Dependability and Safety assessment of complex systems and design optimisation

André Cabarbaye

CAB INNOVATION

3, rue de la Coquille
31500 Toulouse
Tel. 05 61 54 68 08
Andre.Cabarbaye@cabinnovation.fr

Roland Laulheret

Centre National d'Etudes Spatiales (CNES)

18 avenue Edouard Belin
31401 Toulouse Cedex 4
Tel. 05 61 27 47 19
Roland.laulheret@cnes.fr

Abstract:

After a brief presentation of the contribution of design optimisation in regard of availability and safety, this paper describes an original method to assess and optimise complex systems, which are composed of interdependent units. This hybrid method is realised by coupling of different techniques (Fault-tree, markovian processing, Monte-Carlo simulation) and is applied to solve any problems from spatial area (satellites, control centres…). Using tools of CAB INNOVATION Company, the main part of modelling is automatically realised by software and is accessible to non-specialist designers. So, the processing assessment is enough fast to be directly coupled with optimisation techniques as Genetics Algorithms or Nelder Mead algorithm in regard of criteria defined by the user (maximum of availability for the best cost).

Key words: Availability, modelling, optimisation, Markov, Fault-tree, Monte-Carlo simulation.

Approximated word count: 3149

Submission category: Modelling, assessment and optimisation

Introduction

Optimising system architecture, implementation, maintenance and operation, as regards the availability of the service provided for the user, frequently covers major opportunities. According to the problems encountered, the objective may consist in minimising possession and maintenance costs while complying with an availability goal, maximising availability for a given cost target, or complying with a criterion laid down using these results.

This optimisation is the result of multiple trade-off covering the following in particular:

- distribution of system functions in various elements,
- reliability of elements which depends on the technologies implemented and the quality of the components used,
- rates of use,
- redundancies (if any) and their characteristics,
- functioning degraded mode (if any)
- reconfiguration duration

and for repairable elements:

- repair time on site or via a factory return,
- spare part kits on site (if any),

etc.

System availability may be evaluated by miscellaneous modelling techniques (Reliability Block Diagram, Markov graph, Petri net…) and processing (analytical computation, markovian processing, Monte-Carlo simulation…) methods.

In an industrial context, the duration of this evaluation, which covers modelling by analyst and computer processing, must be relatively short, and the results exact. Indeed, the assessment is often made during concurrent design engineering or call for tenders answer period. So the company are more and more often directly concerned by the good functioning of their products in operation by contractual incentive or penalty. In fact, the evaluation is now part of an optimisation process and no more a mere verification.

1 - Problematical:

In such context, CNES has two kinds of assessment and optimisation problems to solve:

- The satellites are non-repairable systems but can be reconfigured by using on board redundancies or degraded modes.
- The on ground systems (antenna, control centres, telecommunication networks...) which can be repaired or reconfigured.

Independently of their sizing (about one hundred different units), these two kinds of system can be considered as “complex”, because they are composed of interdependent units. By example, the failure rate of one unit in cold redundancy (λ or λ_{OFF}) depends of the states of nominal units (in failure or not). In the same way, maintenance characteristics of one unit (reparation duration….) could be affected by the states of the other ones in case of common resources using (spares, repair officers...).

2 – Methodological aspect:

Two main criteria have to be considered in this context:

- It shall be possible for designers to validate the model, which must also, if possible, be accessible to non-specialists,
- processing time shall be compatible with the search for an optimum design

This search could be conducted by means of sensitivity analyses conducted on each parameter individually. But theses methods quickly reach their limits, in particular when there are numerous variables and multiple optima, and it then becomes necessary to call upon more sophisticated optimisation techniques from the fields of operational research and artificial intelligence

In this context, the duration of Monte-Carlo simulation is a very hard constraint.

Markov processes have significant advantages, in terms of computation time and accuracy, over the Monte-Carlo simulation techniques, but are limited by the possible number of states of the system. However models in Markov graph form are relatively complex and can only really be implemented by specialists.

Models in fault-tree form are very simple and corresponding analytical processing is fast. But this modelling method is static and can’t take into account the complexity (stochastic dependencies).

Nevertheless, a hybrid method realised by coupling of different tools has allowed solving the main part of the CNES problems. This method has been implemented with tools of CAB INNOVATION company (SUPERCAB: markovian processing, CABTREE: Fault-tree, GENCAB: optimisation and SIMCAB: simulation).

