Optimal design of dependable control system architectures using temporal sequences of failures

Joffrey Clarhaut, Blaise Conrard, Saïd Hayat, Vincent Cocquempot

IEEE Transactions on Reliability, Vol 58 Issue 3, Sept. 2009, pp 511 - 522
Doi: 10.1109/TR.2009.2026790

http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=24

Optimal design of dependable control system architectures using temporal sequences of failures

Joffrey Clarhaut, Blaise Conrard, Said Hayat and Vincent Cocquempot*

Key Words—**Dependability, Operational architecture, Railroad transportation systems, System design, sequences of failures**

Summary and Conclusions

Summary— **Designing a dependable control system requires accurate methods to evaluate efficiently the dependability level of one given component architecture. This evaluation is crucial in order to determine the risks associated with system failures and the remaining properties after the fault occurrence. The dependability level of a control system depends not only on the kind of component failures that may occur but also on the ordered sequences of the failure appearance. Classical evaluation methods, i.e. Fault Trees or Failure Mode and Effect Analysis, are not appropriate to handle these sequences. That paper contributes on this aspect and proposes a complete design methodology for dependable systems. This methodology uses ordered sequences of multiple failures to evaluate accurately the dependability level of all possible system's equipments architectures. Starting with the hierarchical functional decomposition of the system, the first step is to identify the dreaded events. Thus, the faulty behaviors of all possible system architectures are characterized with temporal operators. The set of system's operational architectures is finally determined by solving an optimization problem that considers both dependability objectives and cost constraint. This methodology is applied to design a fire detection system for a railroad transportation system.**

Conclusions — **In this paper, a complete methodology to design dependable control systems is presented. The innovative feature of this methodology is that it attempts to take into account time ordered sequences of**

This work is supported by the French Regional Council of "Nord – Pas de Calais" and by the INRETS institute.

Joffrey Clarhaut is with the French National Institute for Transport and Safety Research (INRETS), 20, Rue Elisée Reclus, BP 317, 59666 Villeneuve d'ascq Cedex, France (phone: +33 320438407; fax: +33 320438398; e-mail: joffrey.clarhaut@inrets.fr).

Blaise Conrard is with the Automatic Control Laboratory of Lille University (LAGIS), UMR CNRS 8146, Bat. Polytech'Lille, Cité Scientifique, 59655 Villeneuve d'ascq Cedex, France (e-mail: blaise.conrard@polytech-lille.fr).

Said Hayat is with the French National Institute for Transport and Safety Research (INRETS), 20, Rue Elisée Reclus, BP 317, 59666 Villeneuve d'ascq Cedex, France (e-mail: said.hayat@inrets.fr).

Vincent Cocquempot is with the Automatic Control Laboratory of Lille University (LAGIS), UMR CNRS 8146, Bat. Polytech'Lille, Cité Scientifique, 59655 Villeneuve d'ascq Cedex, France (e-mail: vincent.cocquempot@univ-lille1.fr). * Corresponding author.

failures. A new representation, called improved multi-fault tree, is defined. This tool allows first to model failure relationships between functions and second to evaluate the dependability level of a set of equipment architectures by the use of time ordered sequences of failures. Our design method provides a set of optimal architectures with given cost and dependability level. The designer can choose among these solutions depending on the cost and dependability level specifications. The comparison between the new approach and the classical dependability method shows that the set of solutions for the multi-fault tree is smaller than the set of solutions for the classical one. The set is smaller but the solutions are better because the new approach integrates temporal functions and evaluates more precisely the level of dependability than with the traditional one. Future work will concern the enhancement of the comparison algorithm in order to be able to design more complex distributed systems, i.e. with a great number of functions and components and with shared functions.

I. INTRODUCTION

A control system is composed of physical components, sensors, and actuators which are organized in order to achieve a set of missions, not only in normal situation but also in faulty cases. In the design stage, the objective of the designer is to find a feasible architecture that guarantees an acceptable level of dependability [1]. This level of dependability is evaluated either in a quantitative way using failure rates [2] or using qualitative characteristics such as the set of failures that leads to a given dreaded event [3]. Our work is concerned with the second kind of approach. Faulty system's behaviors depend not only on the system's functions that are affected by the faulty components but also on the temporal ordered sequence of failures appearance. Fault trees or Failure Mode and Effect Analysis (FMEA) are classical methods that are used to evaluate and analyze qualitative dependability characteristics. However, they are not well appropriate to take into account these sequences of failures [4]–[5]. Other methods based on Monte Carlo techniques, on Petri Nets and Markov Graphs may be used to take into account such temporal sequences, however they present some limits like a long simulation time and an exponential increase of combinatorial states or places [6]–[7]–[8]. As a consequence, these methods, which are developed to assess one existing physical system, are not well adapted to design such control system architecture. Indeed, designing a system

requires fast evaluation tools and optimization algorithms in order to find the best architecture of components among a set of numerous possible architectures.

This paper presents a novel design methodology for dependable control systems. Our method takes explicitly into account the ordered sequences of failures. A graphical representation, called "improved multi-fault tree" is provided. This tool allows to represent all possible systems architectures that fulfill the dependability specifications. Improved multi-fault trees are derived from classical fault trees by adding temporal operators that are useful to represent the temporal constraints on the system faulty behaviors. The dependability level of all possible systems architectures may be quickly and accurately evaluated using this new tree.

