`FailureToRunShort`	λ_{FRS} : failure to run rate short $MTTR_{\mathrm{FRS}}$: Mean time to repair a failure to run short
`FailureToRunLong`	λ_{FRL} : failure to run rate short $MTTR_{\mathrm{FRL}}$: Mean time to repair a failure to run short
`FailureToOperateAbstract`	/
`FailureToOpen`	γ_{FO} : probability associated to the failure to open $MTTR_{\mathrm{FO}}$: Mean time to repair a failure to open
`FailureToClose`	γ_{FC} : probability associated to the failure to close $MTTR_{\mathrm{FC}}$: Mean time to repair a failure to close
`FailureToStart`	γ_{FS} : probability associated to the failure to start $MTTR_{\mathrm{FS}}$: Mean time to repair a failure to start
`FailureToStartShort`	γ_{FRL} : Probability associated to the failure to start short $MTTR_{\mathrm{FRL}}$: Mean time to repair a failure to start short
`FailureToStartLong`	γ_{FSL} : Probability associated to the failure to start long $MTTR_{\mathrm{FSL}}$: Mean time to repair a failure to start long

Table 5.3: Failure modes and associated parameters.

5.3.2 Real Components

The components that actually have been modeled and used in the application study are listed in tables below:

Type name	Description	Parent type
`TurboAlternator`	Alternator in power stations driven by the steam turbines	`Alternator`
`Alternator`	Electromechanical device that converts mechanical energy to electrical energy in the form of alternating current	`ElecSource`
`Transformer`	Static electrical device that transfers energy by inductive coupling between its winding circuits	`ElecConsumer`
`Breaker`	Electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another	`ElecConsumer, FailureToOpen, FailureToClose`
`Busbar`	Component of the electricity supply system which divides electrical power feed into subsidiary circuits	`ElecConsumer`
`Battery`	It guarantees electricity supply to busbar for a specified long period (2 hours, 12 hours or 24 hours)	`ElecSourceControlled`
`Diesel`	Combination of a diesel engine with an electric generator (often an alternator) to generate electrical energy	`ElecSourceControlled`
`Connection`	Connection points (unidirectional or bidirectional) called busses	`ElecConsumer`
`Substation`	Part of an electrical generation, transmission, and distribution system which transforms voltage from high to low or the reverse	`ElecConsumer`
`Grid`	Interconnected network for delivering electricity from suppliers to consumers	`ElecSource`

Table 5.4: Real components modeled in the knowledge base.

5.4 Preventive Maintenance Modeling

Maintenance actions have been implemented by creating groups of the maintained components. A maintenance group is a set of components which may be unavailable on an exclusive basis. At the beginning of each history, a component that may be unavailable because of maintenance is randomly selected from a uniform distribution (e.g. in a group 4 elements each of them has a chance of 25% to be selected). Then, a sampling is made

based on the probability of unavailability of the component due to maintenance. If maintenance occurs, the component will not be available to fulfill its mission for the whole duration of maintenance actions. As already mentioned, the unavailability for maintenance is taken into account in the definition of the generic component. All N components are of `ComponentAbstract` type , and each member of the group has a probability $1/N$ of being selected.

From the point of view of modeling, the parameters involved in modeling the process are listed in the table below :

Name	Description	Value
`duration_upm`	Duration time of a scheduled maintenance action (downtime)	real
`gamma_upm`	Probability of unavailability due to preventive maintenance	real

Table 5.5: Maintainance parameters.

5.5 Common Cause Failures Modeling

The common cause failures (CCF) have been implemented in CONCERTO to model system dynamics in terms of failures. In CONCERTO we used the **alpha factor model.** This is a multi-parametric model and the relationships between the alpha factor and the parameters β,γ and δ of the Multiple Greek Letter (MGL) model, which are the only available in the summary for reliability data of EPR PSA, are the following :

- a group with $N = 2$:

$\alpha_1^2 = 2(1-\beta)/(2-\beta)$

$\alpha_2^2 = \beta/(2-\beta)$

- a group with $N = 4$:

$\alpha_1^4 = 12(1-\beta)/(12-\beta)(6+\gamma(2+\delta))$

$\alpha_2^4 = 6\beta(1-\gamma)/(12-\beta)(6+\gamma(2+\delta))$

$\alpha_3^4 = 4\beta\gamma(1-\delta)/(12-\beta)(6+\gamma(2+\delta))$

$\alpha_4^4 = 3\beta\gamma\delta(12-\beta(6+\gamma(2+\delta))$

5.6 Instrumentation and Control Modeling

To modeling the control system which acts on the components of the EPR power distribution , the so-called **Compact Model** (CM) [14] has been taken as reference. Conceptually, the CM model of the I&C system consists of the following three basic parts :

1. Acquisition part
2. Common logic part
3. Operative part.

Concerning the first part, i.e. the acquisition, an object has been created called *Monitor*. The Monitor, as it may be inferred from the name itself, contains those functions of the I&C system which are related to the system state monitoring and particularly of its associated components. At the same time, an other object has been created aimed to perform a part of the processing logic functions, defined as *Controller.* Consequently, in line with the conceptual model of CM, a third object was designed that shares by means of inheritance properties the attributes of *Monitor* and *Controller*, and which is defined as *I&C*. Finally, it has been created *I&C_compact_model*, an object designed to be in relation with *I&C* by means of the same inheritance mechanism . When its common logic is faulty, this latter is in the unavailable state. It is shown below a scheme of the different parts of the I&C system that has been modeled :

Component	Type	Failure mode	Parameter
I&C Automaticl actuation (EDG)	EPR_IandCAutoEDG	Spurious operation	lambda_spo
I&C Automaticl actuation (EDG)	EPR_IandCAutoEDG	False acquisition	gamma_cm_fac
I&C Automaticl actuation (EDG)	EPR_IandCAutoEDG	Failure of specific logic	gamma_cm_fspl
I&C Automaticl actuation (EDG)	EPR_IandCAutoEDG	Number of acquisition trains	nb_ac_trains
I&C Automaticl actuation (EDG)	EPR_IandCAutoEDG	Number of specific logic trains	nb_spl_trains
I&C Manual actuation (SBO)	EPR_IandCManualSBO	Failure to send order	gamma_fso
I&C Manual actuation (SBO)	EPR_IandCManualSBO	Spurious operation	lambda_spo
I&C common logic (EDG)	EPR_EDG_CLCM	Failure of common logic	gamma_cm_fcl
I&C common logic (SBO)	EPR_SBO_CLCM	Failure of common logic	gamma_cm_fcl

Table 5.6: The scheme of the I&C model implemented in the kowledge base.

6 Application : EPR power supply modeling

6.1 Overview of the system modelling

The perimeter of the EPR electrical distribution model described in this document includes the following general subsystems :

- Connections to the 400kV grid : main and auxiliary lines;
- Interface of the main 400kV line with the turbo-alternator through the main line breakers and the main transformers;
- Interface of the 10kV distribution with the main 400kV line through the 400kV line breakers and both step-down transformers TS1 and TS2 along with the auxiliary line;
- 10kV busbars and associated Emergency Diesel Generators (EDG);
- 690V busbars and associated Station Black Out(SBO) diesel generators for line A and D.
- I&C signals allowing the following automatic and manual (from the control room) operations :
  - EDG starting and related breakers positioning;

– SBO starting and related breakers positioning.

The block diagram below illustrates the system boundaries included in the power supply distribution modeled for the case study.

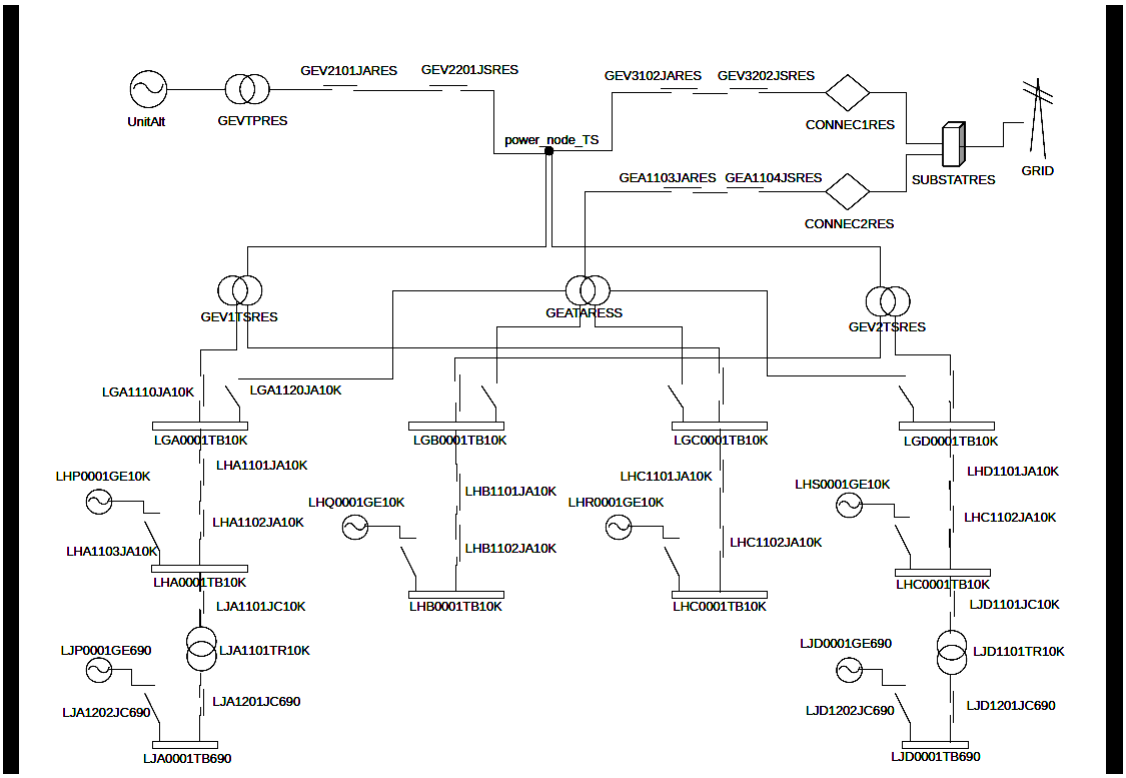


Figure 6.1.1: The figure shows the functional layout of the EPR power system that has been modeled for the study case.

The power supply system of the plant begins with the generation of electricity by the turbo-alternator `UnitAlt` which is connected to the main grid `Grid`. The components that make possible this connection are the main transformer `GEVTPRES` and two high voltage circuit breakers. These latters are in closed state in nominal conditions. The point where current flows between the plant and the main grid is represented by an electric node (which in reality is composed of specific components to which we are not interested in for the modeling) and that allow the main grid to provide in turn the auxiliary power in the case of failure of the turbo-alternator group. Then electric power

reaches the two step-down transformers `GEV1TSRES` and `GEV2TSRES` , consequently it is distributed to the four trains of the power supply system. In each electric train, the current provides power to the consumers through the busbars of the various series LG, LH and LJ. In particular, as shown by Figure 6.1.1 , the busbars type LH and LJ are connected to a system of emergency diesel generators (EDG) and last rescue (SBO) diesel generators which guarantee continuity of busbar power supply as long as it is required for the nominal operating conditions to be restored. In the dynamic model of the system shown in 6.1.1 the components that have been designed are (in parentheses specifies the number of components for each individual type:

- turbo-alternator group (1);
- transformer (6);
- breaker (32);
- busbar (10)
- diesel (6)
- grid (1) + substation (1) + connections (4). In the following Section 6.2 it will be discussed in more detail the system topology form the point of view of its subsystems and components.

6.2 System architecture

The EPR distribution model described here[21] is characterized by the following features:

- a distribution to 4 trains, from 1 to 4;
- four power levels, 400kV, 10kV, 690V and 220V;
- a main electric line (busbar LG, LI, LK) and a secondary line rescued by diesels (LH, LJ) one for each train.

The High-Voltage (HV)10kV power distribution of the EPR involves two different families of busbars on 4 trains :

- the busbar system LG, not rescued by diesels;
- the busbar system LH rescued by the EDG diesel.

6.2.1 External Sources

There are two types of external power sources feeding the 10kV distribution of the EPR :

- Main external source
- Auxiliary external source

6.2.1.1 Main external source

This primary source consists ofthe following components:

- the main grid 400kV
- the grid substation, that provides the interface between the main grid and the main grid 400kV and the power lines 400kV of the sites
- the main electric line 400kV (LP)
- the 400kV breakers of the main electric line (SLP, DPL)
- 2 step-down transformers (TS1) GEV1TSRES and (TS2) GEV2TSRES.

The main external source interfaces with the turbo-alternator through the main transformer and the circuit 400kV breakers. In normal conditions, when the reactor is in operation and the tranche is coupled to the main grid, the alternator operates as a generator and produces power supply to the electric system by means of the main transformer (TP) GEVTPRES. It also provides power to the auxiliary line through the main

transformer TP, the two step-down transformers TS1 and TS2, and the electric distribution system. In the case of loss of the main grid, with the reactor still running and the alternator connected, it produces the energy needed to power auxiliary line of the tranche through the TP, the two TS1 and TS2 and the distribution system. In this case the tranche is said to be in *House load operation* state [21]. This means that the main grid is unavailable and the main circuit breakers are open : then the system is capable to reduce the power from its nominal value 100% to 5% of it.

6.2.1.2 Auxiliary external source

This secondary source is constituted by the following components:

- the main grid 400kV
- the grid substation, that provides the interface between the main grid and the main grid 400kV and the power lines 400kV of the sites
- the auxiliary line 400kV (LA)
- the 400kV breakers of the auxiliary electric line (SDA, DLA)
- the auxiliary transformer (TA) GEATARESS

This source comes to the rescue of the main line when it is lost. In this case, the permutation of power source is carried out by means of a switching device in order to ensure a continuity of supply for the internal electric distribution of the tranche.

6.2.2 Internal Sources

Two families of internal sources defined in relation to the HV 10kV and LV electrical distribution have been modeled:

- the batteries

- the diesel generators.

In the following, a brief description of both types of internal source will be provided.

6.2.2.1 Batteries

The batteries ensure a temporarily feeding of low voltage busbars aiming to power I&C devices involved inthe following situations :

- loss of the main power source : switch LG busbars feeding from main to auxiliary line;
- loss of external power sources : start-up and coupling mechanism of diesel generators

The following two types of batteries are used in the EPR unit :

- 4 batteries LAA, LAB, LAC, LAD of autonomy equal to 2 hours : one for each of the 4 divisions of the nuclear island to start the corresponding diesel and SBO if needed;
- batteries of autonomy equal to 12 hours, one for each of the division 1 and 4 (LVP, LVS). These latters are not modeled in the konwledge base..

In nominal conditions, the batteries do not generate any current except in occasional points. However, in case of lack of tension upstream of the relative busbars, the battery alone ensures the supply of energy. .

The battery of trains A,B and trains C,D are diversified.

6.2.2.2 Diesels

There are two types of diesel generators:

- 4 Emergency Diesel Generators (EDG), distributed within the 4 divisions (LHP, LHQ, LHR, LHS), providing an emergency power supply to the 10kV busbar LH (LHA, LHB, LHC, LHD);
- 2 Station Blackout (SBO) diesel generators, one for each of the the division 1 and 4 (LJP, LJS), providing an emergency power supply to the main 690V busbar LJ.

As will be explained in more detail in the following, the operation of the EDG and SBO diesels has been modeled as they operate in the same way, even if this is not really the case (they are required in different situations) . The diesel generators 10kV EDG (LHP-Q-R-S), installed one for each division, refeed all consumers of the busbars LHA-B-C-D, respectively, in case of simultaneous loss of main grid, auxiliary line and alternator. EDG of each division may equally be started in case of a fault on the lines coming from the normal operating busbars LGA-B-C-D. In the case of total loss of power supplies, i.e. simultaneous loss of external sources, alternator and the four diesels EDG, the two diesel generators SBO (LJP and LJS) are required to feed busbar 690V LJA0001TB690 and LJD0001TB690.

6.2.3 High Voltage 10kV busbars (LG, LH)

The main role of 10kV busbar is to power all consumers of the nuclear island and those of the conventional one whose power is greater than 500kW. The power supply of the auxiliary line LH involves external sources provided from the main grid and internal sources represented by the main diesels EDG. The 10kV busbars LG (LGA0001TB10K, LGB0001TB10K, LGC0001TB10K and LGD0001TB10K) feed the distribution system of HV 10kV AC of the busbars LH (LHA0001TB10K, LHB0001TB10K, LHC0001TB10K and LHD0001TB10K) and the LV 690V AC system of the busbar (LJ) LJA0001TB690 and LJD0001TB690 (only train 1 and 4). The 10kV busbars LH instead are rescued by the EDGs and provide electricity to the LV 690V LJ system.

6.2.3.1 The LG busbar system

The main busbars (LGA-B-C-D) ensure the function of switching the power source TS / TA. They have two distinct pathways:

- normal power supply through one of the two TS;
- emergency power supply through the TA.

In both cases the power supply is guaranteed by the closure of 10kV breakers located upstream and downstream of for each busbar LG on the 4 trains.

6.2.3.2 The LH busbar system

This group of busbars contains the set of materials that allow the distribution of electricity 10kV, rescued by EDG diesels, e.g. LHP0001GE10K on the first train, and designed to supply consumers with the needs of high availability or included in safety functions. They ensure the power supply to 690V busbars LJA-B-C-D.