3 – Used method

The main principles of this method are:

- coupling between fault tree and Markovian-processing functions,
- using of generic M among N redundancy models and automatic Markovian modelling tools,
- coupling between assessment and optimisation tools (hybrid method using Genetic Algorithms and non-linear Simplex),

- coupling between assessment and Monte-Carlo simulation tools to control dispersions in computations.

3.1 - Processing of the complexity

Taking into account of the complexity in all kinds of design is today impossible. Any dependence relations, linking certain transition rate values to system conditions (conditional maintenance, cold redundancy...) can't be taken into account in fault-trees. Such dependence relations can be modelled by Markov graph but only for small system due to the limitation of number of states (2^n states when n elements make up the system). This complexity can be taken into account by Monte-Carlo simulation but the processing duration is long and the results not very precise. However, the majority of systems are not entirely complex but only partially. In fact, such systems are composed of complex or simple subsystems. Coupling between a fault-tree tool (CABTREE), to describe the functioning of a system from subsystem conditions, with a markovian tool (SUPERCAB), to modelise the subsystems functioning, allows solving the main part of CNES problems. This coupling possibility is illustrated by the example of figure 1.

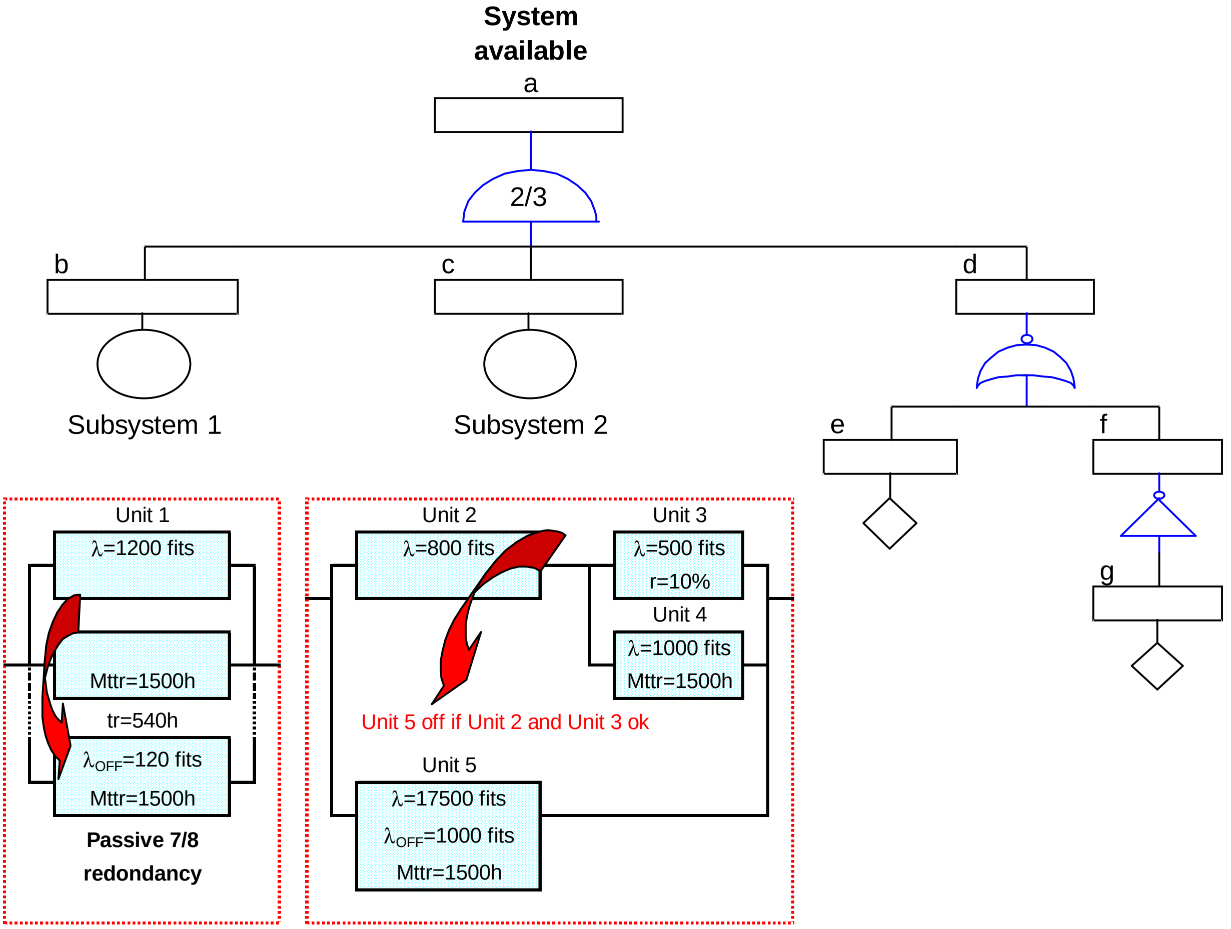


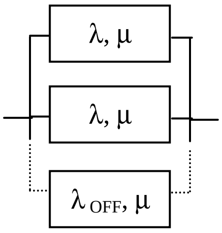
Figure 1. Coupling between fault-tree and markovian processing tools

3.2 - Automatic markovian modelling

The Markov graph modelling is long, hard and difficult. The risk of mistake is important especially if a non-specialised designer makes the modelling. This is why SUPERCAB tool includes generic M among N redundancy models and an automatic Markovian modelling tools.

3.2.1 - Generic M among N redundancy models

From a parametrical function, the tool built a markovian matrix and then assesses the availability or reliability in transient or steady state, as shown in figure 2.



REDONDANCY (M; N; $\lambda;\lambda_{OFF}$; μ; T)

	1	2	3	...	N-M+1	N-M+2