The design methodology is divided into two phases: the first phase aims at modeling dependable control systems by taking into account the ordered sequences of failures, while the second one allows to obtain the set of possible system's operational architectures that fulfill dependability objectives for a given cost.

This methodology is applied to design a safety system for a railroad wagon. This railroad wagon presents significant risks for itself, like derailment and fire, because of its transported goods, i.e. trucks, and needs safety systems to improve its dependability level. It has been shown in [9]-[10] that a fire detection system allows to increase the dependability level of the railroad wagon. In this paper we focus on the design of such fire detection system.

The rest of this paper is divided into three parts. In section II, general concepts related to the design of dependable systems are summarized and the contribution is pointed out clearly. In section III, the design methodology is presented. Finally in section IV, the methodology is applied to design a fire detection system for a railroad transportation system. A comparison with a classical method is provided that shows clearly the efficiency and the benefit of the proposed approach.

II. General Concepts related to the design of dependable control systems

A. Control systems design methodology

The design of a control system involves three kinds of architectures [11]–[12]–[13]: the functional architecture, the equipment architecture and the operational architecture.

The *functional architecture* is built according to the functional specifications (Appendix I, Fig. 1, activity A1) and represents the links between the system's functions. This model is based on a hierarchical functional decomposition :

main functions are broken up into sub-functions and so on. This decomposition is achieved when basic components are reached (Fig. 1, activity A2).

The *equipment architecture* represents the system equipments list. This architecture reflects the choices made for the equipments (Fig. 1, activity A3). These choices are improved in order to satisfy dependability criteria (Fig. 1, activity A5). The equipment architecture may contain equipments like hardware and software components as well as communication systems.

The *operational architecture* is defined as the projection, using several constraints, of a functional architecture on one equipment architecture (Fig. 1, activity A4). The operational architecture takes into account constraints linked to the dependability level and to the cost objective (Fig. 1, activity A6). If the objectives are not met, the feedback loop between activities A6 and A1 provides information that can be used to improve system performances like modification of functions or integration of redundant components.

In the projection step (Fig. 1, activity A4) the evaluation of the dependability level is either static or dynamic. The static evaluation is a probabilistic evaluation of dependability characteristics. The dynamic evaluation considers the temporal ordered sequences of failures which are called scenarios. This dynamic evaluation allows to determine all the scenarios bringing the system to a precise dangerous situation.

B. Dependability characteristics

Dependability is classically defined as the *science of failures* [14]. It is characterized by the system's failure analysis and their consequences. The four main characteristics of dependability are [15]–[16]–[17]:

- Reliability: Ability of an entity to achieve a required function, under given conditions, during a given time interval.
- Availability: Ability of an entity to be in a state to perform a required function under given conditions at a given instant of time, assuming that the required external resources are provided.
- Maintainability: Ability of an entity to be repaired within a given time interval, in a state in which it can achieve a necessary function.
- Safety: Ability to avoid critical or catastrophic events.

A uniform way to consider availability, reliability and safety consists in using the concept of dreaded event [18]. For instance the events "incapacity to achieve its mission", "shutdown" or "dangerous behavior" are respectively

related to each one of these dependability characteristics. Dreaded events allow to consider dependability characteristics in a qualitative but more comprehensive way.

A dependable system [19] is a system that carries out what it was designed for, without incident infringing its availability and without accident endangering its safety. The first aspect is based on the concept of availability so that a system must be able to achieve its task in an awaited way. The second aspect is based on the concept of safety so that a system must avoid any risk situation for its environment and its operators. Fig. 2 (Appendix 1) shows that these four dependability aspects are not independent [20]. Designing a dependable system means finding the best availability-reliability-safety-maintainability combination.

C. Dependability evaluation methods

The evaluation of the system's dependability characteristics can be carried out by several methods. Many research books and papers [21]–[22] and International Standards dedicated to dependability [23] are devoted to these methods.

Fault trees and reliability block diagrams are classical static methods that are focused on a probabilistic evaluation of dependability characteristics. These methods allow to isolate the components and the system's parts that are sensitive to failures. These methods do not take into account ordered sequences of failures and they do not consider temporal dependencies between functions [5]. As it will be shown later, these temporal aspects are very important in the accurate evaluation of dependability characteristics of an equipment architecture.

Other methods based on Petri nets or Markov graphs, aim at including these temporal characteristics. These models can integrate various failure rates or transitions according to the states of a system. They also take into account repairable states of system components and temporal dependencies between functions. However, these models are often not easy to construct and to study. They require efficient simulations methods and suffer from a combinatorial explosion of their states [6]–[7].

The idea to combine both static and dynamic methods to solve construction and exploitation problems and to reduce the combinatorial explosion problem was studied in a number of ways:

- Dugan [24] developed the concept of dynamic fault tree. This tree has the ability to capture sequence dependencies in operational systems. It can be evaluated via a combination of Binary Decision Diagram and Markov processes.
- Bouissou [8]–[25] defined a new semantics for fault trees augmented with a new kind of link called "trigger" by

using Markov processes. This new formalism is called "Boolean Logic Driven Markov Process" (BDMP) ®. This formalism allows to reduce the combinatorial explosion of states when studying huge systems.

- Cepin and Mavko [26] extended a classical fault tree with a "house" events matrix and temporal function dependencies. The "house" events matrix models connections between gates and events of the fault tree. It also represents the system operation and its environment.

These methods allow to evaluate the dependability characteristics of one existing system component architecture. Our objective, which is to design a control system with given dependability specifications, is quite different. This motivates the development of a new static/dynamic model.