6.2.4 Low Voltage 690V busbars (LJ)

Two 690V busbar, LJA0001TB690 and LJD0001TB690, are installed on train 1 and 4, respectively. They are rescued by means of the diesels generators EDG through the main busbar LH upstream, while in the case of fault of the upstream line they are also rescued by the SBO diesel generators LJP0001GE690 and LJD0001GE690.

6.3 I&C with Compact model

The elements that were modeled for the I & C are the following :

1. Voltage sensor on busbars LHX;
2. Signal for the automatic diesel start-up and the opening / closing of the associated 10kV breakers ;

3. Signal for the manual start-up of the SBO from the main control room when a loss of voltage on all four busbars LH occurs.

6.4 Common Cause Failures groups

Considering the effects of propagation of common cause failures (CCF) among similar and symmetrical components with respect to the four electric trains, common cause component groups have been created that suffer the propagation mentioned above. Components within a CCF group were considered to have similar properties and on the basis of studies already carried out by the operating division of EDF CNEN, the likelihood of cascading failures or simultaneous failure of order greater than one[1] was found to be not negligible in order toevaluate the CDF of the LOOP initiator event. Consequently, for the considered case study, 4 groups of components were modeled : two groups of order 2, and others two groups of order 4. These CCF groups are described as below :

The CCF groups that have been considered in the model are listed here below :

- EDG: FRS, FRL, FSS, SLF . These are 4 groups of order 4;
- SBO: FRS, FRL, FSS, SLF . These are 4 goups of order 2;
- 10kV Breakers: FO, FC . These are 2 groups of order 4
- Breakers 690V: FO, FC. These are 2 groups of order 2
- I&C: Failure of Acquisition, Send spurious order. These are 2 groups of order 4 (start-up EDG) two groups of order 2 (start-up SBO)

[1] In a common cause components group, the order has the meaning of number of components that can fail simultaneously. For example, in a group of order 4, there is a finite probability that 4 components may fail simultaneously, because of their similarity and so all subject to the propagation of the common cause failure.

6.4.1 10kV circuit breakers

This is an order 4 component group composed of the breakers upstream the LH busbars. From train 1 to 4, these are LHA1102JA10K, LHB1102JA10K, LHC1102JA10K and LHD1102JA10K, respectively. In nominal conditions, they are in the closed state to provide power to the busbars LH series.

6.4.2 690V circuit breakers

This group is made of two line breakers only, LJA1201JC690 and LJD1201JC690, connected to busbars LJA0001TB690 and LJD0001TB690, respectively. As in the prevous group, these breakers are initially closed to power the busbar LJ downstream.

6.4.3 Main diesel generators EDG

This group is composed of the four Emergency Diesel Generators connected to the LH busbars series, from train 1 to 4. They are LHP0001GE10K, LHQ0001GE10K, LHR0001GE10K and LHS0001GE10K. At the initial conditions, the EDGs are turned off and the breakers placed on the link between diesels and busbars are in the open state.

6.4.4 Diesels SBO

This group is composed of the two Station Blackout diesels connected to the LJ busbars series for train 1 and 4 only. At the initial conditions, the SBO diesel are turned off and the breakers placed on the link between diesels and busbars are in the open state.

6.5 Maintenance groups

With regard to preventive maintenance actions that are routinely performed for various systems of the EPR, groups of components have been considered on which these actions can be performed. Maintenance is possible for one component only at a time among those

belonging to the group. This selected component will be no longer available, and therefore it will not be subject to any operation by the control system during the maintenance time period. In particular, it has been assumed that maintenance is performed on all diesels generators EDG and SBO, also in agreement with what has been required by CNEN for the study of the situations of interest. Then there are only one group consisting of the 4 EDG and the 2 SBO.

6.6 Reliability data

In the base of the facts which constitute the modeling of the system from point of view of configuration and arrangement of system components and the values that the variables take in the specific case study, data collected by EDF of the installed components and equipments of the EPR FLA3 power supply system were used as reference to give numerical consistency to the model in order to test it by Monte Carlo simulation. Nevertheless, for reason of confidentiality the calculations have been performed not really using these data but others values were adopted being close to reality as well. So, failure rates as well as probabilites of occurrence were taken with strict adherence to reality although not necessarily identical to those found in the official documentation provided by the CNEN.

7 Case study : Loss of Offsite Power Risk Assessment

In this Chapter the results obtained from simulation data analysis of the case study described in Chapter 6 will be presented and the most relevant situations of interest demanded to study by the CNEN engineering division of EDF will be discussed. In Section 7.1 , generic features of the case study scenario are presented focusing mainly on the typical accident sequences from the loss of offsite power in state A and B. Then, in Section 7.2the system assumptions onthe Monte Carlo technique are reminded for the characterization of the simulation from a dynamic PSA point of view. Then, in Section 7.3 a discussion on the main outcomes arising from the simulation are presented with reference to the FLA 3 case.

7.1 Case study scenario

Loss Of Offsite Power (LOOP) has been already defined in Chapter 3 as loss of both the main and auxiliary grid connections, but a resume of the case study context is provided in this section. The analysis carried out in this work covers also the so-called Station Black Out (SBO) situation which is defined as a Loss Of Offsite Power with the occurrence of low voltage on the four 10kV safety busbars LHA, LHB, LHC, LHD, each one backed up by an Emergency Diesel Generators (EDG).

Reminding to the reader that this apply for each initiating event of the LOOP group,

the typical accident sequences from the LOOP in state A and B are:

- the Emergency Diesel Generators are automatically started when a LOOP has occurred, while the Station Blackout Diesels are usually started by the operator from the Main Control Room (MCR) after the loss of all Emergency Diesel Generators. Both starts require the availability of the 220V batterieswhichgarantee uninterrupted power supply [19].

- the reloading sequence, which follows the start of the Emergency Diesel Generators, maintains the power supply to the safety trains and their support systems. The following systems are supported by this action: the Emergency Feedwater System , the Component Cooling Water System , and the Essential Service Water System , the Chemical and Volume Control System, the Safety Injection System and the Containment Heat Removal System.

- if the batteries fail, the Station Blackout Diesel Generators can be started manually by local action. This backup is only considered when the batteries fail in operation after a long time window.

As it has been said in other elsewhere in this thesis, the aim of the study concerns with the analysis of specific situations of risk with respect to which some statistical indicators have been evaluated by Monte Carlo simulation . However, the definition of these indicators for the probabilistic risk assessment of the consequences deriving from LOOP initiating events arised from the specific requests demanded by the CNEN and not merely from the need of testing the model itself. So, it was necessary to take into account certain aspects of the most intersting accident dynamics and go deep into the complexity of the model precisely in order to fulfill the required study demanded by CNEN. Entering into the details of the study, the simulation was aimed to analyze the following situations of interest :

- **Situation 1 (S1)**: loss of four EDGs LHP QHL, LHR, LHS ;

- **Situation 2 (S2)** : S1 + 2 loss SBO LJP, LJS;
- **Situation 3 (S3)** : loss of at least three EDGs;
- **Situation 4 (S4)** : S3 + loss of at least one SBO.

The indicators which have been estimated for each situation are listed below :

- $P\left(T_{Si} \leq 24h \mid LOOPL\right)$: probability that occurence time of situationSi is before the mission time T^M, give that the LOOPL event has occurred;
- $E[T_{Si} \mid LOOPL, T_{Si} \leq 24h](h)$: mean time to situation Si, given that the LOOPL event occurred and situation i has occurred before T^M
- $E\left[D_{Si} \mid LOOPL, T_{Si} \leq 24h\right](h)$: mean duration time of situation Si, given that the LOOPL event occurred and situation Si has occurred before T^M .

7.2 Quantification via Monte Carlo simulation

7.2.1 Introduction

In technical systems like nuclear power plants, an accident sequence starts with an initiating event and evolves over time through the interaction of dynamics and stochastics. This interaction is capable of producing infinitely many different sequences. Along the time line, they define a continuous dynamic event tree with infinitely many branch points. At each point of time, the stochastic variability of the accident consequences is summarized by a multivariate probability distribution. A probabilistic safety analysis (PSA) requires an approximation of this distribution for selected consequence variables. It is felt that the conventional event tree analysis of Level 1 and of Level 2 PSA does often not permit a satisfactory probabilistic representation for dynamic PSA purposes. For this reason various methods of dynamic PSA have been suggested over the past decade. In this work we have combined the developement of a dynamic model of the system with an

analysis with Monte Carlo simulation. The advantages of this combination are explained and illustrated in the next sections.

7.2.2 Monte-Carlo Simulation for Dynamic PSA

Event sequences evolve over time through the interaction of dynamics and stochastics in the system [18]. In the conventional event tree analysis of Level 1 PSA and in the accident sequence analysis of Level 2 PSA, the analysts prescribe the stochastic events together with the order in which they occur. While temporal information may be available for few selected sequences in Level 1 and on an event scale in Level 2, it is usually not given for the tree. These customary trees largely develop along a so called effect line rather than a time line. Branch points of Level 1 event trees are prescribed by the order of safety system demands at set points. Usually, there are only two branches per point, namely "system starts" and "system fails to start". Due to limitations of the conventional event tree methodology no consideration can be given, for instance, to the consequences of failure to run for the intended time and/or with the required capacity. Branchings in Level 2 accident sequence trees are frequently used to account for the stochastic variability of the consequence magnitudes from phenomena that do not permit mechanistic modeling. Presently, the multitude of possible accident sequences in Level 2 PSA is reduced to the degrees of freedom of a rather coarse grid in time (i.e. "early", "late" or "before", "after"), in space (i.e. "top", "bottom") and in magnitude (i.e. "small", "medium", "large"), etc. This does not permit to model the possible spectrum of interactions of dynamics, phenomena, component behaviour and human actions as close to reality as is desirable. Inherent to this coarse grid is the danger that important sequences, resulting from details in time, space, magnitude and order of events, remain unknown unrealistic sequences are generated, based on analyst specified conditions which otherwise would result from preceding events.

Probabilistic dynamics enable us to fully account for the temporal evolution of the

interaction of dynamics and stochastics in the evaluation of accident consequences and their conditional (condition is the initiating event) probabilities. Probabilistic dynamics operates on the actual time/state space although discretizations have to be performed for the evaluations of indicators to be numerically manageable. The computational effort is considerably larger compared to a conventional event tree analysis. For this reason, their application is still restricted to specific aspects of a PSA. The vision is, however, to be able to perform a dynamic PSA. The most straightforward numerical procedure for such an analysis would be a Monte Carlo simulation. Its transition probabilities may depend on the state of the dynamic quantities, systems and components and even on residence times as well as on details of the sequence history. It suffices to prescribe rules for the evaluation of the probabilities of transitions to those states that are directly accessible from the present state. One Monte Carlo element generates only one sequence out of the population of infinitely many possible sequences and the Monte Carlo simulation produces a random sample of sequences. Low probability transitions will be adequately represented only if the sample is of sufficiently large size. The generation of each sequence requires a complete dynamics calculation starting from the initiating event and ending in one of the "absorbing" states. The latter include specified damage states, e.g. the station black out for an electrical network, states of no damage and controlled operation (safety states) and possibly the arrival at the endpoint of the specified observation time (mission time).

7.2.3 YAMS Software Overview

The underlying idea of Monte Carlo simulation is to simulate a number of histories or trials (if we comparate the simulation to a dart game [18] of duration T. For each trial the state of the system at the various times ($<$T) is observed and for which some performance indicators are calculated. The software that allows the integration of the Monte Carlo simulation in the automatic chain of the PRA studies designed around

the platform KB3 is the operating software YAMS [5, 17]. YAMS is a simulator that allows to extend the application domaine to the systems that follow a Markov process with a large combinatorial explosion as well as systems with a non-Markovian behavior. The originality of this simulation tool is that it allows unrestricted use of the powerful modeling language FIGARO. Its capacity in defining very diverse performance indicators without the need to overload the model make it very utilized in the Industrial Risk Management department at EDF R&D. For example it very easy to do the following calculations:

- evaluate the integral of a variable over a period of time,
- estimate the time spent in a category of states,
- detect the first entry in a category of states after a given time or after the entry into another category of states.

The description in YAMS treatments is based on two types of data, as described in the following.

7.2.3.1 Main data parameterization of treatments :

- Mission time and memorization points of each history. This data defines the time interval on which make the simulations as well as moments for which indicators will be calculated to be stored.
- Maximum number of simulated histories.

7.2.3.2 Data defining the desired results :

- List of states. A state is defined as a boolean expression of FIGARO type , which implements the system variables. A state may be defined as target. In this case, the state is considered absorbent by YAMS , even if it is not based on the behavior of the model. Otherwise, the state could be described as a state of analysis.

These two categories of states are respectively adapted to reliability calculations (by optimizing the calculation time) and availability.

- List of indicators. Each indicator is the result of a function (e.g. TEMPS_DE_SEJOUR, DEJA_REALISE, INTEGRAL...) applied to each FIGARO expression.

7.3 Flamaville 3 quantification results

In this section the results achieved during the Monte Carlo simulation of the model implemented by CONCERTO are presented and discussed. All the results shown in the next paragraphs have been estimated working in the following case study scenario :

- Long-term LOOP (LOOPL) initiating event;
- Mission time of 24h;
- Reliability data associated to EPR FLA3.

7.3.1 Situation 1

7.3.1.1 Indicators

The PRA performed via Monte Carlo simulation for the first situation (S1) has conducted to the results which have been summerized as below :

- Table 7.1 provides the mean estimation of the desired indicators.
- Values for the probability distribution and the cumulative distribution respectively of occurrence time associated to situation 1 are depicted in Figures 7.3.1 and 7.3.2 respectively.
- Values for the probability distribution and the cumulative distribution respectively of situation 1 duration. are depicted in Figures **??** and 7.3.4 respectively.

Indicators	Value	CI99%
$P\left(T_{S1} \leq 24h \mid LOOPL\right)$	6.4E-03	2.0E-04
$E[T_{S1} \mid LOOPL, T_{S1} \leq 24h](h)$	15.4	0.3
$E\left[D_{S2} \mid LOOPL, T_{S1} \leq 24h\right]](h)$	7.5	0.3

Table 7.1: Indicators estimation associated to situation 1 after LOOPL initiating event over 24h for EPR FLA3.

Mean indicators

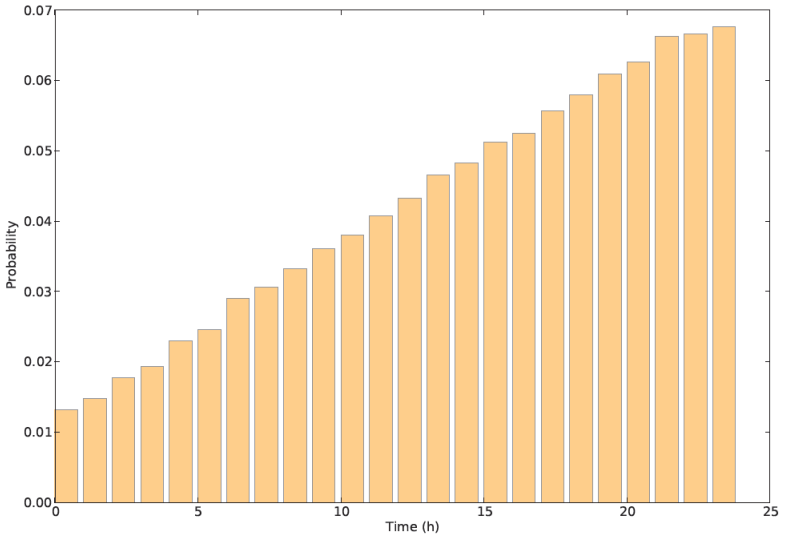


Figure 7.3.1: Probability distribution of S1 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

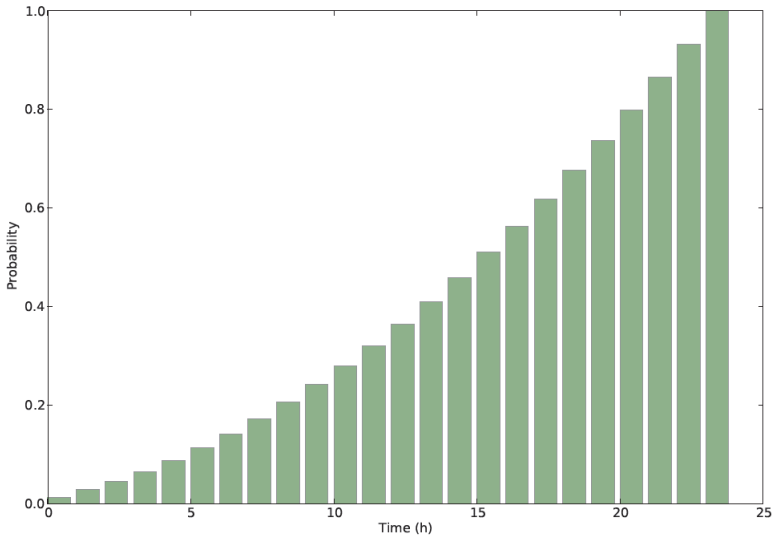


Figure 7.3.2: Cumulative distribution of S1 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation occurrence time e quindi niente.