No failure : 1	-	$M\lambda + (N-M)\lambda_{OFF}$				
Loss of 1 unit : 2	μ	-	$M\lambda + (N-M-1)\lambda_{OFF}$			
Loss of 2 unit : 3		μ	-			
...						
Loss of N-M unit : N-M+1					-	$M\lambda$
System loss : N-M+2					μ	-

Figure 2. M among N reparable cold redundancy (one single repair officer)

A more generic model has been realised to assess the whole redundancy types used by CNES, for repairable systems (on ground) or only reconfigurable systems (on satellites). This model can take into account ON and OFF failure rates (λ and λ_{OFF}), probability of failure before using up (γ), average reconfiguration duration in which the mission is stopped (MDT_s) and average non-operational duration due to failure (MDT_l). Both durations can be modelled by exponential (rates μ_i) or Erlang law (with k fictitious states). Several politics of maintenance can be considered with 1 or n repair officers or with duration of reparation independent of the number of units in failure.

REDONDANCY (M; N; $\lambda;\lambda_{OFF}$; γ; T; MDT_l; kl; Type_l; MDT_s; ks; Type_s; R)

This parametrical function assesses the availability (at t or ∝) or the reliability, considering the loss of system as an absorbing state.

In case of the mission is not stopped during reconfiguration duration, the markovian matrix with dimension N-M+2 as shown in figure 3 is used.

	1	2	3	4	...	N-M+1	N-M+2
No failure : 1	-	$M\lambda(1-\gamma) + (N-M)\ \lambda*$	$M\lambda\gamma\ (1-\gamma)$	$M\lambda\gamma^2\ (1-\gamma)$		$M\lambda\gamma^{N-M-1}\ (1-\gamma)$	$M\lambda\gamma^{N-M}$
Loss of 1 unit : 2	μl_1	-	$M\lambda(1-\gamma) + (N-M-1)\lambda*$	$M\lambda\gamma\ (1-\gamma)$		$M\lambda\gamma^{N-M-2}\ (1-\gamma)$	$M\lambda\gamma^{N-M-1}$
Loss of 2 unit : 3	$\mu l'$	μl_2	-	$M\lambda(1-\gamma) + (N-M-2)\lambda*$		$M\lambda\gamma^{N-M-3}\ (1-\gamma)$	$M\lambda\gamma^{N-M-2}$
Loss of 3 unit : 4	$\mu l'$		μl_3	-		$M\lambda\gamma^{N-M-4}\ (1-\gamma)$	$M\lambda\gamma^{N-M-3}$
...							
Loss of N-M unit : N-M+1	$\mu l'$					-	$M\lambda$
System loss: N-M+2	$\mu l''$					μl_{N-M+1}	-

Figure 3. M among N redundancy (system available during reconfigurations)

In case of the mission is stopped during reconfiguration duration, the markovian matrix shown in figure 4,with dimension 2(N-M)+2, is used.