Our new model, called *improved multi-fault tree*, is based on classical fault trees. Nevertheless, it takes into account not only the sequences of multiple temporal ordered events but also the temporal dependencies between functions. This allows to evaluate accurately the dependability level. In fact, it is very important to consider the temporal order of fault events (also called failures) from the design stage because consequences on dependability characteristics are not the same if this order of events changes. Indeed, consequences change because the physical system under consideration has its own state evolution. Temporal dependencies between functions are represented with temporal operators. Furthermore, our tree is improved for the design purpose by adding alternative nodes and associative nodes. These nodes will show all possibilities of component combinations and allow to handle multiple system faulty behaviors.

III. PRESENTATION OF THE DESIGN METHODOLOGY

The main objectives of our design methodology are

- to model dependable control systems by taking into account the faulty scenarios,
- to evaluate accurately and efficiently the dependability level of an overall system,
- to obtain a set of realizable system's operational architectures, with each solution characterized by its dependability level and its global cost.

The design methodology is divided into two phases: the modeling phase and the optimization phase. These two phases will be defined in this section (parts B and C). Let us first give in the following subsection some useful definitions and evaluations of dependability characteristics.

A. Definitions and properties

This section defines and formalizes concepts and notions that will be used in the proposed design method. A scenario is first defined and its parameters are characterized. Operators of the improved multi-fault tree and their properties are also presented**.** The property of an operator is called here a composition law. Cost, dependability level and the set of systems Ω are also defined.

1) Failure, scenario and relative reliability coefficient

Definition 1: A *failure* is a non-desired event. It is generally associated with the transition of a component or a set of components from an acceptable state towards a non-desired state. In this non-desired state, it is supposed that the component does not achieve accurately its mission. It is supposed in the following that the system is not repairable, that is to say a component can't return after a failure to its initial state. From the system point of view, a *dreaded event* corresponds to a particular failure.

Definition 2: A *scenario* corresponds to a sequence of failures that brings the system to a precise dreaded event *D.* In other words, a scenario is a time-ordered set of failures denoted φ_D.

$$\varphi_D = \left[F_i^1, \cdots, F_j^n\right] \tag{1}$$

where F_α^β is the failure F_α which appears at position β in the scenario φ_D.

Definition 3: (**Scenario and set of scenarios**) Let Φ_D be the set of all scenarios leading the system to the dreaded event *D*. Let φ_D^i denote one element of Φ_D.

$$\Phi_D = \left\{\varphi_D^1, \cdots, \varphi_D^m\right\} \tag{2}$$

where φ_D^i denotes the i^{th} element of Φ_D.

Definition 4: (**Relative reliability coefficient**) The relative reliability coefficient, defined for a given failure F_i, and denoted $RRC(F_i)$, characterizes the probability that the failure F_i occurs when the system realizes one given mission.

In fact, RRC is introduced in order to differentiate two components that realize the same function but have different robustness levels. Classically as in references **[14], [15], [16]**, the reliability is defined by the following relation:

$$R = e^{-\int \lambda(t).dt}$$

where λ(t) is the failure rate.

Let consider the failure probability $\overline{R}(t) = 1 - R(t) = 1 - e^{-\int \lambda(t).dt}$

For a given failure mode F_i, the relation between the failure probability $\overline{R}_{ref}(t)$ of a reference component and the failure probability $\overline{R}(t)$ of a component which is characterized by $RRC(F_i)$ is:

$$\overline{R}(t) = (\overline{R}_{ref}(t))^{RRC(F_i)}$$

which leads to

$$RRC(F_i) = \frac{\ln\left(\overline{R}(t)\right)}{\ln\left(\overline{R}_{ref}(t)\right)}$$

Generally speaking, the RRC is a function of the duration of the considered mission and of the failure rate of the component.

$RRC(F_i) = 1$ refers to a reference (standard) component while $RRC(F_i) > 1$ refers to a component which is more robust (reliable) than a standard component. For instance, if $RRC(F_i)=2$ for a given failure mode of a component, the failure probability of this component is equal to the failure probability of two reference components.

The concept of RRC can be extended to a scenario and characterizes its occurrence probability in a qualitative way.

Definition 5: (**RRC of a scenario**) For a given scenario composed by a set of failures $F_i \quad i = 1...n$ with $n = \text{card}(\varphi_D)$, $RRC(\varphi_D)$ is the sum of $RRC(F_i)$ of each failure in φ_D.

$$RRC(\varphi_D) = \sum_{1 \le i \le card(\varphi_D)} RRC(F_i) \qquad (6)$$

Definition 6: (**RRC_{min} of a set of scenarios**) $RRC^D_{\min}$ for the set Φ_D is the minimal RRC of all scenarios contained in Φ_D.

$$RRC^D_{\min} = \min_{1 \le i \le card(\Phi_D)} RRC\left(\varphi^i_D\right) \qquad (7)$$

Given a particular dreaded event, this value (RRC) expresses a fictive number of failures. This fictive number corresponds to the equivalent number of failures that the system can tolerate before this event occurs.