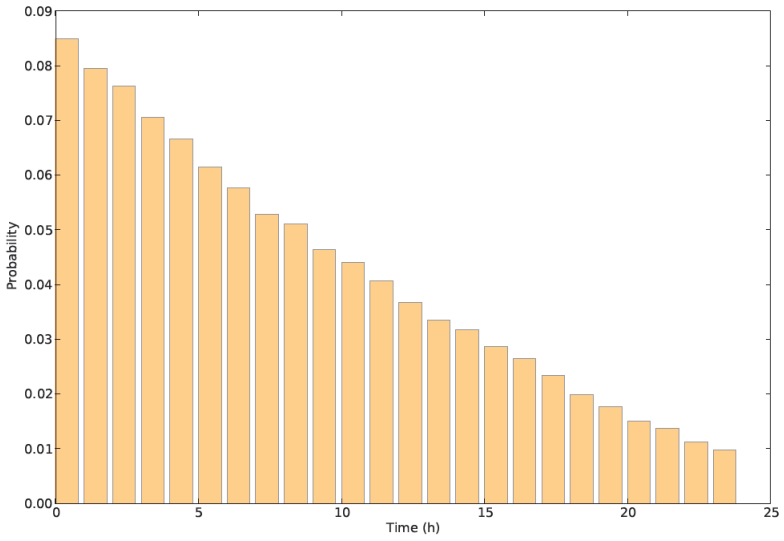


Figure 7.3.3: bla bla

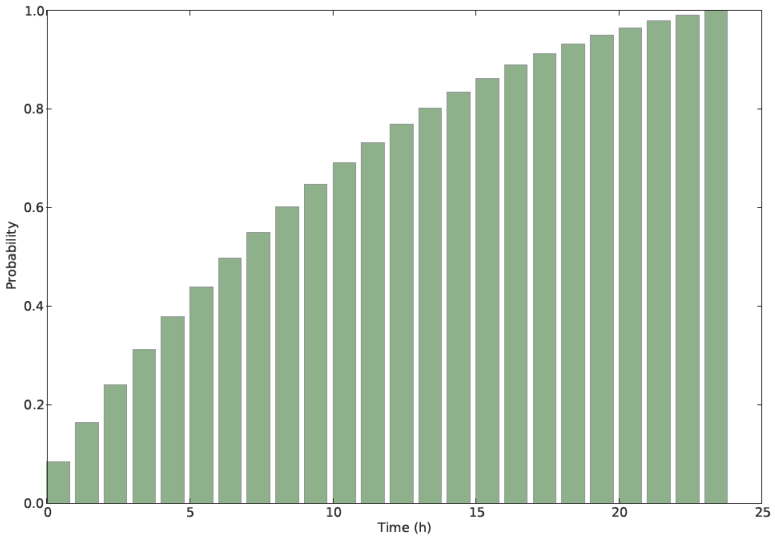


Figure 7.3.4: Cumulative distribution of S1 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation duration time

7.3.1.2 First sequences

N°	Value	Contrib.	Contrib. cum.	Sequence of transitions
0	1.900E-06	0.9%	0.9%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.84) CC4LH0001GE10K_FRL_1_to_3_4 LHP0001GE10K_FRL LHS0001GE10K_FRL LHR0001GE10K_FRL [(8.84) LAA1101BT_STE LAD1101BT_STE LAC1101BT_STE [(14.74) LHQ0001GE10K_FRL
1	1.900E-06	0.86%	1.7%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(5.53) CC4LH0001GE10K_FRL_1_to_4 LHP0001GE10K_FRL LHS0001GE10K_FRL [(7.53) LAA1101BT_STE LAD1101BT_STE [(15.76) CC4LH0001GE10K_FRL_2_to_3 LHQ0001GE10K_FRL LHR0001GE10K_FRL
2	1.700E-06	0.8%	2.5%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.47) CC4LH0001GE10K_FRL_4_to_1 LHP0001GE10K_FRL LHS0001GE10K_FRL [(8.47) LAA1101BT_STE LAD1101BT_STE [(15.40) CC4LH0001GE10K_FRL_2_to_3 LHQ0001GE10K_FRL LHR0001GE10K_FRL]
3	1.700E-06	0.8%	3.3%	[(0.00) GRID_FRL UnitTurbine_FSHO [(5.45) CC4LH0001GE10K_FRL_4_to_1_3 LHP0001GE10K_FRL LHS0001GE10K_FRL LHR0001GE10K_FRL [(7.45) LAA1101BT_STE LAD1101BT_STE LAC1101BT_STE [(16.24) LHQ0001GE10K_FRL]
4	1.600E-06	0.7%	4.0%	[(0.00) GRID_FRL UnitTurbine_FSHO [(7.76) CC4LH0001GE10K_FRL_4_to_1_3 LHP0001GE10K_FRL LHS0001GE10K_FRL LHR0001GE10K_FRL [(9.76) LAA1101BT_STE LAD1101BT_STE LAC1101BT_STE [(16.60) LHQ0001GE10K_FRL]
5	1.600E-06	0.7%	4.7%	[(0.00) GRID_FRL UnitTurbine_FSHO [(7.08) CC4LH0001GE10K_FRL_2_to_1_4 LHP0001GE10K_FRL LHS0001GE10K_FRL LHQ0001GE10K_FRL [(9.08) LAA1101BT_STE LAD1101BT_STE LAB1101BT_STE [(14.92) LHR0001GE10K_FRL]
6	1.600E-06	0.7%	5.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.05) CC4LH0001GE10K_FRL_1_to_4 LHP0001GE10K_FRL LHS0001GE10K_FRL [(8.05) LAA1101BT_STE LAD1101BT_STE [(15.60) CC4LH0001GE10K_FRL_3_to_2 LHQ0001GE10K_FRL LHR0001GE10K_FRL]
7	1.600E-06	0.7%	6.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.92) CC4LH0001GE10K_FRL_4_to_1 LHP0001GE10K_FRL LHS0001GE10K_FRL [(8.92) LAA1101BT_STE LAD1101BT_STE [(16.92) CC4LH0001GE10K_FRL_3_to_2 LHQ0001GE10K_FRL LHR0001GE10K_FRL]
8	1.500E-06	0.78%	6.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.82) CC4LH0001GE10K_FRL_4_to_1 LHP0001GE10K_FRL LHS0001GE10K_FRL [(8.82) LAA1101BT_STE LAD1101BT_STE [(15.34) CC4LH0001GE10K_FRL_2_to_3 LHQ0001GE10K_FRL LHR0001GE10K_FRL]
9	1.500E-06	0.7%	7.5%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(7.62) CC4LH0001GE10K_FRL_2_to_1_4 LHP0001GE10K_FRL LHS0001GE10K_FRL LHQ0001GE10K_FRL [(9.62) LAA1101BT_STE LAD1101BT_STE LAB1101BT_STE [(18.56) LHR0001GE10K_FRL]
10	1.500E-06	0.68%	8.20%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(6.57) CC4LH0001GE10K_FRL_4_to_1 LHP0001GE10K_FRL

7.3.2 Situation 2

7.3.2.1 Indicators

The probabilistic assessment of situation 2 gives the following results :

- Table 7.3 provides the mean estimation of the desired indicators.
- Values for the estimation of the probability distribution and the cumulative distribution respectively of occurrence time associated to situation 2 are depicted in Figures 7.3.5and 7.3.6respectively.
- Values for the estimation of the probability distribution and the cumulative distribution respectively of situation 2 duration time are depicted in Figures 7.3.7 and 7.3.8 respectively.

Indicators	Value	CI99%
$P\left(T_{S1} \leq 24h \mid LOOPL\right)$	1.3E-02	2.9E-04
$E[T_{S1} \mid LOOPL, T_{S1} \leq 24h](h)$	15.1	0.1
$E\left[D_{S2} \mid LOOPL, T_{S1} \leq 24h\right]](h)$	7.9	0.1

Table 7.3: Indicators estimation associated to situation 3 after LOOPL initiating event over 24h for EPR FLA3.

Mean indicators

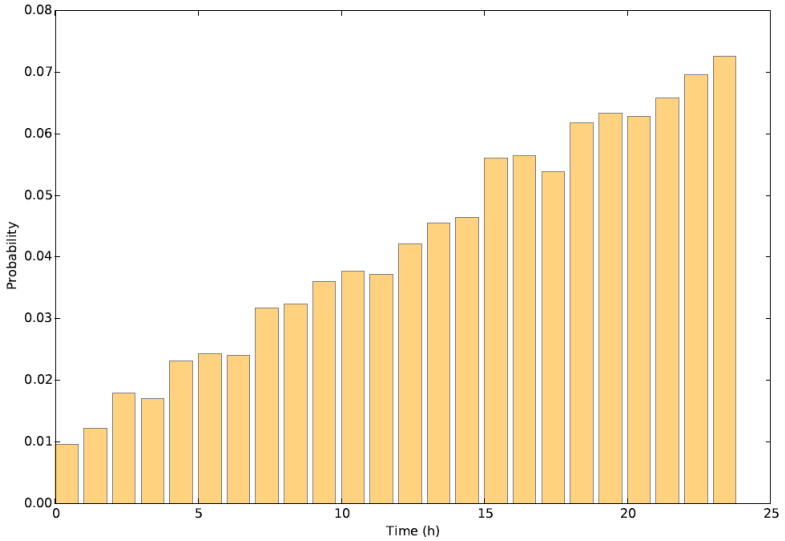


Figure 7.3.5: Probability distribution of S3 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

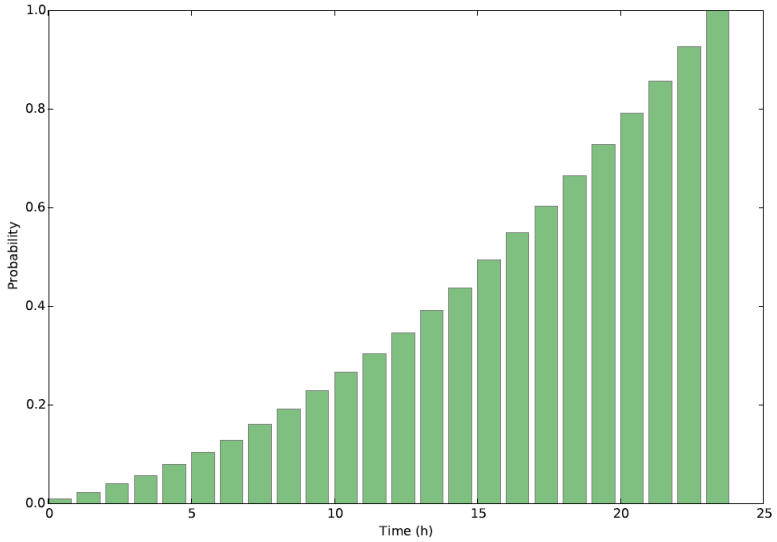


Figure 7.3.6: Cumulative distribution of S3 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation occurrence time

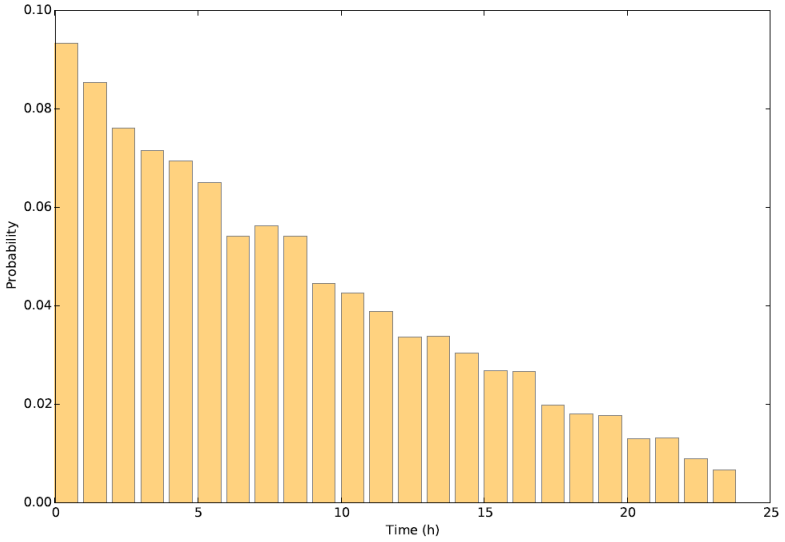


Figure 7.3.7: Probability distribution of S3 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

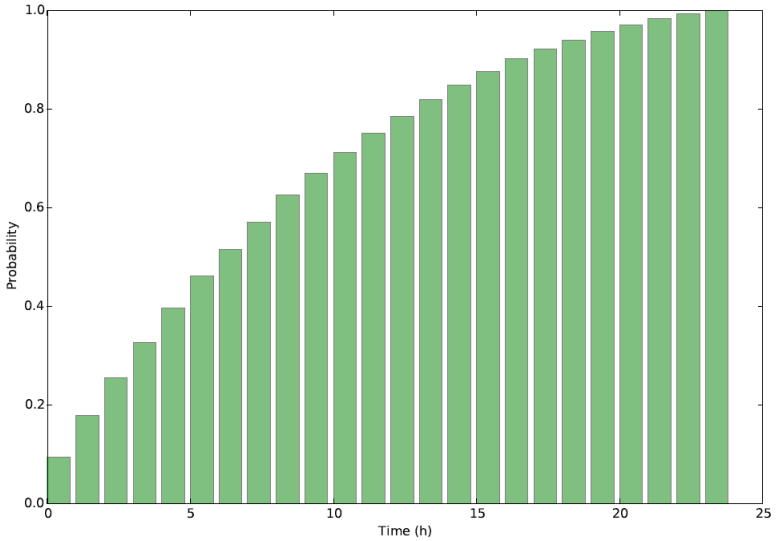


Figure 7.3.8: Cumulative distribution of S3 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation duration time

7.3.2.2 First sequences

N°	Value	Contrib.	Contrib. cum.	Sequence of transitions
0	3.2E-05	0.4%	0.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.06) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
1	3.14E-05	0.4%	0.9%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.69) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
2	3.13E-05	0.4%	1.3%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(9.85) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
3	3.11E-05	0.4%	1.7%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.76) CC4LH0001GE10K_FRL_2_to_1_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
4	3.100E-05	0.4%	2.15%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.94) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
5	3.100E-05	0.4%	2.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.48) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
6	3.050E-05	0.4%	2.3%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.21) CC4LH0001GE10K_FRL_1_to_2_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
7	3.010E-05	0.4%	3.40%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.15) CC4LH0001GE10K_FRL_4_to_1_2 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
8	3.010E-05	0.4%	3.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.33) CC4LH0001GE10K_FRL_2_to_1_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
9	3.010E-05	0.4%	4.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.48) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
10	3.000E-05	0.4%	4.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.04) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
11	2.960E-05	0.41%	5.0%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.29) CC4LH0001GE10K_FRL_4_to_1_2 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
12	2.940E-05	0.4%	5.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.40) CC4LH0001GE10K_FRL_1_to_2_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
13	2.940E-05	0.4%	5.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.77) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
14	2.940E-05	0.40%	6.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.01) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
15	2.940E-05	0.4%	6.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.74) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
16	2.930E-05	0.4%	7.05%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.75) CC4LH0001GE10K_FRL_3_to_1_2 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
17	2.92E-05	0.4%	7.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.89)

7.3.3 Situation 3

7.3.3.1 Indicators

The probabilistic assessment of situation 3 gives the following results :

- Table 7.5 provides the mean estimation of the desired indicators.
- Values for the estimation of the probability distribution and the cumulative distribution respectively of occurrence time associated to situation 3 are depicted in Figures 7.3.9 and 7.3.10 respectively.
- Values for the estimation of the probability distribution and the cumulative distribution respectively of situation 3 duration time are depicted in Figures 7.3.11 and 7.3.12 respectively.

Indicators	Value	CI99%
$P\left(T_{S1} \leq 24h \mid LOOPL\right)$	1.3E-02	2.9E-04
$E[T_{S1} \mid LOOPL, T_{S1} \leq 24h](h)$	15.1	0.1
$E\left[D_{S2} \mid LOOPL, T_{S1} \leq 24h\right]](h)$	7.9	0.1

Table 7.5: Indicators estimation associated to situation 3 after LOOPL initiating event over 24h for EPR FLA3.