	1	2	3	4	5	6	...	2(N-M)+2
No failure : 1	-	$M\lambda$	$(N-M)\ \lambda*$					
Reconfiguration : 2	$\mu 2$	-	$tr(1-\gamma)$	$tr\gamma +(M-1)\ \lambda +(N-M)\ \lambda*$				
Loss of 1 unit : 3	$\mu 1$		-	$M\lambda$	$(N-M-1)\ \lambda*$			
Reconfiguration : 4			$\mu 2$	-	$tr(1-\gamma)$	$tr\gamma +(M-1)\ \lambda +(N-M-1)\ \lambda*$		
Loss of 2 unit : 5			$\mu 1$		-	$M\lambda$		
Reconfiguration : 6						-		
...								
Reconfiguration : 2(N-M)								$tr\gamma +(M-1)\ \lambda + \lambda*$
Loss of N-M unit : 2(N-M)+1								$M\lambda$
System loss : 2(N-M)+2								-

Type_l = 1 : 1 repair officer ($\mu l_i = \mu l$, $\mu l' = \mu l'' = 0$)

Type_l = 2 : n repair officers ($\mu l_i = i * \mu l$, $\mu l' = \mu l'' = 0$)

Type_l = 3: duration of reparation independent of the number of units in failure ($\mu l_i = 0$, $\mu l' = \mu\, l$, $\mu l'' = \mu l$ if R = False)

Type_s = true: repairing starts at the beginning of reconfiguration duration

R = true (Reliability) $\Rightarrow \mu l_{N-M+1}$ and $\mu l'' = 0$.

Figure 4. M among N redundancy (system unavailable during reconfigurations)

An improvement of this last model allows calculating the reliability or the availability of a set of units in redundancy M among N with a stock S of spares as shown in figure 5.

The system is available during the N-M-S first reconfiguration duration but unavailable during the following ones.

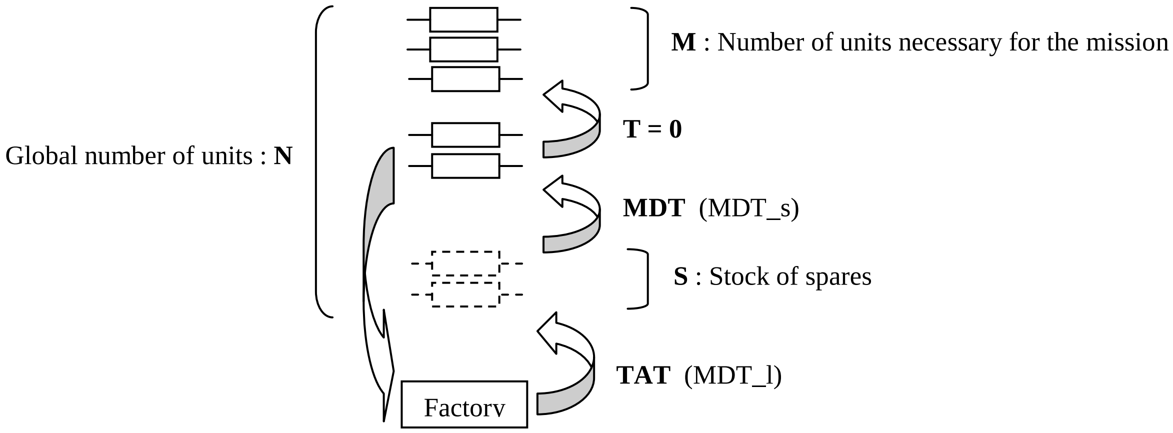
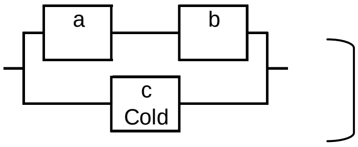

M : Number of units necessary for the mission
T = 0
Global number of units : N
MDT (MDT_s)
S : Stock of spares
TAT (MDT_l)
Factorv

Figure 5. M among N with a stock S of spares

3.2.2 - Automatic Markovian modelling tool

To easily assess different designs, the tool generates automatically Markov matrix of a system, from logic expressions featuring its behaviour. Its principle is described in figure 6.