Definition 7: (**Set of minimal scenarios of a set of scenarios**) The set of minimal scenarios of Φ_D, denoted Δ_D, is a subset of Φ_D that contains all the scenarios whose RRC is $RRC^D_{\min}$.

$$\Delta_D = \left\{\varphi^i_D \in \Phi_D / RRC(\varphi^i_D) = RRC^D_{\min}\right\} \qquad (8)$$

Definition 8: (**Number of combinations of a set of minimal scenario**) The number of scenarios contained in the set Δ_D is denoted $N_{\min}^{D}$. This value associated with $RRC_{\min}^{D}$ expresses the probability that the dreaded event *D* occurs.

$$\mathrm{N}_{\min}^{\mathrm{D}} = \mathrm{card}(\Delta_{\mathrm{d}}) \tag{9}$$

2) Operators for the relationships between failures

Fault trees characterize the relationships between different failures. Classically, two operators are used namely AND operator and OR operator. In order to take into account the ordered sequences of failures, it is necessary to add two other operators namely **PAND** operator and **SEQ** operator [27]–[28]. An improved multi-fault tree is thus obtained. In the modeling phase, these operators represent relations between different failures of functions, sub-functions and components in the improved multi-fault tree. They have also computational properties called composition laws that are applied for the treatment of the improved multi-fault tree in the optimization phase.

Let us consider below A, B two independent dreaded events, such that C results from the association of A and B with one of the operators. Δ_{A}, Δ_{B} and Δ_{C} are the minimal scenarios associated with A, B and C.

The AND operator allows to represent the case when the failure C occurs immediately after the occurrence of the 2 failures A and B.

Property 1: With C = A AND B, the parameters of C can be evaluated thanks to the following relations:

$$\mathrm{RRC}_{\min}^{\mathrm{C}} = \mathrm{RRC}_{\min}^{\mathrm{A}} + \mathrm{RRC}_{\min}^{\mathrm{B}} \tag{10}$$

$$N_{\min}^{C} = \frac{\left(RRC_{\min}^{B} + RRC_{\min}^{A}\right)!}{\left(RRC_{\min}^{B}! \times RRC_{\min}^{A}!\right)} \times N_{\min}^{B} \times N_{\min}^{A} \tag{11}$$

The OR operator is used when the failure C occurs immediately after the occurrence of the failure A or the failure B.

Property 2: With C = A OR B, the parameters of C can be evaluated thanks to the following relations:

$$\text{if } \mathrm{RRC}_{\min}^{\mathrm{A}} < \mathrm{RRC}_{\min}^{\mathrm{B}};\ \begin{cases} \mathrm{RRC}_{\min}^{\mathrm{C}} = \mathrm{RRC}_{\min}^{\mathrm{A}} \\ \mathrm{N}_{\min}^{\mathrm{C}} = \mathrm{N}_{\min}^{\mathrm{A}} \end{cases} \tag{12}$$

$$\text{if } \mathrm{RRC}_{\min}^{\mathrm{A}} = \mathrm{RRC}_{\min}^{\mathrm{B}};\ \begin{cases} \mathrm{RRC}_{\min}^{\mathrm{C}} = \mathrm{RRC}_{\min}^{\mathrm{A}} \\ \mathrm{N}_{\min}^{\mathrm{C}} = \mathrm{N}_{\min}^{\mathrm{A}} + \mathrm{N}_{\min}^{\mathrm{B}} \end{cases} \tag{13}$$

$$\text{if } \mathrm{RRC}^{\mathrm{A}}_{\min} > \mathrm{RRC}^{\mathrm{B}}_{\min};\ \begin{cases} \mathrm{RRC}^{\mathrm{C}}_{\min} = \mathrm{RRC}^{\mathrm{B}}_{\min} \\ \mathrm{N}^{\mathrm{C}}_{\min} = \mathrm{N}^{\mathrm{B}}_{\min} \end{cases} \quad (14)$$

Definition 9: (**PAND operator**) The PAND operator is a temporal operator and is used when the failure C occurs after the consecutive occurrence of A followed by B.

This operator is useful when the consequences of two component failures are different according to their order of occurrence. This case appears when a safety component is used and can not avoid a dangerous situation if it fails at first.

Property 3: With C = A **PAND** B, the parameters of C can be evaluated thanks to the following relations:

$$RRC^{C}_{\min} = RRC^{A}_{\min} + RRC^{B}_{\min} \quad (15)$$

$$N^{C}_{\min} = \frac{\left(RRC^{B}_{\min} + RRC^{A}_{\min} - 1\right)!}{\left(RRC^{B}_{\min}!\times\left(RRC^{A}_{\min} - 1\right)!\right)} \times N^{B}_{\min} \times N^{A}_{\min} \quad (16)$$

Definition 10: (**SEQ operator**) The SEQ operator is used when the failure C occurs after the consecutive occurrence of A followed by B. The difference is that the SEQ operator considers that no component failure leading to the occurrence of the dreaded event B, occurs before the occurrence of dreaded event A. This operator is useful when passive redundancies are used. In this case, the replacement function is idle and starts only when the main function fails, thus the failure of the replacement system may only occur after the failure of the main system.

Property 4: With C = A **SEQ** B, the parameters of C can be evaluated thanks to the following relations:

$$\mathrm{RRC}^{\mathrm{C}}_{\min} = \mathrm{RRC}^{\mathrm{A}}_{\min} + \mathrm{RRC}^{\mathrm{B}}_{\min} \quad (17)$$

$$\mathrm{N}^{\mathrm{C}}_{\min} = \mathrm{N}^{\mathrm{B}}_{\min} \times \mathrm{N}^{\mathrm{A}}_{\min} \quad (18)$$

3) Comparisons between dependability levels and between systems

Definition 11: (**Equivalent systems**) Two systems (or components) are equivalent if they can achieve the same function and as a consequence if the same dreaded events can be considered for these two systems.