Mean indicators

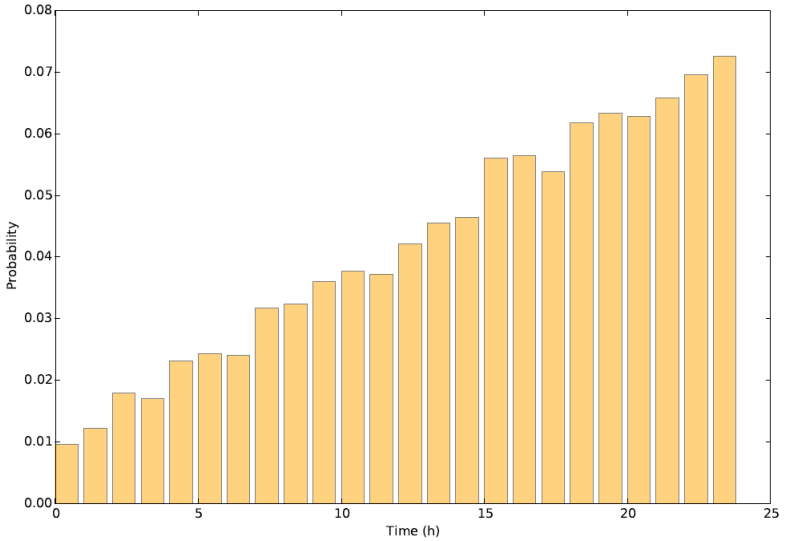


Figure 7.3.9: Probability distribution of S3 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

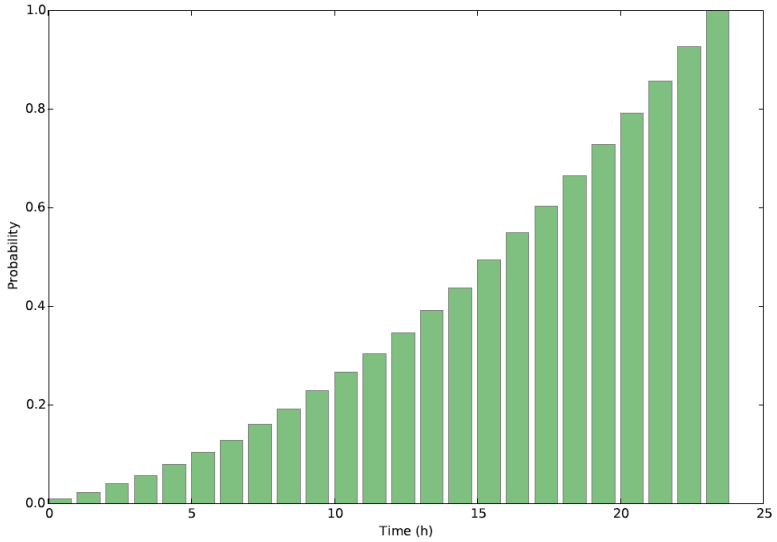


Figure 7.3.10: Cumulative distribution of S3 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation occurrence time

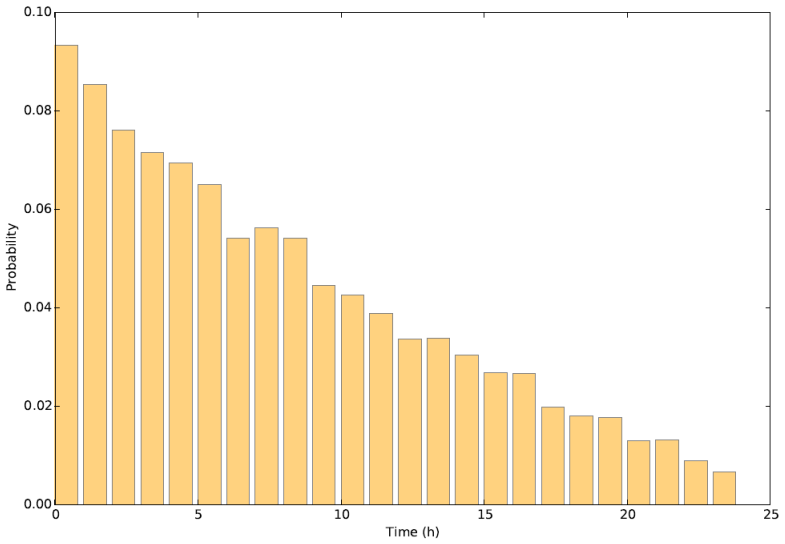


Figure 7.3.11: Probability distribution of S3 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

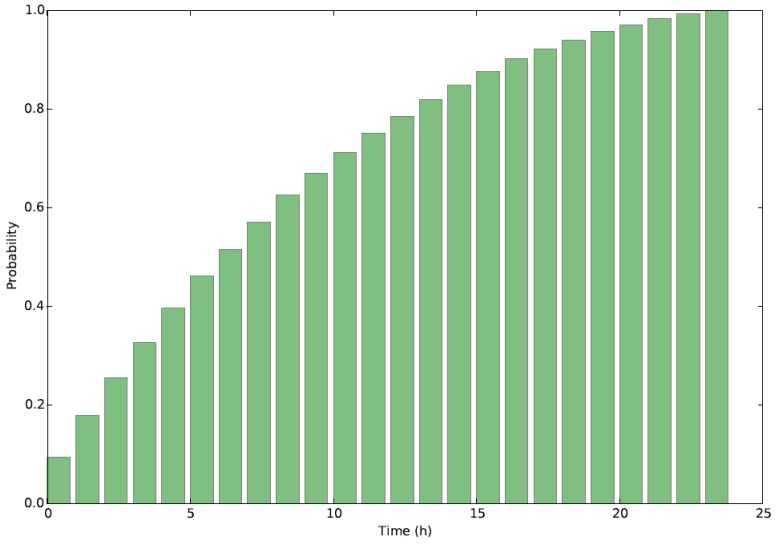


Figure 7.3.12: Cumulative distribution of S3 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation duration time

7.3.3.2 First sequences

N°	Value	Contrib.	Contrib. cum.	Sequence of transitions
0	3.2E-05	0.4%	0.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.06) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
1	3.14E-05	0.4%	0.9%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.69) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
2	3.13E-05	0.4%	1.3%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(9.85) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
3	3.11E-05	0.4%	1.7%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.76) CC4LH0001GE10K_FRL_2_to_1_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
4	3.100E-05	0.4%	2.15%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.94) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
5	3.100E-05	0.4%	2.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.48) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
6	3.050E-05	0.4%	2.3%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.21) CC4LH0001GE10K_FRL_1_to_2_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
7	3.010E-05	0.4%	3.40%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.15) CC4LH0001GE10K_FRL_4_to_1_2 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
8	3.010E-05	0.4%	3.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.33) CC4LH0001GE10K_FRL_2_to_1_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
9	3.010E-05	0.4%	4.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.48) CC4LH0001GE10K_FRL_1_to_3_4 LHS0001GE10K_FRL LHP0001GE10K_FRL LHR0001GE10K_FRL]
10	3.000E-05	0.4%	4.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.04) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
11	2.960E-05	0.41%	5.0%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.29) CC4LH0001GE10K_FRL_4_to_1_2 LHS0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
12	2.940E-05	0.4%	5.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.40) CC4LH0001GE10K_FRL_1_to_2_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
13	2.940E-05	0.4%	5.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.77) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
14	2.940E-05	0.40%	6.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.01) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
15	2.940E-05	0.4%	6.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.74) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
16	2.930E-05	0.4%	7.05%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.75) CC4LH0001GE10K_FRL_3_to_1_2 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
17	2.92E-05	0.4%	7.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.89)

7.3.4 Situation 4

7.3.4.1 Indicators

The probabilistic assessment of situation 4 gives the following results :

- Table 7.7provides the mean estimation of the desired indicators.
- Values for the estimation of the probability distribution and the cumulative distribution of occurrence time associated to situation 4 are depicted in Figures 7.3.13 and 7.3.14respectively.
- Values for the estimation of the probability distribution and the cumulative distribution of situation 4 duration are depicted in Figures 7.3.15 and 7.3.16 respectively.

Indicators	Value	CI99%
$P\left(T_{S1} \leq 24h \mid LOOPL\right)$	7.3E-03	2.1E-04
$E[T_{S1} \mid LOOPL, T_{S1} \leq 24h](h)$	57.2	0.2
$E\left[D_{S2} \mid LOOPL, T_{S1} \leq 24h\right](h)$	7.8	0.2

Table 7.7: Indicators estimation associated to situation 4 after LOOPL initiating event over 24h for EPR FLA3.

Mean indicators

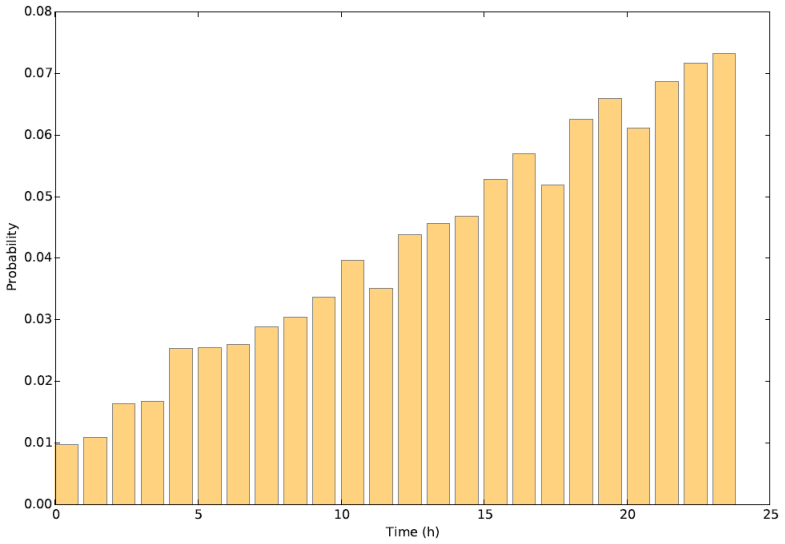


Figure 7.3.13: Probability distribution of S4 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

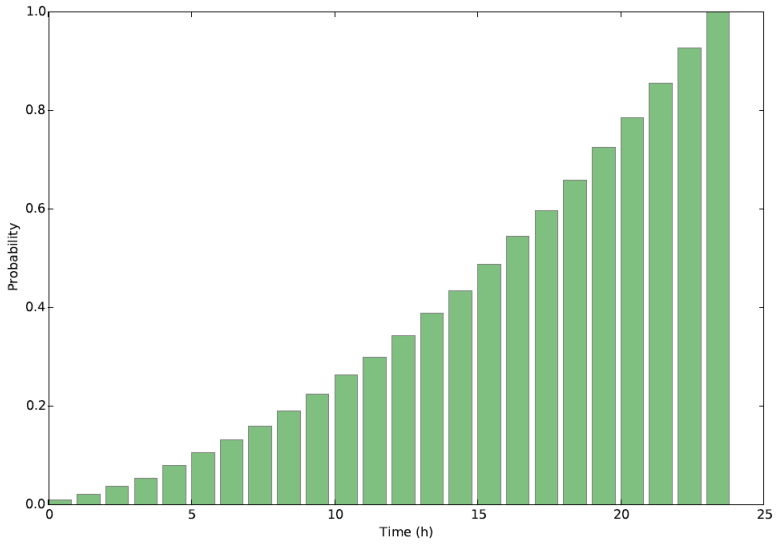


Figure 7.3.14: Cumulative distribution of S4 occurrence time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Distribution of situation occurrence time

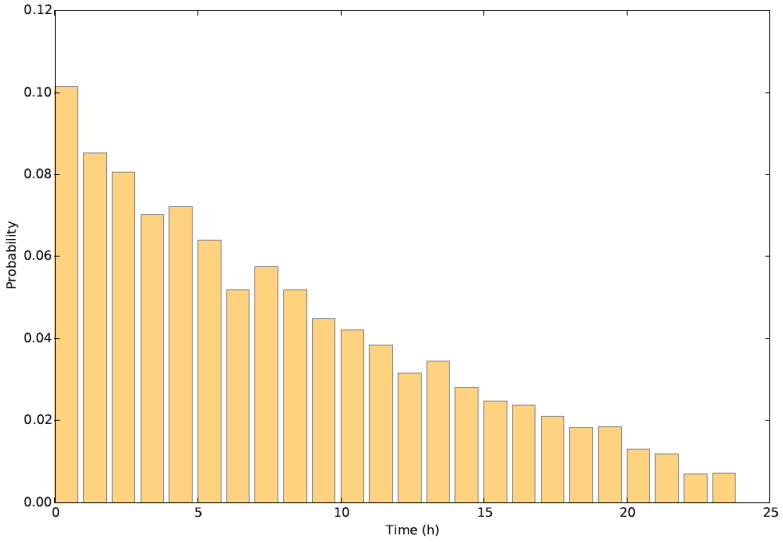
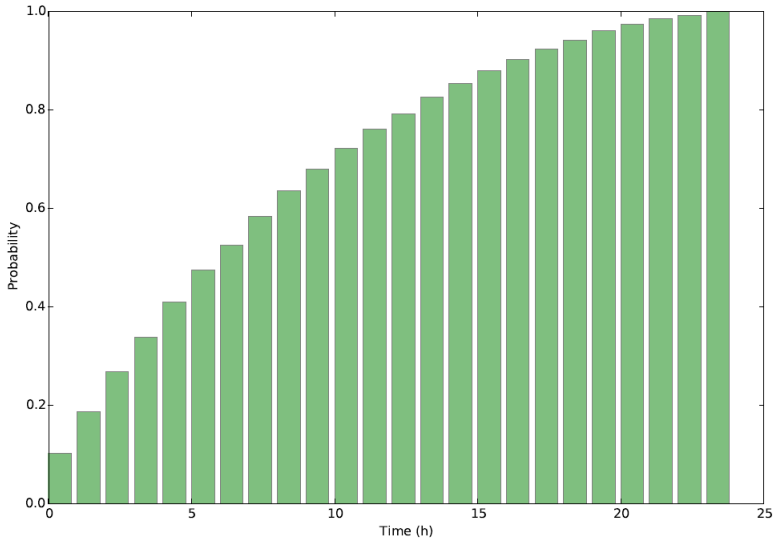


Figure 7.3.15: Probability distribution of S4 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.

Figure 7.3.16: Cumulative distribution of S4 duration time (mission time = 24h) after LOOPL initiating event for EPR FLA3.



Distribution of situation duration time

7.3.4.2 First sequences

N°	Value	Contrib.	Contrib. cum.	Sequence of transitions
0	3.180E-05	0.9%	0.9%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.06) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
1	3.130E-05	0.9%	1.7%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(9.85) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
2	3.100E-05	0.8%	2.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.48) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
3	3.000E-05	0.8%	3.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.04) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
4	2.940E-05	0.8%	4.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.40) CC4LH0001GE10K_FRL_1_to_2_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
5	2.940E-05	0.8%	5.0%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.77) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
6	2.940E-05	0.8%	5.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.01) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
7	2.940E-05	0.8%	6.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.74) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
8	2.930E-05	0.80%	7.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.75) CC4LH0001GE10K_FRL_3_to_1_2 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
9	2.900E-05	0.8%	8.2%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.62) CC4LH0001GE10K_FRL_2_to_1_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
10	2.890E-05	0.8%	9.00%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.84) CC4LH0001GE10K_FRL_3_to_1_2 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
11	2.880E-05	0.8%	9.8%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.40) CC4LH0001GE10K_FRL_3_to_1_2 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
12	2.860E-05	0.8%	10.6%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.90) CC4LH0001GE10K_FRL_1_to_2_3 LHR0001GE10K_FRL LHP0001GE10K_FRL LHQ0001GE10K_FRL]
13	2.850E-05	0.8%	11.3%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.76) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
14	2.850E-05	0.8%	12.1%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.25) CC4LH0001GE10K_FRL_3_to_2_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
15	2.830E-05	0.8%	12.9%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(10.74) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
16	2.830E-05	0.77%	13.68%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.02) CC4LH0001GE10K_FRL_2_to_3_4 LHS0001GE10K_FRL LHQ0001GE10K_FRL LHR0001GE10K_FRL]
17	2.830E-05	0.8%	14.4%	[(0.00) GRID_FRL UnitTurbine_FSHO] [(11.63)

7.4 Analysis and insights of the results

The post processing of the dynamic results shows that among the main contributor events to the occurrence of all situations are :

1. the long loss of all diesels by *CC Failure to Run*
2. the loss of all batteries feeding the I&C by the *CC Failure to start*
3. the loss of *I&C common logic*

However, it is difficult to quickly compare theses results with those given by the static PRA model since the time indicators cannot be estimated by means of a static approach. Further analyses will be performed by CNEN to fully understand the possible differences between the static and the dynamic approach presented in this document. Moreover, thermodynamic support studies will also be carried out in order the assess the relevancies of situation time and duration estimated in the dynamic study.