Available states: **(a × b) + c**

Dependence relation: **a × b ⇒ λc***

(λc* = λcoff)

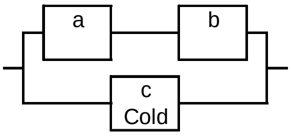
	MAT :	1	2	3	4	5	6	7	8
c b a	1	-	λa	λb		λc*			
c b na	2	μa	-		λb		λc		
c nb a	3	μb		-	λa			λc	
c nb na	4		μb	μa	-				λc
nc b a	5	μc				-	λa	λb	
nc b na	6		μc			μa	-		λb
nc nb a	7			μc		μb		-	λa
nc nb na	8				μc		μb	μa	-

System unavailable in the states 6, 7 and 8

Figure 6 – Principe of automatic markovian modelling

States in which system is available are defined by a logic expression using operators OR (+), AND (×), NOT (n). Any possible dependence relations, linking certain transition rate values to system conditions (conditional maintenance, cold redundancy...), may be similarly expressed. In the example of figure 6, unit c is off if a and b are running.

If system units are not all considered as individually repairable, equivalent states may be regrouped by the program for optimising matrix dimension. The last example can be optimised as shown in figure 7.



(a × b) + c   a × b ⇒ λc*

MAT :		1	2	3	4
c b a	1	-	λc*	λa+λb	
nc b a	2		-		λa+λb
c nb na	3			-	λc
nc nb na	4				-

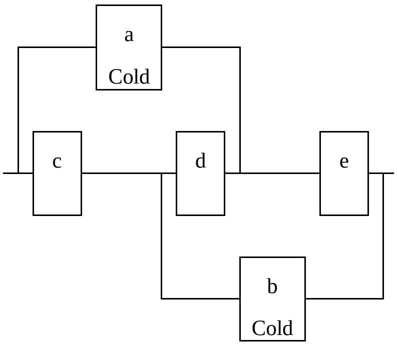
System unavailable in the state 4

Figure 7 – Automatic markovian modelling with optimisation

So as to identify the regrouped states, to each of them is given the name of group states comprising most of failing units (c nb na = c b na + c nb a + c nb na et nc nb na = nc b na + nc nb a + nc nb na).

Repair rates relating to blocks may be subsequently introduced in reduced matrix (example: repair rates of set ab).

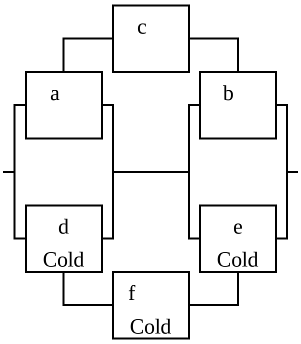
The dimension of the build matrix is variable according to the logic expressions used. However this equivalent states grouping method is efficient particularly for not repairable systems as satellites. The two examples of figure 15 leads to matrix of dimension 8 instead of 32 (2^5) and 10 instead of 64 (2^6).



a*e + c*d*e + c*b

MAT :	1	2	3	4	5	6	7	8
e d c b a : 1	-	λa*	λb*	λd		λc	λe	
e d c b na : 2		-			λb*		λd+λe	λc
e d c nb a : 3			-		λa*	λc+λd		λe
e nd c b a : 4				-		λb*+λc	λa+λe	
e d c nb na : 5					-			λc+λd+λe
e nd nc nb a : 6						-		λa+λe
ne nd c b na : 7							-	λb+λc
ne nd nc nb na : 8								-

System unavailable in the state 8



(a*c + d*f) * (b*c + e*f)

Grouped states	1	2	3	4	5	6	7	8	9	10
a b c d e f	-	λa	λb	λd*	λe*			λc	λf*	
na b c d e f		-				λe		λb*+λc*		λd+λf
a nb c d e f			-				λd		λa*+λc*	λe+λf
a b c nd e f				-			λb		λe*+λf*	λa+λc
a b c d ne f					-	λa			λd*+λf*	λb+λc
na b c d ne f						-				λb+λc+λd+λf
a nb c nd e f							-			λa+λc+λe+λf
na nb nc d e f								-		λd+λe+λf
a b c nd ne nf									-	λa+λb+λc
na nb nc nd ne nf										-

System unavailable in the state 10

Figure 8 – Examples with equivalent states regrouping

3.3 - The optimisation

Coupling between assessment and optimisation tools allows automating the search of the most efficient architecture in accordance with criteria. This criteria links by example availability (or reliability) and possession and maintenance costs. GENCAB tool allows such coupling. It is based on an especially efficient technique aiming at seeking the overall optimum (hybrid method using Genetic Algorithms and non-linear Simplex, also named Nelder Mead algorithm). Its general principle is described in figure 9.

