METFAC: Design and Implementation of a Software Tool for Modeling and Evaluation of Complex Fault-Tolerant Computing Systems

J.A. Carrasco and J. Figueras

Departament d'Enginyeria Electrònica, E.T.S.E.I.B.
Universitat Politécnica de Catalunya, SPAIN

Abstract

The use of discrete-state, continuous-time Markov processes (MP, for short) has been shown useful for the modeling and evaluation of Fault-Tolerant Computing Systems. The efficient application of this approach to complex systems depends largely on the availability of powerful software tools that provide "friendly" utilities to specify and run the models, and that can deal with large size MPs. Both aspects have been considered in the design and implementation of a software tool called METFAC. This paper discusses the framework solutions on which the tool is based and presents some performance metrics. In relation to the framework solutions, the following aspects are considered: generative specification of the models by production rules, and sparse analytical evaluation of the measures.

1 INTRODUCTION

The development of software tools for the modeling and evaluation of Fault-Tolerant Computing Systems has been an active research area during the last years. ARIES 76 (NgAv80) unified previous, heuristically developed, models using MPs as the mathematical framework. A later version—ARIES 81 (Mak82)—included among other enhancements the ability to process user-defined models specified through their transition matrices. Higher level specification methodologies were provided in the SURF (Cos 81) and the tool described in (Bey 81). In the former, a MP for the system is obtained by combining subsystem models and merging the resulting states. In (Bey 81) the use of Petri nets with inhibitor arcs is proposed. An advantage of this approach is that the merged MP is directly obtained if the symmetries of the system are considered in the specification. However, the constraints imposed by the Petri nets to the firing conditions, and, specially to the "new state" functions (constant displacements in the state variable space) may make difficult or non-natural the modeling of some activities. It seems that more powerful specification methodologies are necessary.

Modeling of current complex fault-tolerant systems presents the problem of processing large MPs. Two approaches are available: a) to use approximations in order to reduce the number of states, and b) to use efficient numerical methods. Good specification methodologies should allow the introduction of approximations. In relation to the second approach, the sparseness of the transition matrices should be exploited.

This paper discuss some of the methodological solutions incorporated into METFAC. In the next section the model specification methodology will be described and an example given. Next, the sparse solutions for the numerical problems involved in the evaluation of the measures will be discussed. Finally, the organization of the tool will be briefly described and some performance metrics presented.

2 MODEL SPECIFICATION AND CONSTRUCTION

2.1 Methodology

The proposed specification methodology is based on production rules (BaFe82). At a certain abstraction level the behavior of a fault-tolerant computing system results from the combination of random processes that manifest themselves in specific actions (faults, maintenance operations, etc.) which change the system state. In the proposed methodology each process (or group of homogeneous processes) is described by a production rule, that specifies when the process is active and the consequences that may be derived from the execution of the associated action.

A high level behavioral description called *generative specification* is used to construct the ultimate model called *evaluated transition digraph* (transition digraph with numerical values for transition rates and state indices). The process is illustrated in Fig. 1. The generative specification includes the following items: PE (set of integer variables called *structural parameters*), PF (set of real variables called *functional parameters*, VE (set of integer variables called *state variables*), R (set of production rules r_k), Λ (set of positive real functions $\lambda^k(PE, PF, VE)$ called *action rates*), C (set of positive real functions $c_i^k(PE, PF, VE)$ called *response probabilities*) and $I(PE, PF, VE)$ (positive real function called *index*). The structure of the production rules has been tailored to provide naturalness and is shown below.