Appendix

In this appendix is shown a part of the knowledge base CONCERTO (unless of the CCF, which consists of a significant number of lines of code).

```
ORDRE_DES_ETAPES
        init_step;
        i_and_c_operate;
        fluid_propagation;
        i_and_c_propagation;
        ccf_propagation;
        default_step; (* In a perfect world, no rule should use the default_step *)
        saving_step;
        reset_step;
NOMS_DES_GROUPES
        simu_group;
(* Global attribute to manage the general behavior of the knowledge base *)
GLOBAL
    CONSTANTE
                global_param_threshold_min DOMAINE REEL PAR_DEFAUT 1e-99;
                global_param_threshold_max DOMAINE REEL PAR_DEFAUT 1e99;

    ATTRIBUT
                global_maint_enabled DOMAINE BOOLEEN PAR_DEFAUT VRAI;
                global_maint_step_finished DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        INTERACTION
                test_global_maint_enabled
                GROUPE simu_group
                ETAPE init_step
                SI NON global_maint_enabled
                ALORS global_maint_step_finished <-- VRAI;

(* Classe =ComponentAbstract= *)
```

```
(* Tous les objets de la BdC =CONCERTO= héritent et partagent donc les
attributs de la classe =ComponentAbstract=. *)

(* - Attributs : *)
(*   - =gamma_upm= : Probabilité d'indisponibilité du composant pour maintenance
(*     préventive. *)
(*   - =duration_upm= : Durée de la maintenance préventive. *)
(*   - =is_selected_upm= : Indicateur logique rendant possible une maintenance
(*       préventive sur le composant. *)

TYPE ComponentAbstract;

        ATTRIBUT
                gamma_upm DOMAINE REEL PAR_DEFAUT 1e-100 LIBELLE "Probability of
unavailability for maintenance";
                duration_upm DOMAINE REEL PAR_DEFAUT 1e100 LIBELLE "Duration time
of maintenance operations";
                is_selected_upm DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        EFFET
                flowed;
                fed;

        PANNE
                u_UPM;

        OCCURRENCE
                occ_upm
                GROUPE simu_group
                SI is_selected_upm
                IL_PEUT_SE_PRODUIRE
                INDISPONIBILITE u_UPM LIBELLE "Maintenance unavailability of %OBJET"
                LOI INS ( gamma_upm )
```

```
                PROVOQUE is_selected_upm <-- FAUX, global_maint_step_finished <-- VRAI
                OU_BIEN
                TRANSITION no_UPM
                PROVOQUE is_selected_upm <-- FAUX, global_maint_step_finished <-- VRAI;

                occ_end_upm
                GROUPE simu_group
                SI u_UPM
                IL_PEUT_SE_PRODUIRE
                REPARATION end_UPM REPARE u_UPM
                LOI T_C ( duration_upm );

(* TODO Classe =ControlledCompAbstract= *)

(* TODO : Si un controlledCompAbstract ne possède pas au moins un type
de FailureToStart ou FailureToOperate, proceed_request ne peut passer
à vrai... Il faut envisager la création d'un état intermédiaire
(e.g. order_received qui permette ensuite le passage à vrai de proceed_request *)

TYPE ControlledCompAbstract SORTE_DE ComponentAbstract;

        ATTRIBUT
                proceed_request DOMAINE BOOLEEN PAR_DEFAUT FAUX;
                is_requested DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        INTERFACE
                control GENRE ControlAbstract CARDINAL 0 JUSQUA INFINI;

        EFFET
                stop_is_requested_check;

        INTERACTION
                (* reset_is_requested *)
```

```
                (* GROUPE simu_group *)
                (* ETAPE i_and_c_operate *)
                (* SI proceed_request *)
                (* ALORS is_requested <-- FAUX; *)

                test_is_requested
                (* ETAPE default_step *)
                GROUPE simu_group
                ETAPE i_and_c_propagation
                SI ((IL_EXISTE x UN control TEL_QUE send_order DE x) ET MARCHE)
                ET NON stop_is_requested_check
                ALORS is_requested <-- NON is_requested,
                stop_is_requested_check <-- VRAI;

(* Classe =StartingCompAbstract= *)


TYPE StartingCompAbstract SORTE_DE ControlledCompAbstract;

        ATTRIBUT
                is_started DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        INTERACTION
                test_request_proceeding
                GROUPE simu_group
                ETAPE i_and_c_operate
                SI proceed_request
                ALORS is_started <-- NON is_started, proceed_request <-- FAUX,
                        is_requested <-- FAUX;

(* Classe =OperatedCompAbstract= *)
```

```
TYPE OperatedCompAbstract SORTE_DE ControlledCompAbstract;

        ATTRIBUT
                is_open DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        INTERACTION
                test_request_proceeding
                GROUPE simu_group
                ETAPE i_and_c_operate
                SI proceed_request
                ALORS is_open <-- NON is_open, proceed_request <-- FAUX,
                        is_requested <-- FAUX;


(* Classe =SourceAbstract= *)

TYPE SourceAbstract SORTE_DE ComponentAbstract;

        INTERACTION
                test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE;

        test_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS flowed <-- fed;

(* Classe =SourceControlledAbstract= *)

TYPE SourceControlledAbstract SORTE_DE SourceAbstract StartingCompAbstract;
```

```
        INTERACTION
                test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET is_started;

(* Classe =ConsumerAbstract= *)

TYPE ConsumerAbstract SORTE_DE ComponentAbstract;

    INTERFACE
                linked_comp GENRE ComponentAbstract CARDINAL 0 JUSQUA INFINI;

        EFFET
                linked_comp_flowed;

    INTERACTION

                test_linked_comp_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS linked_comp_flowed <-- IL_EXISTE x UN linked_comp TEL_QUE flowed DE x;

        test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET linked_comp_flowed;

        test_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS flowed <-- fed;
```

```
(* Classe =NodeAbstract= *)

TYPE NodeAbstract SORTE_DE ConsumerAbstract;

    ATTRIBUT
                k DOMAINE ENTIER PAR_DEFAUT 1;

    INTERACTION
                test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
                ALORS fed <-- IL_EXISTE AU_MOINS k x
                INCLUS_DANS linked_comp TEL_QUE flowed( x );

                test_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
                ALORS flowed <-- fed;

(* Classe =ElecComp= *)

TYPE ElecComp SORTE_DE ComponentAbstract FailureToRunComp;

(* Classe =ElecLink= *)

TYPE ElecLink;
        INTERFACE ori_comp GENRE ElecComp CARDINAL 1;

(* Classe =ElecSource= *)

TYPE ElecSource SORTE_DE ElecComp SourceAbstract;

(* Classe =ElecSourceControlled= *)
```

```
TYPE ElecSourceControlled SORTE_DE ElecComp SourceControlledAbstract FailureToStartComp;

(* Classe =ElecConsumer= *)

TYPE ElecConsumer SORTE_DE ElecComp ConsumerAbstract;

    INTERFACE
                linked_comp GENRE ElecComp CARDINAL 0 JUSQUA INFINI;

(* Classe =ElecNode= *)

TYPE ElecNode SORTE_DE NodeAbstract ElecConsumer;

(* Classe =Connection= *)

TYPE Connection SORTE_DE ElecConsumer;

(* Classe =Substation= *)

TYPE Substation SORTE_DE ElecConsumer;

(* Classe =Grid= *)

TYPE Grid SORTE_DE ElecSource;

(* Classe =Transformer= *)

TYPE Transformer SORTE_DE ElecConsumer;

(* Classe =Alternator= *)

TYPE Alternator SORTE_DE ElecSource;
```

```
(* Classe =TurboAlternator= *)

TYPE TurboAlternator SORTE_DE Alternator;

        INTERFACE
                turbine GENRE Turbine CARDINAL 1 JUSQUA 1;

        INTERACTION
        test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET flowed( turbine );

        test_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS flowed <-- fed;

(* Classe =Diesel= *)

TYPE Diesel SORTE_DE ElecSourceControlled;

(* Classe =Battery= *)

TYPE Battery SORTE_DE ElecSourceControlled;

        CONSTANTE
        battery_autonomy_full DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Battery theoretical autonomy in the considered time unit";

    ATTRIBUT
                battery_autonomy_current DOMAINE REEL PAR_DEFAUT
```

```
                battery_autonomy_full LIBELLE "Current battery autonomy";

        INTERACTION
                test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET is_started ET battery_autonomy_current > 0;

    OCCURRENCE
                occ_empty_battery
                GROUPE simu_group
                SI fed ET battery_autonomy_full < global_param_threshold_max
                IL_PEUT_SE_PRODUIRE
                TRANSITION t_BTE LIBELLE "Battery of %OBJET is empty"
                LOI T_C ( battery_autonomy_current )
                PROVOQUE battery_autonomy_current <-- 0;

(* Classe =Busbar= *)

TYPE Busbar SORTE_DE ElecConsumer;


    ATTRIBUT
                (* This is used apply the occurrence rules when Busbar is not fed anymore *)
                (* Useful for normal/spare mechanism to take place and fed back the busbar *
                fed_delay DOMAINE BOOLEEN PAR_DEFAUT VRAI;

        PANNE
                (* Busbar destruction due to multiple feeding *)
                f_FDE;

        INTERACTION
        test_fed
```

```
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET (linked_comp_flowed OU fed_delay);

        test_fed_delay
                GROUPE simu_group
                ETAPE saving_step
                SI fed
                ALORS fed_delay <-- fed;


    OCCURRENCE
                (* We suppose that the busbar is always destroyed if fed by at least *)
                (* two sources *)
                occ_FDE
                GROUPE simu_group
                SI IL_EXISTE AU_MOINS 2 comp INCLUS_DANS linked_comp TEL_QUE flowed( comp )
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FDE
                LOI INS (1);

                occ_unfed
                GROUPE simu_group
                SI NON linked_comp_flowed ET fed_delay
                IL_PEUT_SE_PRODUIRE
                TRANSITION t_UNFED LIBELLE "%OBJET is unfed"
                LOI T_C( 0 )
                PROVOQUE fed_delay <-- FAUX;

(* Classe =Breaker= *)

TYPE Breaker SORTE_DE ElecConsumer FailureToOperateComp;
```

```
        (* A breaker is supposed to may fail only when requested and not when it runs,*)
    (* so it is defined as failure to close and a failure  to open component.
*)
        (* Failures to run are not considered at all.
*)

        INTERACTION
        test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- (IL_EXISTE x UN linked_comp TEL_QUE flowed DE x) ET NON u_UPM;

        test_flowed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS flowed <-- NON is_open ET fed;

(* Classe =HydrComp= *)

TYPE HydrComp SORTE_DE ComponentAbstract FailureToRunComp;

(* Classe =HydrLink= *)

TYPE HydrLink;
        INTERFACE ori_comp GENRE HydrComp CARDINAL 1;

(* Classe =HydrSource= *)

TYPE HydrSource SORTE_DE HydrComp SourceAbstract;

(* Classe =HydrSourceControlled= *)

TYPE HydrSourceControlled SORTE_DE HydrComp SourceControlledAbstract
```

```
FailureToStartComp;

(* Classe =HydrConsumer= *)

TYPE HydrConsumer SORTE_DE HydrComp ConsumerAbstract;

    INTERFACE
                linked_comp GENRE HydrComp CARDINAL 0 JUSQUA INFINI;

(* Classe =HydrNode= *)

TYPE HydrNode SORTE_DE NodeAbstract HydrConsumer;

(* Classe =Turbine= *)

TYPE Turbine SORTE_DE HydrConsumer;

(* Classe =MainUnitTurbine= *)

TYPE MainUnitTurbine SORTE_DE Turbine ControlledCompAbstract;

        CONSTANTE
                lambda_frho DOMAINE REEL PAR_DEFAUT 1e-100 LIBELLE
                "Failure rate associated to household operation of %OBJET";
                mttr_frho DOMAINE REEL PAR_DEFAUT 1e100 LIBELLE "MTTR of
                household operation of %OBJET";
                gamma_fsho DOMAINE REEL PAR_DEFAUT 1e-100 LIBELLE "Failre to
                start probability associated to household operation of %OBJET";

        ATTRIBUT
                is_household_operating DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        PANNE
```

```
        f_FRHO;
        f_FSHO;

INTERACTION
        test_request_proceeding
        GROUPE simu_group
        ETAPE i_and_c_operate
        SI proceed_request
        ALORS is_household_operating <-- VRAI, proceed_request <-- FAUX;


OCCURRENCE
        occ_FSHO
        GROUPE simu_group
        SI MARCHE ET is_requested ET NON is_household_operating
        ET gamma_fsho > global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FSHO
        LOI INS (gamma_fsho)
        PROVOQUE proceed_request <-- FAUX
        OU_BIEN
        TRANSITION no_f_FSHO
        PROVOQUE proceed_request <-- VRAI;

        occ_FRHO
        GROUPE simu_group
        SI MARCHE
        ET is_household_operating ET lambda_frho > global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FRHO
        LOI EXP (lambda_frho);

occ_rep_FRHO
```

```
                GROUPE simu_group
        SI f_FRHO ET mttr_frho < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FRHO
        REPARE f_FRHO
        LOI EXP (1/(mttr_frho));

(* Classe =FailureAbstract= *)

TYPE FailureAbstract;

(* Classe =FailureToRunAbstract= *)

TYPE FailureToRunAbstract SORTE_DE FailureAbstract ComponentAbstract;

        ATTRIBUT
                init_occurred DOMAINE BOOLEEN PAR_DEFAUT FAUX;

(* Classe =FailureToRun= *)

TYPE FailureToRun SORTE_DE FailureToRunAbstract;

    CONSTANTE
                init_fr     DOMAINE BOOLEEN PAR_DEFAUT FAUX;
        lambda_fr  DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Failure rate associated to f_FR of %OBJET";
        mttr_fr     DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Mean time to repair f_FR";

    PANNE
        f_FR LIBELLE "Abstract failure to run of %OBJET";

        ATTRIBUT
```

```
        (* Indicates if a FR was in progress during the last interaction rules *)
        f_FR_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fr DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current failure
        (* comes already from a CCF *)
        stop_ccf_fr_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FR happened not due to CCF propagation *)
        occ_fr_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fr_enabled;

INTERACTION
        save_f_FR_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FR_previous <-- f_FR;

        (* cf. effect occ_fr_no_ccf *)
        test_occ_fr
        GROUPE simu_group
ETAPE fluid_propagation
        ALORS occ_fr_no_ccf <-- f_FR ET (NON f_FR_previous);

        (* cf. stop_ccf_frs_propagation *)
        test_stop_ccf_fr_propagation
        GROUPE simu_group
ETAPE default_step
        SI stop_ccf_fr_propagation ALORS occ_fr_no_ccf <-- FAUX;
```

```
            stop_ccf_fr_propagation <-- FAUX;

            (* cf. ccf_fr_enabled *)
            test_ccf_fr_enabled
            GROUPE simu_group
            ETAPE default_step
            ALORS ccf_fr_enabled <-- NON f_FR;

OCCURRENCE
    occ_FR
            GROUPE simu_group
    SI MARCHE ET fed ET lambda_fr > global_param_threshold_min
    IL_PEUT_SE_PRODUIRE
    DEFAILLANCE f_FR
        LOI EXP (lambda_fr);

    occ_FR_ccf
            GROUPE simu_group
    SI MARCHE ET fed ET ccf_force_fr
    IL_PEUT_SE_PRODUIRE
    DEFAILLANCE f_FR
        LOI INS (1)
            PROVOQUE ccf_force_fr <-- FAUX, stop_ccf_fr_propagation <-- VRAI;

    occ_FR_init
            GROUPE simu_group
    SI init_fr ET NON init_occurred ET global_maint_step_finished
    IL_PEUT_SE_PRODUIRE
    DEFAILLANCE f_FR
        LOI INS (1)
            PROVOQUE init_occurred <-- VRAI;

    occ_rep_FR
```

```
            GROUPE simu_group
        SI f_FR ET mttr_fr < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FR
        REPARE f_FR
        LOI EXP (1/(mttr_fr));

(* Classe =FailureToRunShort= *)

TYPE FailureToRunShort SORTE_DE FailureToRunAbstract;

    CONSTANTE
            init_frs    DOMAINE BOOLEEN PAR_DEFAUT FAUX;
        lambda_frs DOMAINE REEL PAR_DEFAUT 1e-100
            LIBELLE "Failure rate associated to f_FRS of %OBJET";
        mttr_frs    DOMAINE REEL PAR_DEFAUT 1e100
            LIBELLE "Mean time to repair f_FRS ";

    PANNE
        f_FRS LIBELLE "Short failure to run short of %OBJET";

        ATTRIBUT
            (* Indicates if a FRS was in progress during the last interaction rules *)
            f_FRS_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

            (* Attribute used by a CCF group to propagate a CCF event *)
            ccf_force_frs DOMAINE BOOLEEN PAR_DEFAUT FAUX;

            (* Allow to stop a CCF propagation if the current failure
            (* comes already frsom a CCF *)
            stop_ccf_frs_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        EFFET
```

```
        (* Indicates that a FRS happened not due to CCF propagation *)
        occ_frs_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_frs_enabled;

INTERACTION
        save_f_FRS_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FRS_previous <-- f_FRS;

        (* cf. effect occ_frs_no_ccf *)
        test_occ_frs
        GROUPE simu_group
ETAPE fluid_propagation
        ALORS occ_frs_no_ccf <-- f_FRS ET (NON f_FRS_previous);

        (* cf. stop_ccf_frs_propagation *)
        test_stop_ccf_frs_propagation
        GROUPE simu_group
ETAPE default_step
        SI stop_ccf_frs_propagation ALORS occ_frs_no_ccf <-- FAUX,
        stop_ccf_frs_propagation <-- FAUX;

        (* cf. ccf_frs_enabled *)
        test_ccf_frs_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_frs_enabled <-- NON f_FRS;


OCCURRENCE
    occ_FRS
```

```
            GROUPE simu_group
        SI MARCHE ET fed ET lambda_frs > global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FRS
            LOI EXP (lambda_frs);

        occ_FRS_ccf
            GROUPE simu_group
        SI MARCHE ET fed ET ccf_force_frs
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FRS
            LOI INS (1)
                PROVOQUE ccf_force_frs <-- FAUX, stop_ccf_frs_propagation <-- VRAI;

        occ_FRS_init
            GROUPE simu_group
        SI init_frs ET NON init_occurred ET global_maint_step_finished
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FRS
            LOI INS (1)
                PROVOQUE init_occurred <-- VRAI;

        occ_rep_FRS
            GROUPE simu_group
        SI f_FRS ET mttr_frs < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FRS
        REPARE f_FRS
        LOI EXP (1/(mttr_frs));

(* Classe =FailureToRunLong= *)

TYPE FailureToRunLong SORTE_DE FailureToRunAbstract;
```

```
CONSTANTE
        init_frl    DOMAINE BOOLEEN PAR_DEFAUT FAUX;
    lambda_frl DOMAINE REEL PAR_DEFAUT 1e-100
        LIBELLE "Failure rate associated to Failure to RUN of %OBJET";
    mttr_frl    DOMAINE REEL PAR_DEFAUT 1e100
        LIBELLE "Mean time to repair f_FRL ";

PANNE
    f_FRL LIBELLE "Long failure to run of %OBJET";

    ATTRIBUT
        (* Indicates if a FRL was in progress during the last interaction rules *)
        f_FRL_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_frl DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current failure
        (* comes already frlom a CCF *)
        stop_ccf_frl_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

    EFFET
        (* Indicates that a FRL happened not due to CCF propagation *)
        occ_frl_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_frl_enabled;

    INTERACTION
        save_f_FRL_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FRL_previous <-- f_FRL;
```

```
            (* cf. effect occ_frl_no_ccf *)
            test_occ_frl
            GROUPE simu_group
    ETAPE fluid_propagation
            ALORS occ_frl_no_ccf <-- f_FRL ET (NON f_FRL_previous);

            (* cf. stop_ccf_frl_propagation *)
            test_stop_ccf_frl_propagation
            GROUPE simu_group
    ETAPE default_step
            SI stop_ccf_frl_propagation ALORS occ_frl_no_ccf <-- FAUX,
            stop_ccf_frl_propagation <-- FAUX;

            (* cf. ccf_frl_enabled *)
            test_ccf_frl_enabled
            GROUPE simu_group
            ETAPE default_step
            ALORS ccf_frl_enabled <-- NON f_FRL;

OCCURRENCE
    occ_FRL
            GROUPE simu_group
    SI MARCHE ET fed ET lambda_frl > global_param_threshold_min
    IL_PEUT_SE_PRODUIRE
    DEFAILLANCE f_FRL
        LOI EXP (lambda_frl);

    occ_FRL_ccf
            GROUPE simu_group
    SI MARCHE ET fed ET ccf_force_frl
    IL_PEUT_SE_PRODUIRE
    DEFAILLANCE f_FRL
```

```
            LOI INS (1)
                PROVOQUE ccf_force_frl <-- FAUX, stop_ccf_frl_propagation <-- VRAI;

        occ_FRL_init
                GROUPE simu_group
        SI init_frl ET NON init_occurred ET global_maint_step_finished
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FRL
            LOI INS (1)
                PROVOQUE init_occurred <-- VRAI;

        occ_rep_FRL
                GROUPE simu_group