Definition 12: (**Dependability level for a dreaded event**) For a system *S* and for the dreaded event *D*, let us denote DL^{S}_{D} the dependability level formed by the couple ($\mathrm{RRC}^{\mathrm{D,S}}_{\min}$, $\mathrm{N}^{\mathrm{D,S}}_{\min}$).

For a given system, this couple characterizes the probability that the dreaded event occurs. Thus, it can be used to compare several systems.

Definition 13: (**Comparison between dependability levels for a same dreaded event**) For two equivalent systems S_1 and S_2 and for the same dreaded event D, the dependability level of S_1 is said greater than the dependability level of S_2, which is denoted $DL_D^{S_1} > DL_D^{S_2}$, if the following relations are verified:

$$\begin{aligned} & RRC_{min}^{D,S_1} > RRC_{min}^{D,S_2} \\ \text{or} & \begin{cases} RRC_{min}^{D,S_1} = RRC_{min}^{D,S_2} \\ N_{min}^{D,S_1} < N_{min}^{D,S_2} \end{cases} \end{aligned} \tag{19}$$

This relation expresses that the dreaded event occurs with a less probability for S_1 than for S_2 due to:

- A higher number of failures in case of S_1.
- Or a reduced number of sequences even if the number of failures is the same for S_1 and S_2.

Definition 14: The dependability levels are said to be identical ($DL_D^{S_1} = DL_D^{S_2}$) if $RRC_{min}^{D,S_1} = RRC_{min}^{D,S_2}$ and $N_{min}^{D,S_1} = N_{min}^{D,S_2}$

Definition 15: (**Dependability level of a system**) For a system (or a component) associated with a number n of dreaded events Di ($i = 1..n$), the set of all $DL_{D_i}^S$ is denoted DL^S.

$$DL^S = \left\{ DL_{D_1}^S, \cdots, DL_{D_n}^S \right\} \tag{20}$$

This set expresses the ability of the considered system to tolerate the failures for various dreaded events.

Definition 16: (**Comparison between dependability levels of equivalent systems**) Let us consider two equivalent systems S_1 and S_2 and a set of dreaded events D. The dependability level DL^{S1} is said greater than DL^{S2} and is written $DL^{S1} > DL^{S2}$ if

$$\forall i \; DL_{D_i}^{S1} \geq DL_{D_i}^{S2} \text{ and } \exists j \; DL_{D_j}^{S1} > DL_{Dj}^{S2} \tag{21}$$

$DL^{S1} > DL^{S2}$ expresses that the first system may better tolerate the failures than the second one.

Definition 17: (**Cost of a system**) A component is associated with a value corresponding to its cost. For a system S, its cost is the sum of individual costs of its q components.

$$\text{Cost}_S = \sum_{i=1}^{q} \text{Cost}_{\text{component}_i} \tag{22}$$

Definition 18: (**Characteristics of a system**) A system S is entirely characterized by its cost and its dependability level. These characteristics are denoted by C_S.

$$C_S = \left\{ Cost_S, DL^S \right\} \tag{23}$$

Thanks to their characteristics, two equivalent systems can be compared.

Definition 19: (**Comparison of systems**) The system S_1 is said better than S_2, which corresponds to $C_{S1} > C_{S2}$, if

$$\begin{cases} \text{Cost}_{S1} = \text{Cost}_{S2} \\ \text{DL}^{S1} > \text{DL}^{S2} \end{cases} \quad \text{or} \begin{cases} \text{Cost}_{S1} < \text{Cost}_{S2} \\ \text{DL}^{S1} \geq \text{DL}^{S2} \end{cases} \tag{24}$$

Definition 20: (**Optimal systems**) For a set Ω of equivalent systems, the set of optimal systems Ω_{optimal} is defined by

$$\Omega_{optimal} = \left\{ S \in \Omega, \text{ such that } \bar{\exists} S_i \in \Omega \text{ with } C_{Si} > C_S \right\} \tag{25}$$

In the two following subsections, the design methodology of control systems is explained. This methodology uses ordered sequences of failures to evaluate the cost and dependability level of equipment architectures. It also identifies the dreaded events of a control system, lists all the equipment architectures and finally presents the obtained set of optimal operational architectures.

B. Modeling phase

The first phase of the design procedure consists in representing all potential, realizable component architectures. This model is built using two non independent activities: the first one describes the system based on a functional decomposition while the second consists in adding failure modes and failure relationships. The functional decomposition can be modified and improved when failure modes are defined. Indeed, the possibilities of redundancies or fault detections may be easily identified knowing the failure modes.

1) Hierarchical model

In our design methodology, the model of the system is based on a functional hierarchical analysis represented by a tree. This analysis is often used to model control systems [19]–[29].

Three types of nodes are used: the associative node, the alternative node and the elementary node.

- The first type of node corresponds to an associative relationship. It expresses that a complex function requires, to be realized, a set of necessary sub-functions.

  For instance in Fig. 3a, a basic control loop function needs (or is composed by) a measurement function, a control function and an acting function.
- The second type is an alternative node. It is used to propose various possibilities of performing a function.

  For instance in Fig. 3b, for a measurement function, the designer can propose to use a single sensor, an analytical estimation function or a set of redundant sensors. Thanks to this node, different solutions can be envisaged.
- The last type of node is an elementary node. It corresponds to basic function that can be associated with a single component. This type of nodes forms the leaves of the hierarchical decomposition.