if $a_k(PE, VE)$ **then**

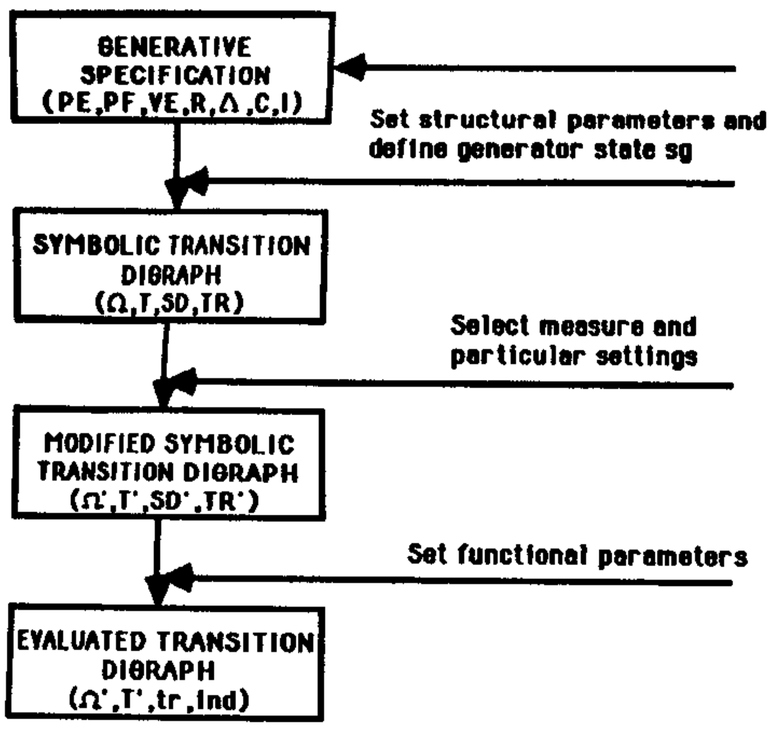
  if $a_1^k(PE, VE)$ **then** $VE \leftarrow s_1^k(PE, VE)$

  ...

  if $a_{qk}^k(PE, VE)$ **then** $VE \leftarrow s_{qk}^k(PE, VE)$

end

0731-3071/86/0000/0424$01.00 © 1986 IEEE



Ω,Ω': state sets
T, T': transition sets
SD, SD': state descriptions
TR, TR': transition labellings
tr: transition rates
ind: state indices

Figure 1: Automatic construction of transition digraphs.

Several responses, with different *response conditions* a_j^k and *new state functions* s_j^k, can be defined. For semantic convenience a global *action activation condition* a_k has been included also. Each production rule r_k has an associated action rate function λ_k, and each response j, $j = 1, \ldots, qk$, a function c_j^k evaluating the probability that the response occurs given that it is active. This multiple-response scheme allows a natural modeling of probabilistic reactions of the system (i.e., limited effectiveness of fault-handling procedures) as well as reconfiguration strategies depending upon the current system configuration. Finally, the index function I associates weighting factors with states for the evaluation of performance and cost-related measures.

The structural parameters include those parameters on which the topology of the transition digraph depends. They are usually structural characteristics of the system (e.g., number of processors). Setting this parameters and defining a *generator state sg* allows the construction of the *symbolic transition digraph*, which is obtained by applying the active production rules in *sg* and the states successively generated. During the process, the values of the state variables and the identifiers of the pairs action/response that were active are recorded in, respectively, the labellings SD and TR. Some measures require the modification of the transition digraph. For instance, to evaluate reliability-type measures, the faulty states have to be merged into an absorbing state and transitions from it have to be suppressed. The result is the *modified symbolic transition digraph*. In the last stage, the user sets values for the functional parameters, and the transition rates and state indices are easily computed from the labellings SD' and TR', and the functions λ_k, c_i^k and I.

The generality of the methodology depends on the syntax allowed for the functions appearing in the generative specification. In METFAC a high-level embedded language has been used. We note that a great specification power is thus obtained. In particular, Petri net-based specifications are obtained as particular cases. Moreover, approximate models can be easily defined. Let us consider, for instance, the method used in (Arl83) to avoid the "combinatorial explosion" of the state space for repairable systems, based on limiting the number of simultaneous faulty subsystems. This approximation can be implemented by introducing state variables counting the number of faulty subsystems of a given class, and structural parameters establishing their maximum values. The condition functions a_k and a_j^k should be modified so that they disable the production rule and the responses when their execution would overflow a counter variable. Once such a generative specification has been created, the size of the digraphs can be adjusted by setting adequately the structural parameters.