        SI f_FRL ET mttr_frl < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FRL
        REPARE f_FRL
        LOI EXP (1/(mttr_frl));

(* Classe =FailureToRunComp= *)

TYPE FailureToRunComp SORTE_DE FailureToRun FailureToRunShort FailureToRunLong;

(* Classe =FailureToOperateAbstract= *)

TYPE FailureToOperateAbstract SORTE_DE FailureAbstract OperatedCompAbstract;

(* Classe =FailureToOpen= *)

TYPE FailureToOpen SORTE_DE FailureToOperateAbstract;
        CONSTANTE
                gamma_fo DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability associated to the failure to open";
```

```
        mttr_fo  DOMAINE REEL PAR_DEFAUT 1e100
        LIBELLE "Mean time to repair the failure to open";

PANNE
        f_FO LIBELLE "Failure to open of %OBJET";

ATTRIBUT
        (* Indicates if a FO was in progress during the last interaction rules *)
        f_FO_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fo DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current failure
        (* comes already foom a CCF *)
        stop_ccf_fo_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FO happened not due to CCF propagation *)
        occ_fo_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fo_enabled;


INTERACTION
        save_f_FO_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FO_previous <-- f_FO;

        (* cf. effect occ_fo_no_ccf *)
        test_occ_fo
        GROUPE simu_group
```

```
ETAPE fluid_propagation
        ALORS occ_fo_no_ccf <-- f_FO ET (NON f_FO_previous);

        (* cf. stop_ccf_fo_propagation *)
        test_stop_ccf_fo_propagation
        GROUPE simu_group
ETAPE default_step
        SI stop_ccf_fo_propagation ALORS occ_fo_no_ccf <-- FAUX,
        stop_ccf_fo_propagation <-- FAUX;

        (* cf. ccf_fo_enabled *)
        test_ccf_fo_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fo_enabled <-- NON f_FO;

OCCURRENCE

        occ_FO
        GROUPE simu_group
        SI MARCHE ET is_requested ET NON is_open ET gamma_fo > global_param_threshol
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FO
        LOI INS ( gamma_fo )
        PROVOQUE proceed_request <-- FAUX
        OU_BIEN
        TRANSITION no_f_FO
        PROVOQUE proceed_request <-- VRAI;

        (* The is_open condition is used to deal with the case of simultaneous dema
        occ_FO_ccf
        GROUPE simu_group
        SI (is_requested OU is_open) ET
```

```
                gamma_fo > global_param_threshold_min ET ccf_force_fo
                IL_PEUT_SE_PRODUIRE
                DEFAILLANCE f_FO
                LOI INS (1)
                PROVOQUE proceed_request <-- FAUX,
                ccf_force_fo <-- FAUX, stop_ccf_fo_propagation <-- VRAI,
                (* Try to return the component back to its context before
                (* the CCF propagation (case of simultaneous demands *)
                is_open <-- FAUX,
                is_requested <-- VRAI;


                occ_rep_FO
                GROUPE simu_group
                SI f_FO ET mttr_fo < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FO
        REPARE f_FO
        LOI EXP (1/(mttr_fo));

(* Classe =FailureToClose= *)

TYPE FailureToClose SORTE_DE FailureToOperateAbstract;

        CONSTANTE
                gamma_fc DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability associated to the failure to close";
                mttr_fc  DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Mean time to repair the failure to close ";

        PANNE
                f_FC LIBELLE "Failure to close of %OBJET";
```

```
ATTRIBUT
        (* Indicates if a FC was in progress during the last
        (* interaction rules *)
        f_FC_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fc DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current failure
        (* comes already fcom a CCF *)
        stop_ccf_fc_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FC happened not due to CCF propagation *)
        occ_fc_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fc_enabled;

INTERACTION
        save_f_FC_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FC_previous <-- f_FC;

        (* cf. effect occ_fc_no_ccf *)
        test_occ_fc
        GROUPE simu_group
        ETAPE fluid_propagation
        ALORS occ_fc_no_ccf <-- f_FC ET (NON f_FC_previous);

        (* cf. stop_ccf_fc_propagation *)
        test_stop_ccf_fc_propagation
        GROUPE simu_group
```

```
        ETAPE default_step
        SI stop_ccf_fc_propagation ALORS occ_fc_no_ccf <-- FAUX,
        stop_ccf_fc_propagation <-- FAUX;

        (* cf. ccf_fc_enabled *)
        test_ccf_fc_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fc_enabled <-- NON f_FC;


OCCURRENCE
        occ_FC
        GROUPE simu_group
        SI MARCHE ET is_requested ET is_open ET
        gamma_fc > global_param_threshold_min ET NON ccf_force_fc
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FC
        LOI INS ( gamma_fc )
        PROVOQUE proceed_request <-- FAUX
        OU_BIEN
        TRANSITION no_f_FC
        PROVOQUE proceed_request <-- VRAI;

        (* The is_open condition is used to deal with the case
        (* of simultaneous demands *)
        occ_FC_ccf
        GROUPE simu_group
        SI (is_requested OU NON is_open) ET
        gamma_fc > global_param_threshold_min ET ccf_force_fc
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FC
        LOI INS (1)
```

```
                PROVOQUE proceed_request <-- FAUX,
                ccf_force_fc <-- FAUX, stop_ccf_fc_propagation <-- VRAI,
                (* Try to return the component back to its context before
                (* the CCF propagation (case of simultaneous demands *)
                is_open <-- VRAI,
                is_requested <-- VRAI;


                occ_rep_FC
                GROUPE simu_group
                SI f_FC ET mttr_fc < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FC
        REPARE f_FC
        LOI EXP (1/(mttr_fc));

(* Classe =FailureToOperateComp= *)

TYPE FailureToOperateComp SORTE_DE FailureToOpen FailureToClose;

(* Classe =FailureToStartAbstract= *)

TYPE FailureToStartAbstract SORTE_DE FailureAbstract StartingCompAbstract;

        OCCURRENCE
                occ_stop
                GROUPE simu_group
                SI MARCHE ET is_requested ET is_started
                IL_PEUT_SE_PRODUIRE
                TRANSITION stop LIBELLE "Stop %OBJET"
                LOI INS (1)
                PROVOQUE proceed_request <-- VRAI;
```

```
(* Classe =FailureToStart= *)

TYPE FailureToStart SORTE_DE FailureToStartAbstract;

        CONSTANTE
                gamma_fs DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability associated to failure to start";
                mttr_fs  DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Mean time to repair failure to open";

        PANNE
                f_FS LIBELLE "Failure to start of %OBJET";

        ATTRIBUT
                (* Indicates if a FS was in progress during the last
                (* interaction rules *)
                f_FS_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

                (* Attribute used by a CCF group to propagate a CCF event *)
                ccf_force_fs DOMAINE BOOLEEN PAR_DEFAUT FAUX;

                (* Allow to stop a CCF propagation if the current failure
                (* comes already fsom a CCF *)
                stop_ccf_fs_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        EFFET
                (* Indicates that a FS happened not due to CCF propagation *)
                occ_fs_no_ccf;
                (* Indicates that a CCF can happen during the next occurrence rules *)
                ccf_fs_enabled;

        INTERACTION
                save_f_FS_previous
```

```
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FS_previous <-- f_FS;

        (* cf. effect occ_fs_no_ccf *)
        test_occ_fs
        GROUPE simu_group
        ETAPE fluid_propagation
        ALORS occ_fs_no_ccf <-- f_FS ET (NON f_FS_previous);

        (* cf. stop_ccf_fs_propagation *)
        test_stop_ccf_fs_propagation
        GROUPE simu_group
        ETAPE default_step
        SI stop_ccf_fs_propagation ALORS occ_fs_no_ccf <-- FAUX,
        stop_ccf_fs_propagation <-- FAUX;

        (* cf. ccf_fs_enabled *)
        test_ccf_fs_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fs_enabled <-- NON f_FS;


OCCURRENCE
        occ_FS
        GROUPE simu_group
        SI MARCHE ET is_requested ET NON is_started ET
        gamma_fs > global_param_threshold_min ET NON ccf_force_fs
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FS
        LOI INS (gamma_fs)
        PROVOQUE proceed_request <-- FAUX
```

```
                OU_BIEN
                TRANSITION no_f_FS
                PROVOQUE proceed_request <-- VRAI;

                (* The is_started condition is used to deal with the
                (* case of simultaneous demands *)
                occ_FS_ccf
                GROUPE simu_group
                SI (is_requested OU is_started) ET
                gamma_fs > global_param_threshold_min ET ccf_force_fs
                IL_PEUT_SE_PRODUIRE
                DEFAILLANCE f_FS
                LOI INS (1)
                PROVOQUE proceed_request <-- FAUX,
                ccf_force_fs <-- FAUX, stop_ccf_fs_propagation <-- VRAI,
                (* Try to return the component back to its context before
                (* the CCF propagation (case of simultaneous demands *)
                is_started <-- FAUX,
                is_requested <-- VRAI;

                occ_rep_FS
                GROUPE simu_group
                SI f_FS ET mttr_fs < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FS
        REPARE f_FS
        LOI EXP (1/(mttr_fs));

(* Classe =FailureToStartShort= *)

TYPE FailureToStartShort SORTE_DE FailureToStartAbstract;

        CONSTANTE
```

```
        gamma_fss DOMAINE REEL PAR_DEFAUT 1e-100
        LIBELLE "Probability associated to short failure to start";
        mttr_fss  DOMAINE REEL PAR_DEFAUT 1e100
        LIBELLE "Mean time to repair short failure to open";

PANNE
        f_FSS LIBELLE "Short failure to start of %OBJET";

ATTRIBUT
        (* Indicates if a FSS was in progress during the last interaction rules *)
        f_FSS_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fss DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current failure
        (* comes already fssom a CCF *)
        stop_ccf_fss_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FSS happened not due to CCF propagation *)
        occ_fss_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fss_enabled;

INTERACTION
        save_f_FSS_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FSS_previous <-- f_FSS;

        (* cf. effect occ_fss_no_ccf *)
        test_occ_fss
```

```
        GROUPE simu_group
ETAPE fluid_propagation
        ALORS occ_fss_no_ccf <-- f_FSS ET (NON f_FSS_previous);

        (* cf. stop_ccf_fss_propagation *)
        test_stop_ccf_fss_propagation
        GROUPE simu_group
ETAPE default_step
        SI stop_ccf_fss_propagation ALORS
        occ_fss_no_ccf <-- FAUX, stop_ccf_fss_propagation <-- FAUX;

        (* cf. ccf_fss_enabled *)
        test_ccf_fss_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fss_enabled <-- NON f_FSS;

OCCURRENCE
        occ_FSS
        GROUPE simu_group
        SI MARCHE ET is_requested ET
        NON is_started ET gamma_fss > global_param_threshold_min ET NON ccf_force_fs
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FSS
        LOI INS (gamma_fss)
        PROVOQUE proceed_request <-- FAUX
        OU_BIEN
        TRANSITION no_f_FSS
        PROVOQUE proceed_request <-- VRAI;

        (* The is_started condition is used to deal with the case
        (* of simultaneous demands *)
        occ_FSS_ccf
```

```
                GROUPE simu_group
                SI (is_requested OU is_started) ET
                gamma_fss > global_param_threshold_min ET ccf_force_fss
                IL_PEUT_SE_PRODUIRE
                DEFAILLANCE f_FSS
                LOI INS (1)
                PROVOQUE proceed_request <-- FAUX,
                ccf_force_fss <-- FAUX, stop_ccf_fss_propagation <-- VRAI,
                (* Try to return the component back to its context before *)
                (* the CCF propagation (case of simultaneous demands *)
                is_started <-- FAUX,
                is_requested <-- VRAI;

                occ_rep_FSS
                GROUPE simu_group
                SI f_FSS ET mttr_fss < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FSS
        REPARE f_FSS
        LOI EXP (1/(mttr_fss));

(* Classe =FailureToStartLong= *)

TYPE FailureToStartLong SORTE_DE FailureToStartAbstract;

        CONSTANTE
                gamma_fsl DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability associated to long failure to start";
                mttr_fsl  DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Mean time to repair long failure to open";

        PANNE
                f_FSL LIBELLE "Long failure to start of %OBJET";
```

```
ATTRIBUT
        (* Indicates if a FSL was in progress during the last interaction rules *)
        f_FSL_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fsl DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        (* Allow to stop a CCF propagation if the current *)
        (* failure comes already fslom a CCF *)
        stop_ccf_fsl_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FSL happened not due to CCF propagation *)
        occ_fsl_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fsl_enabled;

INTERACTION
        save_f_FSL_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FSL_previous <-- f_FSL;

        (* cf. effect occ_fsl_no_ccf *)
        test_occ_fsl
        GROUPE simu_group
        ETAPE fluid_propagation
        ALORS occ_fsl_no_ccf <-- f_FSL ET (NON f_FSL_previous);

        (* cf. stop_ccf_fsl_propagation *)
        test_stop_ccf_fsl_propagation
        GROUPE simu_group
```

```
        ETAPE default_step
        SI stop_ccf_fsl_propagation ALORS
        occ_fsl_no_ccf <-- FAUX, stop_ccf_fsl_propagation <-- FAUX;

        (* cf. ccf_fsl_enabled *)
        test_ccf_fsl_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fsl_enabled <-- NON f_FSL;

OCCURRENCE
        occ_FSL
        GROUPE simu_group
        SI MARCHE ET is_requested ET
        NON is_started ET gamma_fsl > global_param_threshold_min ET NON ccf_force_f
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FSL
        LOI INS (gamma_fsl)
        PROVOQUE proceed_request <-- FAUX
        OU_BIEN
        TRANSITION no_f_FSL
        PROVOQUE proceed_request <-- VRAI;

        (* The is_started condition is used to deal with the case *)
        (* of simultaneous demands *)
        occ_FSL_ccf
        GROUPE simu_group
        SI (is_requested OU is_started) ET
        gamma_fsl > global_param_threshold_min ET ccf_force_fsl
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FSL
        LOI INS (1)
        PROVOQUE proceed_request <-- FAUX,
```

```
                ccf_force_fsl <-- FAUX, stop_ccf_fsl_propagation <-- VRAI,
                (* Try to return the component back to its context before *)
                (* the CCF propagation (case of simultaneous demands *)
                is_started <-- FAUX,
                is_requested <-- VRAI;

                occ_rep_FSL
                GROUPE simu_group
                SI f_FSL ET mttr_fsl < global_param_threshold_max
        IL_PEUT_SE_PRODUIRE
        REPARATION rep_FSL
        REPARE f_FSL
        LOI EXP (1/(mttr_fsl));

(* Classe =FailureToStartComp= *)

TYPE FailureToStartComp SORTE_DE FailureToStart FailureToStartShort
                FailureToStartLong;

(* Classe =InstrumentationAbstract= *)

(* Principe important : on teste le changement à *)
(* l'échelle du groupe de composant en interface. *)

(* Du coup, si on met 'any' et 'fed', *)

(* TODO : *)
(* - Maybe create a class =FailureToAcquire= containing the failure managment *)

(* - Attributs : *)
(*   - comp_attr_diff :: Indique si l'attribut testé à changer de valeur *)

TYPE InstrumentationAbstract SORTE_DE ComponentAbstract;
```

```
    ATTRIBUT
            test_attr DOMAINE 'fed' 'flowed' PAR_DEFAUT 'fed';
            test_type DOMAINE 'any' 'all' PAR_DEFAUT 'any';

            comp_fed_previous DOMAINE BOOLEEN PAR_DEFAUT VRAI;
            comp_fed_now DOMAINE BOOLEEN PAR_DEFAUT VRAI;
            comp_flowed_previous DOMAINE BOOLEEN PAR_DEFAUT VRAI;
            comp_flowed_now DOMAINE BOOLEEN PAR_DEFAUT VRAI;

            comp_attr_diff DOMAINE BOOLEEN PAR_DEFAUT FAUX;


    INTERFACE
            comp_test GENRE ComponentAbstract CARDINAL 0 JUSQUA INFINI;

EFFET
            comp_fed_diff;
            comp_fed_eq;
            comp_flowed_diff;
            comp_flowed_eq;
            comp_attr_eq;

            cond_occ_ac;
            cond_occ_ac_context;

INTERACTION
            test_fed
            GROUPE simu_group
            ETAPE fluid_propagation
            ALORS fed <-- VRAI;

            test_curr_flowed
```

```
GROUPE simu_group
ETAPE fluid_propagation
ALORS flowed <-- VRAI;

test_comp_fed_now
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_fed_now <-- (test_type = 'any' ET
IL_EXISTE x UN comp_test TEL_QUE fed(x)) OU
(test_type = 'all' ET QQSOIT x UN comp_test ON_A fed(x));

save_comp_fed_previous
GROUPE simu_group
ETAPE saving_step
ALORS comp_fed_previous <-- comp_fed_now;

test_comp_fed_diff
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_fed_diff <-- (comp_fed_previous <> comp_fed_now);

test_comp_fed_eq
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_fed_eq <-- (comp_fed_previous = comp_fed_now);

test_comp_flowed_now
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_flowed_now <-- (test_type = 'any' ET
IL_EXISTE x UN comp_test TEL_QUE flowed(x)) OU
(test_type = 'all' ET QQSOIT x UN comp_test ON_A flowed(x));
```

```
save_comp_flowed_previous
GROUPE simu_group
ETAPE saving_step
ALORS comp_flowed_previous <-- comp_flowed_now;

test_comp_flowed_diff
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_flowed_diff <-- (comp_flowed_previous <> comp_flowed_now);

test_comp_flowed_eq
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_flowed_eq <-- (comp_flowed_previous = comp_flowed_now);

test_comp_attr_eq
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS comp_attr_eq <-- NON comp_attr_diff;

(* Occurrence acquisition condition *)
(* test_cond_occ_ac_context can be overloaded in acquisition
subclasses to fit with the specific behaviour of components *)
test_cond_occ_ac_context
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS cond_occ_ac_context <-- VRAI;

test_cond_occ_ac
GROUPE simu_group
ETAPE i_and_c_propagation
ALORS cond_occ_ac <-- MARCHE ET
(comp_flowed_diff ET test_attr = 'flowed') OU
```

```
        (comp_fed_diff ET test_attr = 'fed') ET cond_occ_ac_context;

        test_comp_attr_diff
        GROUPE simu_group
        ETAPE i_and_c_propagation
        SI cond_occ_ac
        ALORS comp_attr_diff <-- VRAI;

        (* reinit_comp_attr_diff *)
        (* ETAPE saving_step *)
        (* SI comp_attr_diff *)
        (* ALORS comp_attr_diff <-- FAUX; *)

(* Failure to acquire management *)
    CONSTANTE
        gamma_fac DOMAINE REEL PAR_DEFAUT 1e-100
        LIBELLE "Probability of acquisition failure";
        mttr_fac DOMAINE REEL PAR_DEFAUT 1e-100
        LIBELLE "Mean time to repair acquisition failure";

    PANNE
        f_FAC LIBELLE "Acquisition failure of %OBJET";

    ATTRIBUT
        (* Allow the occurrence of a failure to acquire signal *)
        fac_enabled DOMAINE BOOLEEN PAR_DEFAUT VRAI;

        (* Indicates if a FAC was in progress during the last *)
        (* interaction rules *)
        f_FAC_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;
        (* Attribute used by a CCF group to propagate a CCF event *)
        ccf_force_fac DOMAINE BOOLEEN PAR_DEFAUT FAUX;
        (* Allow to stop a CCF propagation if the current failure *)
```

```
        (* comes already from a CCF *)
        stop_ccf_fac_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

EFFET
        (* Indicates that a FAC happened not due to CCF propagation *)
        occ_fac_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_fac_enabled;

INTERACTION
        save_f_FAC_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_FAC_previous <-- f_FAC;

        (* cf. effect occ_fac_no_ccf *)
        test_occ_fac
        GROUPE simu_group
        ETAPE fluid_propagation
        ALORS occ_fac_no_ccf <-- f_FAC ET (NON f_FAC_previous);

        (* cf. stop_ccf_fac_propagation *)
        test_stop_ccf_fac_propagation
        GROUPE simu_group
        ETAPE default_step
        SI stop_ccf_fac_propagation ALORS
        occ_fac_no_ccf <-- FAUX, stop_ccf_fac_propagation <-- FAUX;

        (* cf. ccf_fac_enabled *)
        test_ccf_fac_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_fac_enabled <-- NON f_FAC;
```

```
OCCURRENCE
        (* Acquisition failure make the attribute change undetected *)
        occ_FAC