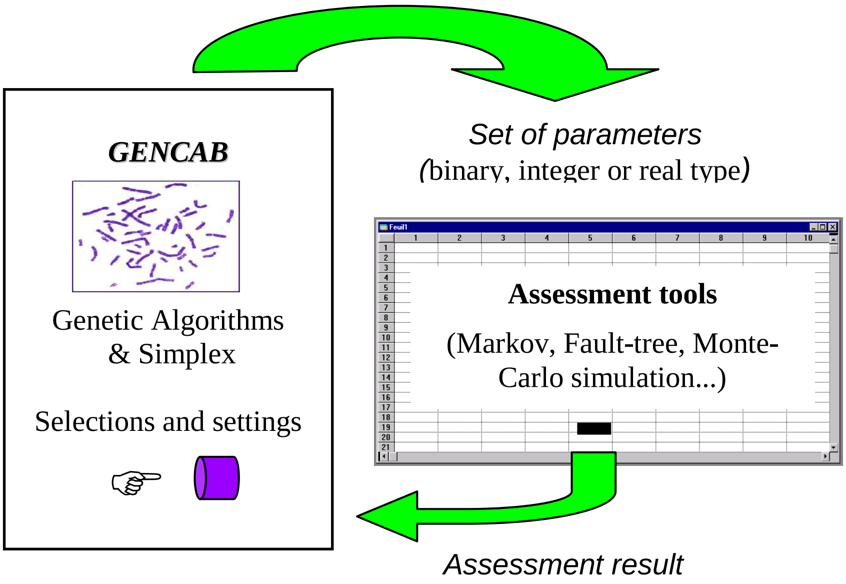
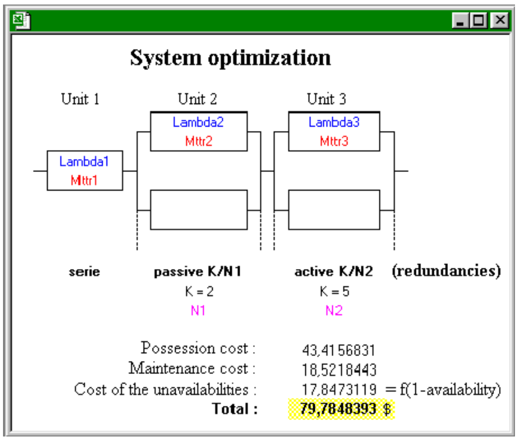

GENCAB
Genetic Algorithms & Simplex
Selections and settings
Set of parameters
(binary, integer or real type)
Assessment tools
(Markov, Fault-tree, Monte-Carlo simulation...)
Assessment result

Figure 8 – General principle of *GENCAB* generic optimising tool

The tool seeks the optimal parameter configuration (binary, integer or real type) which maximises or minimises function's result, without stopping at the first local optimum found. Example of Figure 9 shows this coupling possibility to optimise globally possession, running and maintenance costs. SUPERCAB tool assesses the system availability, which is transformed in cost of service unavailability.



System optimization

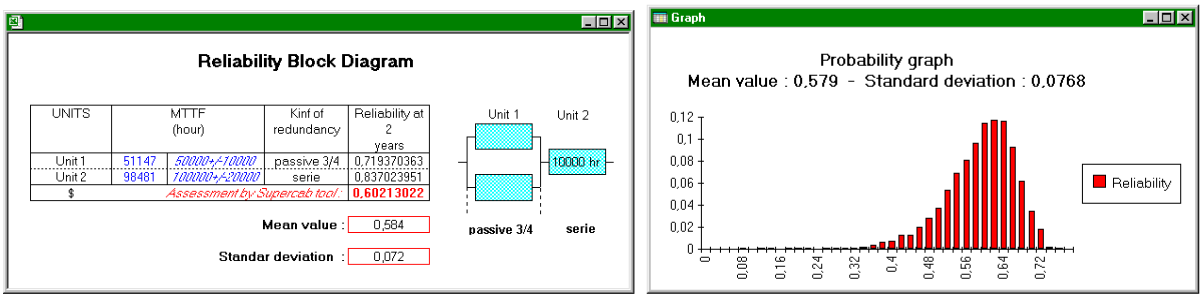
Possession cost : 43,4156831
Maintenance cost : 18,5218443
Cost of the unavailabilities : 17,8473119 = f(1-availability)
Total : 79,7848393 $