2.2 An illustrative example

Let us consider an on-line repairable system with degradation capability, composed of N identical modules. Initially, all modules are active and degradation up to a configuration with $K < N$ modules is allowed. Active modules are subject to transient and permanent faults with rates λ_t and λ_p, respectively. Recovery procedures have negligible durations and their effectiveness is modeled by the following parameters:

- ρ probability that a transient fault is considered permanent and successfully recovered
- u_t probability that a transient fault leads to a catastrophic failure
- u_p probability that a permanent fault is not successfully recovered and leads to a catastrophic failure

Two mutually exclusive maintenance actions are taken by one repairman: reparation of a module in permanent fault (rate μ) and restart from catastrophic failure (rate μ_r). The system state can be represented using three state variables: NA (number of active modules), NPF (number of modules in—or considered in—permanent fault), and CF (1 if catastrophic failure, 0 otherwise). Let us assume that we were interested in evaluating the performance-related measure: "expected number of active modules". An applicable generative specification follows:

Structural parameters $PE = \{N, K\}$

Functional parameters $PF = \{\lambda_p, \lambda_t, \rho, u_p, u_t, \mu, \mu_r\}$

State variables $VE = \{NA, NPF, CF\}$

Production rules $R = \{r_1, r_2, r_3, r_4\}$

r_1: Permanent fault

$a_1 = (NA > 0)$

response 1: There are more than K active modules and the recovery procedures are successful

$a_1^1 = (NA > K)$

s_1^1: $NA \leftarrow NA - 1$, $NPF \leftarrow NPF + 1$

response 2: There are more than K active modules but the recovery procedures fail

$a_2^1 = (NA > K)$

s_2^1: $NA \leftarrow 0$, $NPF \leftarrow NPF + 1$, $CF \leftarrow 1$

response 3: There are K active modules

$a_3^1 = (NA = K)$

s_3^1: $NA \leftarrow 0$, $NPF \leftarrow NPF + 1$

r_2: Transient fault

$a_2 = (NA > 0)$

response 1: The fault leads to a catastrophic failure

$a_1^2 = \text{true}$

s_1^2: $NA \leftarrow 0$, $CF \leftarrow 1$

response 2: The fault is considered permanent and there are more than K active modules

$a_2^2 = (NA > K)$

s_2^2: $NA \leftarrow NA - 1$, $NPF \leftarrow NPF + 1$

response 3: The fault is considered permanent and there are only K active modules

$a_3^2 = (NA = K)$

s_3^2: $NA \leftarrow 0$, $NPF \leftarrow NPF + 1$

r_3: Reparation of a module

$a_3 = (NPF > 0) \wedge (CF = 0)$

response 1: The system is reactivated with K active modules

$a_1^3 = (NA = 0)$

s_1^3: $NA \leftarrow K$, $NPF \leftarrow NPF - 1$

response 2: The repaired module is incorporated into the active configuration

$a_2^3 = (NA > 0)$

s_2^3: $NA \leftarrow NA + 1$, $NPF \leftarrow NPF - 1$

r_4: Restart

$a_4 = (CF = 1) \wedge (NPF \leq K)$

response 1

$a_1^4 = \text{true}$

s_1^4: $NA \leftarrow N - NPF$, $CF \leftarrow 0$

Action rates

$$\lambda_1 = NA\,\lambda_p$$
$$\lambda_2 = NA\,\lambda_t$$
$$\lambda_3 = \mu$$
$$\lambda_4 = \mu_r$$

Response probabilities

$$c_1^1 = 1 - u_p \quad c_2^1 = u_p \quad c_3^1 = 1$$
$$c_1^2 = u_t \quad c_2^2 = \rho \quad c_3^2 = \rho$$
$$c_1^3 = 1 \quad c_2^3 = 1$$
$$c_1^4 = 1$$

Index $I = NA$

Fig. 2 shows the symbolic transition digraph and the evaluated transition digraph for a system with $N = 5$ modules that admits degradation up to $K = 3$ active modules. The generator state sg was chosen $(NA, NPF, CF) = (5, 0, 0)$ and the numbering of the states indicates the order in which they were obtained.