        GROUPE simu_group
SI (fac_enabled ET
        global_maint_step_finished) ET
        gamma_fac > global_param_threshold_min ET NON ccf_force_fac
IL_PEUT_SE_PRODUIRE
DEFAILLANCE f_FAC
    LOI INS (gamma_fac)
        PROVOQUE fac_enabled <-- FAUX
        OU_BIEN
        TRANSITION no_f_FAC
        PROVOQUE fac_enabled <-- FAUX;

        (* Optimistic assumption : The CCF is declared *)
        (*(and repair process is started) even if no acquisition is done *)
        occ_FAC_ccf
        GROUPE simu_group
        SI gamma_fac > global_param_threshold_min ET ccf_force_fac
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FAC
        LOI INS (1)
        PROVOQUE ccf_force_fac <-- FAUX,
        stop_ccf_fac_propagation <-- VRAI, fac_enabled <-- FAUX;

        (* occ_rep_FAC *)
        (* GROUPE simu_group *)
        (* SI (f_FAC ET comp_attr_diff) ET mttr_fac < global_param_threshold_max *)
(* IL_PEUT_SE_PRODUIRE *)
(* REPARATION rep_FAC *)
```

```
        (* REPARE f_FAC *)
        (* LOI EXP (1/(mttr_fac)); *)

(* Classe =ControlAbstract= *)

(* Generic control part *)
(* ---------------------- *)
TYPE ControlAbstract SORTE_DE ComponentAbstract;

        CONSTANTE
                lambda_spo DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Spurious order occurrence rate";
                gamma_fso DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Failure to send order";


        ATTRIBUT
                send_spurious_order DOMAINE BOOLEEN PAR_DEFAUT FAUX
                LIBELLE "Spurious unrequested order sended";
                send_order DOMAINE BOOLEEN PAR_DEFAUT FAUX
                LIBELLE "Indicate if the I&C is sending an order";

                (* Indicates if a SPO was in progress during the last interaction rules *)
                f_SPO_previous DOMAINE BOOLEEN PAR_DEFAUT FAUX;
                (* Attribute used by a CCF group to propagate a CCF event *)
                ccf_force_spo DOMAINE BOOLEEN PAR_DEFAUT FAUX;
                (* Allow to stop a CCF propagation if the current failure *)
                (* comes already spoom a CCF *)
                stop_ccf_spo_propagation DOMAINE BOOLEEN PAR_DEFAUT FAUX;

        EFFET
                cond_occ_so;
                cond_occ_spo;
```

```
        (* Indicates that a SPO happened not due to CCF propagation *)
        occ_spo_no_ccf;
        (* Indicates that a CCF can happen during the next occurrence rules *)
        ccf_spo_enabled;

    INTERACTION
        (*      test_send_order *)
        (*      ETAPE i_and_c_propagation *)
        (*      ALORS send_order <-- (comp_attr_diff OU send_spurious_order);
*)
        test_cond_occ_so
        GROUPE simu_group
        ETAPE saving_step
        ALORS cond_occ_so <-- MARCHE ET send_spurious_order;

        test_cond_occ_spo
        GROUPE simu_group
        ETAPE saving_step
        ALORS cond_occ_spo <-- MARCHE;

        reset_send_order
        GROUPE simu_group
        ETAPE reset_step
        SI send_order
        ALORS send_order <-- FAUX;

        (* cf. attribute f_SPO_previous *)
        save_f_SPO_previous
        GROUPE simu_group
        ETAPE saving_step
        ALORS f_SPO_previous <-- send_spurious_order;
```

```
        (* cf. effect occ_spo_no_ccf *)
        test_occ_spo
        GROUPE simu_group
ETAPE fluid_propagation
        ALORS occ_spo_no_ccf <-- send_spurious_order ET (NON f_SPO_previous);

        (* cf. stop_ccf_spos_propagation *)
        test_stop_ccf_spo_propagation
        GROUPE simu_group
ETAPE default_step
        SI stop_ccf_spo_propagation ALORS
        occ_spo_no_ccf <-- FAUX, stop_ccf_spo_propagation <-- FAUX;

        (* cf. ccf_spo_enabled *)
        test_ccf_spo_enabled
        GROUPE simu_group
        ETAPE default_step
        ALORS ccf_spo_enabled <-- cond_occ_spo;


OCCURRENCE
        occ_FSO
        GROUPE simu_group
        SI cond_occ_so ET gamma_fso > global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        TRANSITION no_f_FSO
        LOI INS (1 - gamma_fso)
        PROVOQUE send_order <-- VRAI, send_spurious_order <-- FAUX
        OU_BIEN
        TRANSITION f_FSO
        PROVOQUE send_order <-- FAUX, send_spurious_order <-- FAUX;

        occ_SO
```

```
                GROUPE simu_group
                SI cond_occ_so ET gamma_fso <= global_param_threshold_min
                IL_PEUT_SE_PRODUIRE
                TRANSITION no_f_FSO
                LOI INS (1)
                PROVOQUE send_order <-- VRAI, send_spurious_order <-- FAUX;

                occ_SPO
                GROUPE simu_group
                SI cond_occ_spo ET lambda_spo > global_param_threshold_min
                IL_PEUT_SE_PRODUIRE
                TRANSITION f_SPO
                LOI EXP (lambda_spo)
                PROVOQUE send_spurious_order <-- VRAI;

        occ_SPO_ccf
                GROUPE simu_group
        SI cond_occ_spo ET lambda_spo > global_param_threshold_min ET ccf_force_spo
        IL_PEUT_SE_PRODUIRE
        TRANSITION f_SPO
            LOI INS (1)
                PROVOQUE send_spurious_order <-- VRAI,
                ccf_force_spo <-- FAUX, stop_ccf_spo_propagation <-- VRAI;

(* Classe =IandCBaseModel= *)

(* Generic I&C model *)
(* ------------------- *)
TYPE IandCBaseModel SORTE_DE InstrumentationAbstract ControlAbstract;

        INTERACTION
                test_cond_occ_so
                GROUPE simu_group
```

```
                ETAPE saving_step
                ALORS cond_occ_so <-- MARCHE ET (comp_attr_diff OU send_spurious_order);


        OCCURRENCE
                occ_FSO
                GROUPE simu_group
                SI cond_occ_so ET gamma_fso > global_param_threshold_min
                IL_PEUT_SE_PRODUIRE
                TRANSITION no_f_FSO
                LOI INS (1 - gamma_fso)
                PROVOQUE send_order <-- VRAI,
                send_spurious_order <-- FAUX, comp_attr_diff <-- FAUX
                OU_BIEN
                TRANSITION f_FSO
                PROVOQUE send_order <-- FAUX,
                send_spurious_order <-- FAUX, comp_attr_diff <-- FAUX;

                occ_SO
                GROUPE simu_group
                SI cond_occ_so ET gamma_fso <= global_param_threshold_min
                IL_PEUT_SE_PRODUIRE
                TRANSITION no_f_FSO
                LOI INS (1)
                PROVOQUE send_order <-- VRAI,
                send_spurious_order <-- FAUX, comp_attr_diff <-- FAUX;

(* Classe =CommonLogicCompactModel= *)

(* Compact model *)
(* ---------------- *)
TYPE CommonLogicCompactModel;
```

```
CONSTANTE
        gamma_cm_fcl DOMAINE REEL PAR_DEFAUT 1e-100
        LIBELLE "Probability of common logic failure";

ATTRIBUT
        init_occurred DOMAINE BOOLEEN PAR_DEFAUT FAUX;

PANNE
        f_FCMCL;

OCCURRENCE
        occ_FCMCL
        GROUPE simu_group
        SI NON init_occurred ET
        global_maint_step_finished ET gamma_cm_fcl > global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        DEFAILLANCE f_FCMCL
        LOI INS ( gamma_cm_fcl )
        PROVOQUE init_occurred <-- VRAI
        OU_BIEN
        TRANSITION no_f_FCMCL
        PROVOQUE init_occurred <-- VRAI;

        occ_CMCL
        GROUPE simu_group
        SI NON init_occurred ET
        global_maint_step_finished ET gamma_cm_fcl <= global_param_threshold_min
        IL_PEUT_SE_PRODUIRE
        TRANSITION no_f_FCMCL
        LOI INS (1)
        PROVOQUE init_occurred <-- VRAI;

(* Classe =IandCCompactModelAuto= *)
```

```
TYPE IandCCompactModelAuto SORTE_DE IandCBaseModel;

        CONSTANTE
                nb_ac_trains DOMAINE ENTIER PAR_DEFAUT 1
                LIBELLE "Number of independant acquisition trains";
                gamma_cm_fac DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability of non acquisition";
                (* Computed from gamma_cm_ac and nb_ac_trains *)
                gamma_fac DOMAINE REEL PAR_DEFAUT gamma_cm_fac**nb_ac_trains
                LIBELLE "Probability of false acquisition";

                nb_spl_trains DOMAINE ENTIER PAR_DEFAUT 1
                LIBELLE "Number of independant specific logical trains";
                gamma_cm_fspl DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability of specific logic failure";
                gamma_fso DOMAINE REEL PAR_DEFAUT gamma_cm_fspl**nb_spl_trains
                LIBELLE "Probability of send order failure";


        INTERFACE
                common_logic GENRE CommonLogicCompactModel CARDINAL 1 JUSQUA 1;


        INTERACTION
                test_cond_occ_so
                GROUPE simu_group
                ETAPE saving_step
                ALORS cond_occ_so <-- MARCHE ET MARCHE( common_logic )
                ET (comp_attr_diff OU send_spurious_order);

(* Classe =IandCCompactModelAutoWithBattery= *)
```

```
(* - Hypothèses : *)
(*    1. Lorsque la batterie est sollicitée une fois, son arrêt n'est
(*       jamais demandé même si le composant est de nouveau alimenté par
                le circuit normal. La gestion de l'arrêt de la batterie
                pose des problèmes lorsqu'une batterie est associée à
                plusieurs composants. Supposons par exemple qu'une batterie B
                est en secours des composant X et Y. Si X sollicite la
                batterie et que Y est toujours alimenté, alors le démarrage de
                la batterie aura bien lieu mais cette dernière sera stoppée
                (* aussitôt du fait que Y est déjà alimenté. *)

TYPE IandCCompactModelAutoWithBattery SORTE_DE IandCCompactModelAuto ElecConsumer;


        INTERFACE
                battery GENRE Battery CARDINAL 1 JUSQUA 1;

        INTERACTION

        test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET (linked_comp_flowed OU fed( battery ));

                (* The OR condition is used to stop the battery when it is not needed anymo
        test_battery_is_requested
                GROUPE simu_group
                ETAPE fluid_propagation
                SI (NON is_started( battery )
                ET NON linked_comp_flowed) (* OU (is_started( battery )
                ET linked_comp_flowed) to stop the battery but not use *)
                ALORS is_requested( battery ) <-- VRAI;
```

```
                test_cond_occ_so
                GROUPE simu_group
                ETAPE saving_step
                ALORS cond_occ_so <-- fed ET MARCHE ET MARCHE( common_logic )
                ET (comp_attr_diff OU send_spurious_order);

(* Classe =IandCCompactModelManual= *)

(* To handle operator action *)
TYPE IandCCompactModelManual SORTE_DE IandCBaseModel;

        INTERFACE
                common_logic GENRE CommonLogicCompactModel CARDINAL 1 JUSQUA 1;

        INTERACTION
                test_cond_occ_so
                GROUPE simu_group
                ETAPE saving_step
                ALORS cond_occ_so <-- MARCHE ET MARCHE( common_logic )
                ET (comp_attr_diff OU send_spurious_order);

(* Classe =IandCCompactModelManualWithBattery= *)

(* - Hypothèses : *)
(*   1. Lorsque la batterie est sollicitée une fois, son arrêt n'est
                jamais demandé même si le composant est de nouveau alimenté par
                le circuit normal. La gestion de l'arrêt de la batterie
                pose des problèmes lorsqu'une batterie est associée à
                plusieurs composants. Supposons par exemple qu'une batterie B
                est en secours des composant X et Y. Si X sollicite la
                batterie et que Y est toujours alimenté, alors le démarrage de
                la batterie aura bien lieu mais cette dernière sera stoppée
                aussitôt du fait que Y est déjà alimenté. *)
```

```
TYPE IandCCompactModelManualWithBattery SORTE_DE IandCCompactModelManual ElecConsumer;

        INTERFACE
                battery GENRE Battery CARDINAL 1 JUSQUA 1;

        INTERACTION

        test_fed
                GROUPE simu_group
                ETAPE fluid_propagation
        ALORS fed <-- MARCHE ET (linked_comp_flowed OU fed( battery ));

                (* The OR condition is used to stop the battery when it is not needed anymo
        test_battery_is_requested
                GROUPE simu_group
                ETAPE fluid_propagation
                SI (NON is_started( battery ) ET NON linked_comp_flowed)
                ALORS is_requested( battery ) <-- VRAI;

                test_cond_occ_so
                GROUPE simu_group
                ETAPE saving_step
                ALORS cond_occ_so <-- fed ET MARCHE ET MARCHE( common_logic )
                ET (comp_attr_diff OU send_spurious_order);

(* Classe =PrevMaintGroup= *)

TYPE PrevMaintGroup;

        ATTRIBUT
                init DOMAINE BOOLEEN PAR_DEFAUT VRAI;
```

```
        CONSTANTE
                duration_upm DOMAINE REEL PAR_DEFAUT 1e100
                LIBELLE "Duration time of preventive maintenance operations";
                gamma_upm DOMAINE REEL PAR_DEFAUT 1e-100
                LIBELLE "Probability of unavailability due to preventive maintenance";

(* Classes =PrevMaintGroup[N]= *)

TYPE PrevMaintGroup1 SORTE_DE PrevMaintGroup;

        INTERFACE
                comp_1 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

        OCCURRENCE
                occ_select_maint_comp
                GROUPE simu_group
                SI init ET global_maint_enabled
                IL_PEUT_SE_PRODUIRE
                TRANSITION select_maint_comp_1
                LOI INS (1)
                PROVOQUE is_selected_upm( comp_1 ) <-- VRAI,
                init <-- FAUX, duration_upm( comp_1 ) <-- duration_upm,
                gamma_upm( comp_1 ) <-- gamma_upm;

TYPE PrevMaintGroup2 SORTE_DE PrevMaintGroup1;

        INTERFACE
                comp_2 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

        OCCURRENCE
                occ_select_maint_comp
                GROUPE simu_group
```

```
        SI init ET global_maint_enabled
        IL_PEUT_SE_PRODUIRE
        TRANSITION select_maint_comp_1
        LOI INS (1/2)
        PROVOQUE is_selected_upm( comp_1 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_1 ) <-- duration_upm,
        gamma_upm( comp_1 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_2
        LOI INS (1/2)
        PROVOQUE is_selected_upm( comp_2 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_2 ) <-- duration_upm,
        gamma_upm( comp_2 ) <-- gamma_upm;


TYPE PrevMaintGroup3 SORTE_DE PrevMaintGroup2;

    INTERFACE
        comp_3 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

    OCCURRENCE
        occ_select_maint_comp
        GROUPE simu_group
        SI init ET global_maint_enabled
        IL_PEUT_SE_PRODUIRE
        TRANSITION select_maint_comp_1
        LOI INS (1/3)
        PROVOQUE is_selected_upm( comp_1 ) <-- VRAI, init <-- FAUX,
        duration_upm( comp_1 ) <-- duration_upm,
        gamma_upm( comp_1 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_2
        LOI INS (1/3)
```

```
        PROVOQUE is_selected_upm( comp_2 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_2 ) <-- duration_upm,
        gamma_upm( comp_2 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_3
        LOI INS (1/3)
        PROVOQUE is_selected_upm( comp_3 ) <-- VRAI, init <-- FAUX,
        duration_upm( comp_3 ) <-- duration_upm,
        gamma_upm( comp_3 ) <-- gamma_upm;


TYPE PrevMaintGroup4 SORTE_DE PrevMaintGroup3;

    INTERFACE
        comp_4 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

    OCCURRENCE
        occ_select_maint_comp
        GROUPE simu_group
        SI init ET global_maint_enabled
        IL_PEUT_SE_PRODUIRE
        TRANSITION select_maint_comp_1
        LOI INS (1/4)
        PROVOQUE is_selected_upm( comp_1 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_1 ) <-- duration_upm,
        gamma_upm( comp_1 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_2
        LOI INS (1/4)
        PROVOQUE is_selected_upm( comp_2 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_2 ) <-- duration_upm,
        gamma_upm( comp_2 ) <-- gamma_upm
        OU_BIEN
```

```
        TRANSITION select_maint_comp_3
        LOI INS (1/4)
        PROVOQUE is_selected_upm( comp_3 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_3 ) <-- duration_upm,
        gamma_upm( comp_3 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_4
        LOI INS (1/4)
        PROVOQUE is_selected_upm( comp_4 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_4 ) <-- duration_upm,
        gamma_upm( comp_4 ) <-- gamma_upm;


TYPE PrevMaintGroup5 SORTE_DE PrevMaintGroup4;

    INTERFACE
        comp_5 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

    OCCURRENCE
        occ_select_maint_comp
        GROUPE simu_group
        SI init ET global_maint_enabled
        IL_PEUT_SE_PRODUIRE
        TRANSITION select_maint_comp_1
        LOI INS (1/5)
        PROVOQUE is_selected_upm( comp_1 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_1 ) <-- duration_upm,
        gamma_upm( comp_1 ) <-- gamma_upm
        OU_BIEN
        TRANSITION select_maint_comp_2
        LOI INS (1/5)
        PROVOQUE is_selected_upm( comp_2 ) <-- VRAI,
        init <-- FAUX, duration_upm( comp_2 ) <-- duration_upm,
```

```
                gamma_upm( comp_2 ) <-- gamma_upm
                OU_BIEN
                TRANSITION select_maint_comp_3
                LOI INS (1/5)
                PROVOQUE is_selected_upm( comp_3 ) <-- VRAI,
                init <-- FAUX, duration_upm( comp_3 ) <-- duration_upm,
                gamma_upm( comp_3 ) <-- gamma_upm
                OU_BIEN
                TRANSITION select_maint_comp_4
                LOI INS (1/5)
                PROVOQUE is_selected_upm( comp_4 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_5
                LOI INS (1/5)
                PROVOQUE is_selected_upm( comp_5 ) <-- VRAI, init <-- FAUX, duration_upm( co


TYPE PrevMaintGroup6 SORTE_DE PrevMaintGroup5;

        INTERFACE
                comp_6 GENRE ComponentAbstract CARDINAL 1 JUSQUA 1;

        OCCURRENCE
                occ_select_maint_comp
                GROUPE simu_group
                SI init ET global_maint_enabled
                IL_PEUT_SE_PRODUIRE
                TRANSITION select_maint_comp_1
                LOI INS (1/6)
                PROVOQUE is_selected_upm( comp_1 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_2
                LOI INS (1/6)
```

```
                PROVOQUE is_selected_upm( comp_2 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_3
                LOI INS (1/6)
                PROVOQUE is_selected_upm( comp_3 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_4
                LOI INS (1/6)
                PROVOQUE is_selected_upm( comp_4 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_5
                LOI INS (1/6)
                PROVOQUE is_selected_upm( comp_5 ) <-- VRAI, init <-- FAUX, duration_upm( co
                OU_BIEN
                TRANSITION select_maint_comp_6
                LOI INS (1/6)
                PROVOQUE is_selected_upm( comp_6 ) <-- VRAI, init <-- FAUX, duration_upm( co
(*
Figaro CCF group of order 2 - model alpha

The following Figaro code describe a Common Cause Failure (CCF) group of order 2 in the mod

To work, the following assumption is made on the target knowledge base :
- The class of the objects in the CCF group is FailureToRun.
- The class FailureToRun has an attribute (or an effect) named ccf_force_fr used to force th
  In other word, we should find something like this in the attribute block of class FailureT

  ccf_force_fr DOMAINE BOOLEEN PAR_DEFAUT FAUX;

  And an occurrence rule like this :

  GROUPE simu_group
  SI <failure condition> ET ccf_force_fr
```

```
  IL_PEUT_SE_PRODUIRE
  DEFAILLANCE <failure_name>
  LOI INS (1)
  PROVOQUE [...],
           ccf_force_fr <-- FAUX;

- An effect (or attribute) named ccf_fr_enabled is supposed to exist in class FailureToRun t
  In most situations, the attribute ccf_fr_enabled merely indicates that an instance is not
- An effect (or attribute) named occ_fr_no_ccf is supposed to exist in class FailureToRun to
- Regarding the previous rule, be sure to stop de CCF propagation when the failure of an ob
*)
TYPE CCFailureToRunAlpha2;

    ATTRIBUT
        (* Indicate if a FR on comp 1 has propagated a CCF on comp 2 *)
        ccf_comp_1_to_2 DOMAINE BOOLEEN
        EDITION NON VISIBLE, NON MODIFIABLE
        PAR_DEFAUT FAUX;

        (* Indicate if a FR on comp 2 has propagated a CCF on comp 1 *)
        ccf_comp_2_to_1 DOMAINE BOOLEEN
        EDITION NON VISIBLE, NON MODIFIABLE
        PAR_DEFAUT FAUX;



    CONSTANTE

        (* Alpha factor CCF model *)

        alpha_1 DOMAINE REEL
        EDITION VISIBLE, MODIFIABLE
        PAR_DEFAUT 0.5;
```

```
alpha_2 DOMAINE REEL
EDITION VISIBLE, NON MODIFIABLE
PAR_DEFAUT 1 - alpha_1;

alpha_t DOMAINE REEL
EDITION VISIBLE, NON MODIFIABLE
PAR_DEFAUT alpha_1 + 2*alpha_2;

alpha_beta_1_2 DOMAINE REEL
EDITION VISIBLE, NON MODIFIABLE
PAR_DEFAUT (1.0*alpha_1)/alpha_t;

alpha_beta_2_2 DOMAINE REEL
EDITION VISIBLE, NON MODIFIABLE
PAR_DEFAUT (2.0*alpha_2)/alpha_t;
```