2) Improved multi-fault tree

The functional hierarchical decomposition gives the skeleton of the multi-fault tree. With the aim of determining its behavior when a failure occurs, it has to be completed with a description of possible faults and their effect.

This phase consists in associating to each node the set of failure modes that affect the accomplishment of the corresponding function. For complex functions, the relationships between their failures and those of their sub-functions have to be added. The operators AND, OR, PAND and SEQ, that were described in preceding section, will be used for that purpose.

For example in Fig. 4, relationship between failures may be:

(Failure 1A = Failure 2A PAND Failure 3A) AND (Failure 1B = Failure 2B AND Failure 3B).

For the last elementary nodes, the RRC associated to each failure mode allows to define if the component, which is proposed to realize the function, is a standard, a robust or a safe one.

For the alternative nodes, the set of failures is not necessarily the same for all proposed alternatives.

In fact, some failure modes can not appear for some alternatives due to the considered technology. In order to handle this special case, a very large value will be affected to the RRC associated with failures that do not appear.

The obtained improved multi-fault tree describes the various different technological realizations of the system and characterizes the faulty behaviors of the system thanks to the indirect relationships between failure component at the bottom of the decomposition and failure of the mission at the top.

C. *Optimization phase*

The aim of the optimization phase is to determine the best control architecture systems among all the potential architectures described by the improved multi-fault tree. A bottom-up approach is proposed that is comparable to a *branch and bound* method [30].

The general principle is to determine the optimal set of solution for each node of the hierarchical model. For the last elementary nodes, the set of solutions is composed only with a single solution characterized by the cost of the associated component and by the RRC attributed to each of its failure modes. For the upper levels composed with associative nodes and elementary nodes, the set of solutions is built according to the set of its lower functions. For an alternative node, this set is determined from the union of various solutions proposed by each possibility of realization. Thanks to the union of different sets of solutions and thanks to the operator of comparison (cf. definition 21), the optimal set is easily built. For the associative nodes, the possible solutions of realization are deduced one after one by scrutinizing all combinations of solutions proposed by each required function. For a particular combination, the cost of the resulting solution is given by the sum of the costs of the solution retained for each required function, while its dependability level is established from the dependability parameters that are evaluated thanks to the use of the relations corresponding to the operator associated with each failure modes. The optimal set is obtained by the union of all found solutions and according to the comparison criterion.

Whatever the considered node (associative or alternative), since potential solutions are found one after one, the new one can be immediately compared to those previously found and the optimal set can be built progressively. More precisely, a new potential solution is added to the optimal set only if no better solution exists in this set. In the same way, when a new solution is added, if other solutions are worse than the new one, they are removed from this set. At the end, this methodology provides the accurate optimal set as plotted in Fig. 5.

For very huge systems, this optimization method may not be applied due to the combinatorial explosion and other methods, as genetic algorithms [31], may be used. This is the purpose of our future work.

IV. CASE STUDY: DESIGN OF A CONTROL SYSTEM FOR A RAILROAD WAGON

The design methodology is applied below to a railroad system. First, problems associated with a railroad system and the needs to design control systems for a railroad wagon are presented. Then, the methodology to design safe control systems for this wagon is applied. Finally, some results and an example of obtained operational architecture

for a fire protection system are presented.

A. Presentations of the railroad system and the case study

Railroad system is a general term of rail freight, indicating all the systems that carry trucks. There are many techniques and projects around the world: Modalhor and Eurotunnel Fret (France), RoadRailer (the USA), Expressway (Canada), Route Roulante (Switzerland), Sail project (Germany)…

This kind of freight transportation system has many advantages like road congestion reduction, lower pollution and consumption. However, it presents significant risks for itself (derailment, fire…) and for its environment due to the transported goods that is to say the trucks. As a consequence, in comparison with classical trains, a railroad system must present additional dependability requirements and other needs like fire protection, protection against external aggressions and load monitoring. The design of a smart wagon answers these needs [9]-[10]. This wagon will have additional features that increase the dependability level of the global railroad system.

The methodology is illustrated to design one of the new features for a smart wagon: an Automatic Fire Protection System (AFPS). The design of AFPS has been the subject of intensive investigation [32]–[33]–[34]–[35]. Our objective is to obtain a control architecture for such AFPS with a good compromise between cost and dependability level.

B. Modeling phase

1) Hierarchical model

Fig. 6 shows the hierarchical functional model of the AFPS. This system has three missions: to detect fires, to notify system's operators (train driver for example) and to extinguish fires by the use of a fire fighting system. These missions are achieved by one or two control systems. The fire detection part of this system is composed of smoke detectors and heat sensors. The notification part consists in a data processing system composed of a Programmable Logic Controller (PLC) and the power supply. The fire fighting system is activated by relays.

Basic components are described in Table I. Several types of components (standard, safe and smart) with different financial costs may be chosen. These types of components correspond to different robustness level (cf. definition 4). Moreover, four types of redundancies may be chosen: active, passive, serial and parallel redundancies. Serial and parallel redundancies refer to their corresponding component organizations. For an active redundancy, components fulfill their function at the same time and for a passive redundancy, the second component is used when the first one

has failed.

The general structure of the control AFPS is shown on Fig. 7. A PLC system produces an alarm by using data coming from the detection system and also starts a fire fighting system using relays.