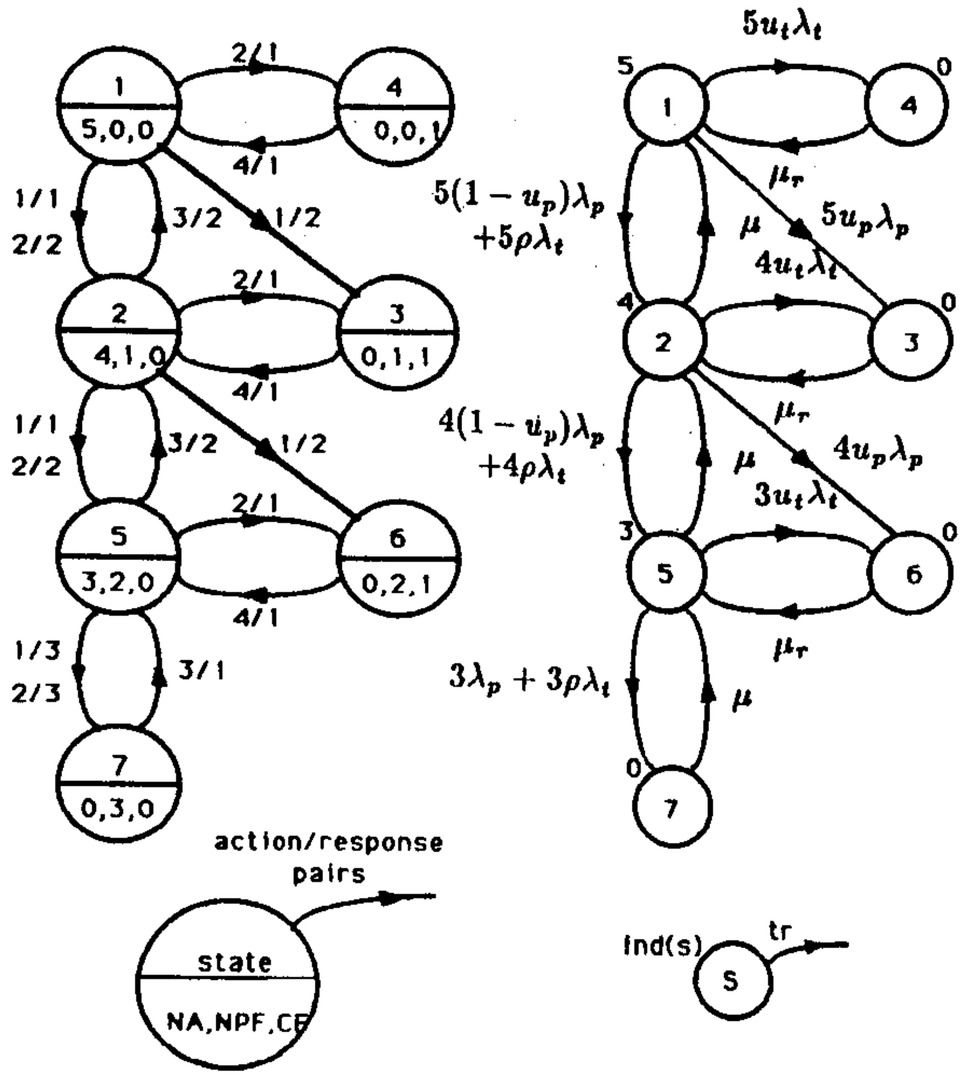


Figure 2: Symbolic transition digraph (left) and evaluated transition digraph (right) obtained for $N = 5$, $K = 3$, and $sg = (5, 0, 0)$.

3 EVALUATION

3.1 Measures

Taking into account the wide spectrum of features, operating environments and objectives of fault-tolerant computing systems, a widespread set of measures (see Table 1) has been incorporated into METFAC. Besides all/nothing measures, suitable for systems which from the user point of view are either working with full capability or failed, some performance and cost-related measures have been included. Some directly derived measures such as unavailability or unreliability, that have been included as evaluation options, have not been shown in Table 1. Most of them are well-known. Life-cycle measures have been used before (Arl83). Performance-related measures result from assigning a performance index $\eta(i)$ to each operative state i. When $\eta(i)$ has significance of service rate (e.g., flops/sec, transactions processed/sec) serviceability $S(u)$ and the related measures $CS(u)$, $MSFF$ and MSO are applicable. The definitions of these measures are obtained by replacing in those of, respectively, $R(t)$, $CR(t)$, $MTFF$ and MTU, the variable time t by the variable service u: i.e., $S(u)$ would be the probability that the total service accomplished by the system until its first failure is greater than u. Cost-related measures result from associating a cost rate $c(i)$ to each state i (operative or faulty). It has been pointed out (Mor80) that constant costs $c_t(i, j)$ associated with transitions should be considered also. These costs can be included by adding the quantities $\lambda_{ij} c_t(i, j)$to the cost rate $c(i)$.