INTERFACE

```
(* CCF group consisting of FailureToRun objects *)

comp_1 GENRE FailureToRun CARDINAL 1
EDITION VISIBLE, MODIFIABLE;

comp_2 GENRE FailureToRun CARDINAL 1
EDITION VISIBLE, MODIFIABLE;
```

INTERACTION

```
(* CCF propagation from comp 1 to comp 2 *)
propagation_ccf_comp_1_to_2
GROUPE simu_group
ETAPE ccf_propagation
SI ccf_comp_1_to_2 ALORS
ccf_force_fr( comp_2 ) <-- VRAI,
ccf_comp_1_to_2 <-- FAUX;

(* CCF propagation from comp 2 to comp 1 *)
propagation_ccf_comp_2_to_1
GROUPE simu_group
ETAPE ccf_propagation
SI ccf_comp_2_to_1 ALORS
ccf_force_fr( comp_1 ) <-- VRAI,
ccf_comp_2_to_1 <-- FAUX;
```

OCCURRENCE

```
(* Propagation options for a CCF from comp 1 to possibly comp 2 *)
occ_ccf_comp_1_to_2
GROUPE simu_group
SI occ_fr_no_ccf( comp_1 ) ET ccf_fr_enabled( comp_2 )
IL_PEUT_SE_PRODUIRE
TRANSITION ccf_comp_1_to_2
LIBELLE "CCF propagation from comp 1 to comp 2"
LOI INS ( alpha_beta_2_2 )
PROVOQUE ccf_comp_1_to_2 <-- VRAI
OU_BIEN
TRANSITION independent_fr_comp_1
LIBELLE "No CCF propagation from comp 1";

(* Propagation options for a CCF from comp 2 to possibly comp 1 *)
```

```
occ_ccf_comp_2_to_1
GROUPE simu_group
SI occ_fr_no_ccf( comp_2 ) ET ccf_fr_enabled( comp_1 )
IL_PEUT_SE_PRODUIRE
TRANSITION ccf_comp_2_to_1
LIBELLE "CCF propagation from comp 2 to comp 1"
LOI INS ( alpha_beta_2_2 )
PROVOQUE ccf_comp_2_to_1 <-- VRAI
OU_BIEN
TRANSITION independent_fr_comp_2
LIBELLE "No CCF propagation from comp 2";
```

Bibliography

[1] R. Manian, D. W. Coppit, K. J. Sullivan, J. Bechta Dugan, "*Bringing the gap between systems and dynamic fault tree models*", In Proceedings IEEE Annual Reliability and Maintainability Symposium, pages 105–111. IEEE Computer Society Press, Washington, DC, 1999.

[2] A. Bobbio, D. Codetta-Raiteri, "*Parametric Fault Trees with Dynamic Gates and Repair Boxes*", Proc. of the Annual Reliability and Maintainability Symposium, pages 459-465, Los Angeles, January 2004.

[3] *Dynamic Event Trees in Accident Sequences Analysis: Application to Steam Generator Tube Rupture*". Acosta, C., Siu, .N. 1993, Reliability Engineering and System Safety, pp. 135-154.

[4] *Fault Tree Models for the Analysis of Complex Computer-Based Systems.* Pullam, L., Dugan, J. 1996, Proceedings: Annual Reliability and Maintainability Symposium. 6. The Development and Application of the Accident Dynamic Simulator for Dynamic Probabilistic Risk Assessment of Nuclear Power Plants. Hsueh, K., Mosleh, A. 1996, Reliability Engineering and System Safety, Vol. 52, pp. 279-296. 465.

[5] Bouissou M., Humbert S., Muffat S., Villatte N., *KB3 tool : Feedback on knowledge bases*, proceedings of Lambda Mu 13 / ESREL 2002, European Conference, Lyon (France), pp.754-759, March 2002.

[6] *Dynamic Event Trees for Probabilistic Analysis.* Hofer, E., et. al. Germany : GRS, 2000.

[7] *A Methodology for Generating Dynamic Accident Progression Event Trees for Level-2 PRA.* Hakobyan, A., Denning, R., Aldemir, T. September 2006, Proceedings of PHYSCOR 2006.

[8] Hakobyan, A. *Severe Accident Analysis Using Dynamic Accident Progression Trees.* The Ohio State University. 2006. PhD Thesis.

[9] *Dynamic logical analytical methodology versus fault tree: The case of auxiliary feedwater system of a nuclear power plant.* Cacciabue, P.C., Amendola, A., Cojazzi, G. 1986, Nuclear Technology, Vol. 74, pp. 195-208.

[10] *Computer-Assisted Markov Failure Modeling of Process Control Systems.* Aldemir, T. 1987, IEEE Transactions on Reliability, Vols. R-36, pp. 133-144.

[11] *The Development and Application of the Accident Dynamic Simulation for Dynamic Probabilistic Risk Assessment of Nuclear Power Plants.* Kae-Sheng, H., Mosleh, A. 1996, Reliability Engineering and System Safety, Vol. 52, pp. 297-314.

[12] *Dynamic Generation of Accident Pgression Trees.* Hkobyan, A., Aldemir, T., Denning, R., Dunagan, S., Kunsman, D., Rutt, B., Catalyurek, U. 2008, Nuclear Engineering and Design, Vol. 238, pp. 3457-3467.

[13] Aldemir et al., A Benchmark Implementation pf Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation And Control. 2009. NUREG/CR-6985.

[14] *Hinckley point C Pre Construction Safety Report - Sub Chapter 15.1 Level 1 PSA.* Version 2.0 Date of Issue 31st August 2012 Document No. HPC-NNBOSL-U0-000-RES-000033 Produced by (Company/Organisation) NNB GenCo.

[15] Marc Bouissou, Jean-Louis Bon. *A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes.* 2003, Reliability Engineering and System Safety, Vol. 82, pp.149-163.

[16] Gallois M., Pilli'ere M., *Benefits expected from automatic studies with KB3 in PSAs at EDF*, proceedings of the PSA99 conference, Washington, August 1999.

[17] Bouissou M., Bouhadana H., Bannelier M., Villatte N., *Knowledge modeling and reliability processing: presentation of the FIGARO language and associated tools*, proceedings of SAFECOMP'91, Trondheim (Norway), November 1991.

[18] M. Marseguerra E. Zio, J. Devooght, P.E. Labeau. *A concept paper on dynamic reliability via Monte Carlo simulation. 1998, Mathematics and Computers in Simulation*, Vol. 47, pp. 371-382.

[19] S.A. Eide et al., *Reevaluation of Station Blackout Risk at Nuclear Power Plants*, U.S. Nuclear Regulatory Commission, NUREG/CR-6890, December 2005.

[20] *A Comparison of Dynamic and Classical Event Tree Analysis for Nuclear Power Plant Probabilistic Safety/Risk Assessment.* Kyle Gardner Metzroth, Graduate Program in Nuclear Engineering. The Ohio State University,2011.

[21] ENFCFF090154-A, *EPS-EPR Phase 2 - Modèle de fiabilité de la distribution électrique de l'EPR FA3.* GENEVOIS P. pp.0-78.