2) Construction of the improved multi-fault tree

For sake of conciseness, only the two missions detect fires and notify system's operators are considered here. The first step in constructing the improved multi-fault tree is to define the dreaded events. Two dreaded events are considered:

- *No fire alarm* is activated by the PLC system when a fire is present. This dreaded event is associated with the system's safety level.
- *False alarm* i.e. the control system activates a fire alarm in the absence of fire. This dreaded event is associated with system's availability level.

The second step of construction is to associate to each node of the hierarchical tree the set of failure modes that affect the accomplishment of the corresponding function.

For example in Fig. 6, the detection of fire can be accomplished in 3 ways (alternative node). They consist in detecting either smoke and heat simultaneously, successively or only one of them. Note that with the classical fault tree, the second way, that is to say the case "successively", can not be considered

In order to explain how the results are obtained, let us consider the fire detection that uses a smoke detection followed by heat detection. Failure relationships represented in Table II and by node B2 in Table III, are explained below:

- No fire alarm from the system during a fire, if:
  - No alarm from smoke detection function (the function is continuously inactive).

  AND

  - No alarm from heat detection function (the function is continuously inactive).
- False alarm from the system, if:
  - False alarm from smoke detection function (the function is continuously active).

  PAND

  - False alarm from heat detection function (the function is continuously active).

Then, for each node of the rest of the hierarchical architecture, failure relationships are associated with the same principle for all functions until basic components are reached. Multi-fault tree for both dreaded events is detailed on Table III. This tree characterizes the behavior of the AFPS when a failure occurs thanks to the indirect relationships between failure component at the bottom of the decomposition and failure of the missions at the top.

C. Optimization phase

1) Set of optimal operational architectures

Extensive evaluation of multi-fault tree and our optimization method leads to 74 optimal control architectures of the AFPS. Table IV synthesizes these solutions with respect to the two dreaded events. The number of optimal systems is given along with minimum cost and maximum cost of these solutions and for each optimal system, the methodology provides basic components and their organization (type of redundancy, number and type of components…).

Table V shows some systems from Table IV whose $\mathrm{RRC}_{\min}$ is equal to 3 for both dreaded events. Note from this table that if components are added, system's cost increases and $\mathrm{N}_{\min}$ parameter decreases for both dreaded events but until a precise level. For example, solution with cost of 31 units shows that $\mathrm{N}_{\min}$ parameter increases for dreaded event false alarm. It is due to the great number of components in this architecture. Indeed, the more components are in the architecture, the more important is the probability of a component to be faulty. So, it is important to choose a good architecture with a good balance between cost and dependability level.

2) Comparison with traditional fault trees

In this section, the proposed approach is compared with a traditional dependability method that uses a classical fault tree. Let us recall that in classical fault trees, only **AND** and **OR** operators are used and temporal aspects are not considered. Extensive evaluation of classical fault tree and optimization leads to 117 optimal control architectures. Table VI synthesizes these solutions. The set of solutions for the multi-fault tree is smaller than the set of solutions for the classical tree and the solutions proposed with the multi-fault are better from a dependability point of view, due to a lower number of scenarios for each dreaded event ($N_{\min}$). These results are explained because multi-fault tree integrates temporal functions, which is not the case in the classical fault tree.

A particular optimal solution obtained with the proposed approach is shown on Fig. 8. It costs 31 units and its

RRC_{min} is of 3 for both dreaded events. It uses two control systems in passive redundancy. The first control system uses 1 standard heat detector, 1 standard smoke detector, 1 standard PLC and 2 standard PLCs with alarm priority, 2 standard power supplies in passive redundancy. The second control system uses 1 standard smoke detector, 1 standard heat detector, 2 standard PLCs without alarm priority and 1 standard power supply. Both detection systems are used to detect successively smoke and heat.

REFERENCES

[1] J.C Laprie « Guide de la sûreté de fonctionnement », Cépaduès Ed., Toulouse, 1995.

[2] A. Villemeur "Reliability, Maintainability and Safety Assessment" Methods and techniques, Vol. n°1, ISBN-13: 978-0471930488, Wiley, 1st edition, February 4, 1992;

[3] B. Conrard and M. Bayart "Design of Dependable Control System thanks to a semi-quantitative Optimisation", Proceedings of Safety and Reliability for Managing Risk (ESREL 06), pages 1583-1589, Estoril, 18-22 September 2006.

[4] S. Swaminathan and C. Smidts "The mathematical formulation for the event sequence diagram framework" Reliability Engineering and System Safety 65, Elsevier Ed., pages 103-118, 1999.

[5] C. Kerhen and C. Seguin « Evaluation qualitative de systèmes physiques pour la sûreté de fonctionnement » Formalisation des activités concurrentes (FAC03), Toulouse, France, 2003.

[6] G. Moncelet "Dependability evaluation of mecatronic automative systems using Petri Nets" PhD Thesis in French, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 1998.

[7] R. Schoenig, J.F. Aubry, T. Cambois and T. Hutinet "An aggregation method of Markov graphs for the reliability analysis of hybrid system" Reliability Engineering and System Safety 91, Elsevier Ed., pages 137-148, 2006.

[8] M. Bouissou and J.L. Bon "A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes" Reliability Engineering and System Safety 82, Elsevier Ed., pages 149-163, 2003

[9] J. Clarhaut, S. Hayat, B. Conrard and V. Cocquempot "Safety intelligent system conception for piggyback service" IEEE ICIT Industrial Conference on Industrial Technology, Volume 6, pages 1659-1664, ISBN 1-4244-0726-5, 2006.