Table 1: Measures incorporated into METFAC.

a/n: all/nothing — C: constant
PR: performance-related — VD: variable-dependent
CR: cost-related — *: applicable to non-repairable systems

Class	Type	Symbol	Denomination
a/n	C	AE	Steady-state availability
		MTU	Mean time up
		MTF	Mean time in failure
		$MTBF$	Mean time between failures
		$MTFF^*$	Mean time to first failure
		$MTTO$	Mean time to operation
	VD	$A(t)$	Availability
		$R(t)^*$	Reliability
		$M(t)$	Maintainability
		$CR(t)$	Life-cycle reliability
		$CM(t)$	Life-cycle maintainability
PR	C	PEE	Steady-state expected performance
		$MSFF^*$	Mean service to first failure
		MSO	Mean service during operation
	VD	$PE(t)^*$	Expected performance
		$S(u)^*$	Serviceability
		$CS(u)$	Life-cycle serviceability
CR	C	TCE	Steady-state expected cost rate
	VD	$TC(t)^*$	Expected cost rate

3.2 Formulation

Notation:

W	subset of operative states
F	subset of faulty states
$\eta(i)$	performance index in state i
$c(i)$	cost rate in state i
$\mathbf{A}$	$n \times n$ transition rate matrix of the MP modeling the system
$\mathbf{A}_{BC}$	matrix $(a_{ij})_{i \in B, j \in C}$
$\mathbf{A}^{\eta}_{BC}$	matrix $(a_{ij}/\eta(i))_{i \in B, j \in C}$
$\mathbf{A}_n$	matrix obtained from $\mathbf{A}$ by setting $a_{in} = 1$, $i = 1, 2, \ldots, n$
$\mathbf{D}$	restriction of some transition rate matrix to some set of states
$\mathbf{p}(t)$	probability vector for the MP with matrix $\mathbf{A}$
$\mathbf{p}$	steady-state probability vector ($\mathbf{p} = \lim_{t \to \infty} \mathbf{p}(t)$)
$\mathbf{q}(t)$	permanence probability vector
$\boldsymbol{\pi}$	average permanence time vector
$\mathbf{x}_B$	restriction of vector $\mathbf{x}$ to states in B
$\mathbf{o}_n$	$n \times 1$ column vector with all elements equal to 0, except the last one, equal to 1
$\mathbf{1}$	vector with all its elements equal to 1

The formulation of the measures can be done compactly as showed in Table 2.

The vectors $\mathbf{p}$ and $\boldsymbol{\pi}$ are the solutions to the systems:

$$\mathbf{p}^T \mathbf{A}_n = \mathbf{o}_n^T \tag{1}$$

$$\boldsymbol{\pi}^T \mathbf{D} = -\mathbf{q}^T(0) \tag{2}$$

Table 2: Formulation of the measures.

Group 1 $m = \mathbf{s}^T\mathbf{p}$	Group 2 $m(t) = \mathbf{s}^T\mathbf{p}(t)$	s_i	
AE	$A(t)$	1 for $i \in W$, 0 for $i \in F$	
PEE	$PE(t)$	$\eta(i)$ for $i \in W$, 0 for $i \in F$	
TCE	$TC(t)$	$c(i)$	
$MTBF^{-1}$		$\sum_{j \in F} \lambda_{ij}$ for $i \in W$, 0 for $i \in F$	
Group 3 $m = \mathbf{1}^T\boldsymbol{\pi}$	Group 4 $m(t) = \mathbf{1}^T\mathbf{q}(t)$	$\mathbf{D}$	$\mathbf{q}(0)$
$MTFF$	$R(t)$	$\mathbf{A}_{WW}$	$\mathbf{p}_W(0)$
$MTTO$	$1 - M(t)$	$\mathbf{A}_{FF}$	$\mathbf{p}_F(0)$
$MSFF$	$S(u)$	$\mathbf{A}^{\eta}_{WW}$	$\mathbf{p}_W(0)$
MTU	$CR(t)$	$\mathbf{A}_{WW}$	$(\mathbf{p}_F^T\mathbf{A}_{FW}\mathbf{1}_W)^{-1}\mathbf{p}_F^T\mathbf{A}_{FW}$
MTF	$1 - CM(t)$	$\mathbf{A}_{FF}$	$(\mathbf{p}_W^T\mathbf{A}_{WF}\mathbf{1}_W)^{-1}\mathbf{p}_W^T\mathbf{A}_{WF}$
MSO	$CS(u)$	$\mathbf{A}_{WW}$	$(\mathbf{p}_W^T\mathbf{A}_{WF}\mathbf{1}_W)^{-1}\mathbf{p}_W^T\mathbf{A}_{WF}$