Figure 9 – Example of system optimisation (6 real parameters and 2 integers)

3.3 - Dispersions analysis

Direct coupling between Monte-Carlo simulation and optimisation tools is often not possible due to the delay of assessment (the number of optimisation steps is multiplied by the number of simulation steps in a first approximation). On the other hand, coupling between assessment and Monte-Carlo simulation tools allows controlling variations in computations in accordance with input dispersions.

SIMCAB tool allows a such coupling which is illustrated by two simple cases: The example of figure 10 shows the effects of input dispersions (Mean Time To Failure) on markovian processing results.



Reliability Block Diagram

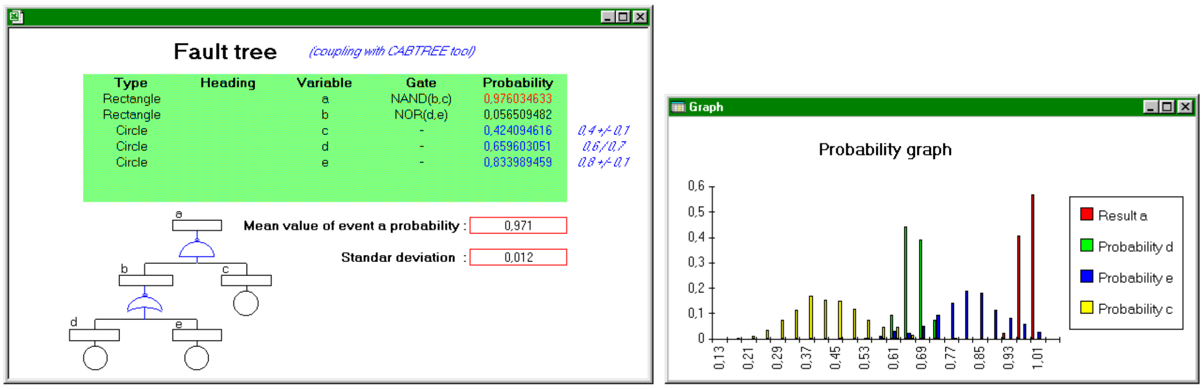
UNITS	MTTF (hour)		Kinf of redundancy	Reliability at 2 years
Unit 1	51147	50000+/-10000	passive 3/4	0,719370363
Unit 2	98481	100000+/-20000	serie	0,837023951
$		Assessment by Supercab tool :		0,60213022

Mean value : 0,584

Standar deviation : 0,072

Figure 10 – Reliability dispersions of a system design

In the same way, the example of figure 11, shows the probability variation of a fault-tree top-event in accordance with the probability dispersions of the elementary events (the tool performs an analytical assessment without approximation).



Fault tree (coupling with CABTREE tool)

Type	Heading	Variable	Gate	Probability	
Rectangle		a	NAND(b,c)	0,976034633	
Rectangle		b	NOR(d,e)	0,056509482	
Circle		c	-	0,424094616	0,4 +/- 0,1
Circle		d	-	0,659603051	0,6 / 0,7
Circle		e	-	0,833989459	0,8 +/- 0,1

Mean value of event a probability : 0,971

Standar deviation : 0,012

Figure 11 – Probability dispersions in fault-tree

Conclusion:

Through this brief presentation of tools to assess complex systems dependability and safety and to optimise design, we wouldn't recommend a generic method but only show any solutions to solve specific problems. Indeed, the methods have to be chosen according to the problem characteristics and not only in accordance with the available tools.

However, moreover the quality of methods and tools, the design optimisation in regard of dependability is often difficult due to the lack of cost parametrical models and valid operational data.