[10] J. Clarhaut, S. Hayat, B. Conrard and V. Cocquempot “Safety system conception by using a semi-quantitative reliability evaluation application. Application to railroad transportation systems” in Proceedings of international conference on Industrial Engineering and Systems Management (IESM 2007), May 30 – June 2, pages 330-331, ISBN 978-7-302-15312-2, Tsinghua University Press, Beijing, China, 2007.

[11] F. Simonot Lion, J.P. Thomesse, M. Bayart, M. Staroswiecki “Dependable distributed computer control systems: analysis of the design step activities” in Sharaoui AEK, editor 13th IFAC Workshop on Distributed Control Systems, Toulouse, France, pages 119-124, 1995.

[12] L. Cauffriez, J. Ciccotelli, B. Conrard and M. Bayart “Design of intelligent distributed control systems: a dependability point of view” Reliability Engineering and System Safety 84, Elsevier Ed., pages 19-32, 2004.

[13] L. Cauffriez, V. Benard and D. Renaux “A new formalism for designing and specifying RAMS parameters for complex distributed control systems: the SAFE-SADT formalism”, IEEE Transactions on Reliability, Volume: 55, Issue: 3, pages 397-410, 2006.

[14] A. Villemeur “Dependability of industrial systems”. Book in French, Eyrolles, Paris, ISSN 0339-4198, 1988.

[15] G. Zwingelstein « Sûreté de fonctionnement des systèmes industriels complexes », Techniques de l'ingénieur N° S8250, 1999.

[16] AFNOR French Standards NF X60-500. « Terminologie relative à la Fiabilité-Maintenabilité-Disponibilité » 1988.

[17] AFNOR French Standards NF EN 292. « Sécurité des machines Notions fondamentales, Principes généraux de conception, partie 1: terminologie de base, méthodologie » 1991.

[18] B. Conrard, V. Cocquempot and M. Bayart “Design of Automation Systems with criterion of cost and dependability”, Qualita 2007 congress, Tanger, Maroco, 20-22 March 2007.

[19] B. Conrard and M. Bayart, “Design of Dependable Control System thanks to a semi-quantitative Optimisation”, ESREL 06 Safety and Reliability for Managing Risk Conference, Estoril, 18-22 september 2006.

[20] C. Sourisse and L. Boudillon « La sécurité des machines automatisées », Groupe Schneider France, 1997

[21] H. Kumamoto and E.J. Henley “Probabilistic risk assessment and management for engineers and scientists” New York, IEEE Press, ISBN 0-780-31004-7, 1996.

[22] M. Rausand and A. Hoyland "System Reliability Theory: Models, Statistical Methods and Applications" Second Edition, Wiley Ed., Pages 99-103, 2004.

[23] IEC 60300-3-1 "Dependability management. Part 3-1: Application guide, Analysis techniques for dependability – guide on methodology. Geneva, Switzerland, IEC, International Electrotechnical Commission, ISBN 2-8318-6791-6, 2003.

[24] J.B. Dugan "Fault tree Analysis of Computer-Based Systems" Annual Reliability and Maintainability Symposium, January 2001.

[25] M. Bouissou and Y. Dutuit "Reliability analysis of a dynamic phased mission system", MMR2004 congress, Santa Fe, June 2004.

[26] M. Cepin and B. Mavko "A dynamic fault tree" Reliability Engineering and System Safety 75, Elsevier Ed., pages 83-91, 2002.

[27] J.B. Dugan, S.J. Bavuso, M.A. Boyd "Dynamic fault-tree models for fault-tolerant computer systems" IEEE Transactions on Reliability, pages 363-377, Volume: 41, Issue:3, 1992.

[28] D. Coppit, K.J. Sullivan, J.B. Dugan "Formal Semantics for Computational Engineering: A Case Study On Dynamic Fault Trees", ISSRE, page 270, 11th International Symposium on Software Reliability Engineering (ISSRE'00), 2000.

[29] V. Benard, L. Cauffriez and D. Renaux "The Safe-SADT method for aiding designers to choose and improve dependable architectures for complex automated systems" Reliability Engineering and System Safety, Elsevier Ed., In Press, 2005.

[30] A. D'Ariano, D. Pacciarelli and M. Pranzo "A branch and bound algorithm for scheduling trains in a railway network" European Journal of Operational Research, Volume 183, Issue 2, Pages 643-657 Science Direct Ed, 2006.

[31] M. Mitchell "An Introduction to Genetic Algorithms", MIT Press, Cambridge, MA, 1996.

[32] H. H. Amer and R.M. Daoud "Fault Secure multi-detectors Fire protection System for trains" IEEE IMTC – Instrumentation and Measurement Technology Conference, Ottawa, Canada, 17-19 May 2005.

[33] H. M. Elsayed, H.H. Amer, and R.M. Daoud "Fire protection System for Cargo Trains using fuzzy logic" IEEE Workshop on Soft Computing in industrial Applications, Helsinki University of Technology, Espoo, Finland, June 28-30 2005.

[34] S. Eisinger and U.K. Rakowsky "Modeling of uncertainties in reliability centered maintenance – a probabilistic approach", Reliability Engineering and System Safety 71, Elsevier Ed., pages 159-164, 2001.

[35] G. Jiang, F. Shang and F. Wang "A combined Intelligent Fire Detector with BP Networks" IEEE Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, June 21-23, 2006.