The vectors $\mathbf{p}(t)$ and $\mathbf{q}(t)$ can be obtained by integrating:

$$\frac{d\mathbf{p}^T}{dt} = \mathbf{p}^T(t)\mathbf{A} \tag{3}$$

$$\frac{d\mathbf{q}^T}{dt} = \mathbf{q}^T(t)\mathbf{D} \tag{4}$$

The sparse numerical methods incorporated in METFAC to solve the problems (1), (2), (3), (4) will be briefly discussed in the next paragraphs. The reader interested in a more detailed exposition is referred to (Carr86).

3.3 Evaluation of the steady-state probability and average permanence time vectors

Two generic classes of numerical methods were considered (Dahl74): a) direct methods (Gaussian elimination and variants), b) iterative methods (Jacobi, Gauss-Seidel, SOR, etc.). Iterative methods do not modify the matrix and have minimum memory requirements. Their use is attractive when dealing with very large matrices and have been successfully used to evaluate the steady-state probability vector of MPs (Kauf81), (Nah84). However, the efficiency of these methods (measured in number of arithmetic operations) depends largely on the spectral properties of the iteration matrix. Some results presented in (Jenn77, p. 220) indicate that high stiffness ratios in the matrices of the systems produce very slow convergence rates. Unfortunately matrix $\mathbf{D}$ has often a high stiffness ratio. For instance, in evaluating the reliability of a repairable system, the maximum eigenvalue module is of the order of magnitude of the repair rates and the minimum eigenvalue module is close (Pag80) to the steady-state failure rate. In fault-tolerant computing systems the ratio between them may be as high as 10^6. Therefore, we concentrated on sparse direct methods.

It is known (Dhal74) that the triangular decomposition of diagonally dominant matrices with diagonal pivoting is numerically stable. Moreover, a technique $\mathcal{T}$ is available that suppress the cancellations when $a_{ii} < 0$ and $a_{ij} \geq 0$ for $i \neq j$. The technique is described in (Dow78).

For the problem (2) we obtain an optimal upper block-triangular form for the matrix $\mathbf{D}$ using topological sorting (Gust76). Taking advantage of this structure, it is only necessary to triangularize the diagonal blocks $\mathbf{D}^{ii}$. The technique $\mathcal{T}$ is directly applicable to matrices $\mathbf{D}^{ii}$ and an algorithm without cancellations can be

obtained for solve the problem. Models for non-repairable systems have usually acyclic digraphs. In this case the matrix obtained after topological sorting is upper triangular and the sparse solution to the problem is very efficient.

The technique $\mathcal{T}$ is not directly applicable to the matrix $\mathbf{A}_n$ in (1). Instead, we apply it to $\mathbf{A}$, that satisfies the required properties, and we obtain the triangular decomposition $\mathbf{A} = \mathbf{LU}$. Let $\mathbf{A}_n = \mathbf{L'U'}$. Considering how the triangularization is made it is easy to see that $\mathbf{L'} = \mathbf{L}$ and that $\mathbf{U'}$ only differs from $\mathbf{U}$ in the column n. The last column of $\mathbf{U'}$ is evaluated by solving the triangular system that results from comparing the columns n in $\mathbf{A}_n = \mathbf{LU'}$. After that, we get vector $\mathbf{p}$ by forward elimination and backward substitution. Taking into account the structure of $\mathbf{o}_n$ the following algorithm results:

1. *Triangular decomposition of* $\mathbf{A}$

2. *Evaluation on* $\mathbf{p}$ *of column* n *of* $\mathbf{U'}$

$$
\begin{aligned}
&p_1 \leftarrow 1\\
&\textbf{for } i = 2 \textbf{ to } n \textbf{ step } 1 \textbf{ do}\\
&\textbf{begin}\\
&\quad p_i \leftarrow 1 - \textstyle\sum_{j<i} l_{ij} p_j\\
&\textbf{end}
\end{aligned}
\qquad (5)
$$