ACKNOLEDGEMENTS AND SHORT REFERENCES

ACKNOLEDGEMENTS

[1] A. Pages & M. Gondran - *Fiabilité des systèmes - Edition Eyrolles, Paris 1980*

[2] A. Villemeur - *Sûreté de Fonctionnement des systèmes industriels - Edition Eyrolles, Paris 1987*

Last publications of the authors

[1] André Cabarbaye - *SUPERCAB PRO : Un atelier d'Ingénierie Système sous Microsoft Excel®* - 2ème Conférence Annuelle d'Ingénierie Système, organisée par l'AFIS, TOULOUSE, 26-28 juin 2001.

[2] André Cabarbaye - *SIMCAB : Un outil générique de Simulation sous Microsoft Excel®* - 3e Conférence Francophone de Modélisation et Simulation (MOSIM'01), Troyes 25 au 27 avril 2001.

[3] André Cabarbaye - *Simulation dynamique des arbres d'événements* - 4e Congrès international pluridisciplinaire Qualité et Sûreté de Fonctionnement (Qualita 2001), Annecy 22 et 23 mars 2001.

[4] Roland Laulheret, André Cabarbaye - *Elaboration d'une politique de Retour d'Expérience en Sûreté de Fonctionnement dans le domaine spatial* - 4e Congrès international pluridisciplinaire Qualité et Sûreté de Fonctionnement (Qualita 2001), Annecy 22 et 23 mars 2001.

[5] A.Cabarbaye, Julien Séroi - *Optimisation dans le domaine de la Sûreté de Fonctionnement* - 12e Colloque National de Sûreté de Fonctionnement (λ/μ 12), Montpellier 28 - 30 mars 2000.

[6] Linda Tomasini, André Cabarbaye, Julien Séroi, Frédéric Garcia, - *Optimisation de la maintenance d'une constellation de satellites* - 12e Colloque National de Sûreté de Fonctionnement (λ/μ 12), Montpellier 28 - 30 mars 2000.

[7] André Cabarbaye, Lamine Ngom - *Simulation dynamique des arbres d'événements* - 12e Colloque National de Sûreté de Fonctionnement (λ/μ 12), Montpellier 28 - 30 mars 2000.

[8] André Cabarbaye – *Outil générique d'optimisation par Algorithmes Génétiques et Simplexe* - 8 èmes Journées Nationales du groupe Mode (Mathématique de l'Optimisation et de la Décision) de la SMAI, Toulouse 23 - 25 mars 2000.

[9] A. Cabarbaye - *Optimisation dans le domaine de la Sûreté de Fonctionnement* - Qualité Espace N°36, décembre 1999.

[10] André Cabarbaye, Julien Séroi, Linda Tomasini - *Optimisation de la Sûreté de Fonctionnement des systèmes spatiaux* - 3e Congrès International de Génie Industriel, Montréal 26 - 28 mai 1999.

[11] A. Cabarbaye, L. Tomasini, L. Ngom, S. Allibe - *Apport des algorithmes génétiques à la Sûreté de Fonctionnement et à l'optimisation des systèmes* - Congrès Qualita 99, Paris 25-26 mars 99.

[12] R. Laulheret, A. Cabarbaye - *Evolution des études qualitatives de Sûreté de Fonctionnement dans le domaine spatial* - Congrès Qualita 99, Paris 25-26 mars 99.

[13] A. Cabarbaye, L. Ngom - *Mise en œuvre de la méthode des états fictifs et génération automatique des matrices de Markov* - Congrès Qualita 99, Paris 25-26 mars 99.

[14] A. Cabarbaye, F. Garcia, L. Tomasini - *Apport de l'apprentissage par renforcement aux problèmes de maintenance optimale : application aux constellations de satellites* - Congrès ROADEF ' 99.

[15] L. Ngom, J.C. Geffroy, C Baron, A. Cabarbaye - *Prise en compte des relations de dépendances dans la simulation de Monte-Carlo des arbres de défaillances non cohérents* - Phoebus N° 9 – 1999

[16] A.Cabarbaye - *Apports, difficultés et prospectives dans le domaine de l'évaluation quantitative en Sûreté de Fonctionnement* - Qualité Espace N°33, 1998.

[17] A. Cabarbaye, *Modélisation et évaluation des systèmes* - Cours de technologies spatiales Edition Cepadues Toulouse 1998.