3. *Forward elimination and backward substitution*

$$
\begin{aligned}
&p_n \leftarrow 1/p_n\\
&\textbf{for } i = n-1 \textbf{ to } 1 \textbf{ step } -1 \textbf{ do}\\
&\textbf{begin}\\
&\quad p_i \leftarrow -\textstyle\sum_{j>i} l_{ji} p_j\\
&\textbf{end}
\end{aligned}
$$

Theorem *No cancellations exist in the algorithm (5)*

Proof *The matrix* $\mathbf{A}$ *is diagonally dominant and verifies* $a_{ii} < 0$ *and* $a_{ij} \geq 0$ *for* $i \neq j$. *Then, using diagonal pivoting and the technique* $\mathcal{T}$ *no cancellations occur in the step 1 of the algorithm (Dow78).Moreover, it can be seen that* $l_{ij} \leq 0$. *The absence of cancellations in steps 2 and 3 follows from an analysis of the algorithm.*

The memory and time complexities of the sparse implementation of the problems (1), (2) largely depend on the fill-in produced in the evaluation of the LU decompositions. One well-known technique used to reduce this fill-in is heuristic ordering of states (Tar76). The matrix manipulations involved in this technique do not modify the properties of the matrix on which the methods are based. An adaptation of the "minimum-degree" algorithm (Rose72) to asymmetric matrices was selected to be used in a compact storage scheme. This scheme only reserves space for the non-zeroes elements in the triangular factors. The efficiency of the heuristic can be estimated looking at Table 3. The fill-in factor F (ratio between total number of non-zeroes after and before the triangularization) increases with the digraph size but remains low even for large sizes.

We note that the complexities of the algorithms that prepare the sparse solution of the system are similar to those of the solution itself. Topological sorting has been implemented in $O(n, nd)$ time and memory with an algorithm similar to the one presented in (Gust76). Heuristic ordering, fill-in evaluation and secondary storage construction have been implemented in $O(n(Fd)^2, nFd \log n)$ time and $O(n, nFd)$ memory.

3.4 Evaluation of the transient regime

Integration of systems (3), (4) can be carried out using several methods: spectral analysis, randomization, numerical integration, etc. Two factors were considered. First, sparseness was to be exploited. Second, as discussed in 3.3, $\mathbf{D}$ may have a very high stiffness ratio. Consideration of these two factors led us to select a sparse implementation of a stiff numerical integration method. In METFAC a well-known stiffly-stable implicit integration method: multiple-order, multiple-step Gear (Gear71) has been used. The method requires the solution of linear systems whose matrices can be proved to verify the same properties as $\mathbf{D}$ in problem (2). Then, the method described in the previous section can be applied. This eliminates cancellations in the evaluation of the LU decompositions. Cancellations are still possible in the forward elimination and backward substitution steps, but, if important, they will be detected and eliminated by changing locally the step or order of the integration formula. We have found that only about 20 or 40 LU factorizations and 200 or 400 integration steps are usually necessary to obtain the full transient regime with 6 or 7 significant digits for transition matrices with stiffness ratios as high as 10^7. This implies times for the evaluation of variable-dependent measures about 20 or 40 times greater that those required for constant measures, when n is large. For very large digraphs ($n \approx 1000$) the evaluation of variable-dependent measures may be quite slow. In order to solve this problem a technique called *state dissolution* has been developed and implemented. In this technique states with low occupation probabilities are "dissolved" among their "fathers". Transition rates from fathers are modified so that sojourn times in them after dissolution equal the average times since the entry in the father to the exit from the pair father-dissolved state. Also, indices in measures $PE(t)$ and $TC(t)$ (see Table 2) are modified. By respecting some rules, the measures in group 2 maintain their limit values (e.g., AE for $A(t)$), and the measures in group 4 maintain their integrals (e.g., $MTFF$ for $R(t)$). The technique reduces the number of states and is applied using an algorithm similar to the LU factorization with similar complexities. Then, the savings in time are significant.

4 IMPLEMENTATION

METFAC is organized in tasks implementing different levels: Model Specification, Transition Digraph Generation, Evaluation Preparation and Evaluation. Communication among tasks is established via files. The third level includes all the processing preparing the sparse numerical algorithms executed in the Evaluation level: state ordering, secondary storage pattern generation, etc. The generative specification is given in a file. This has several advantages: first, it allows the creation of a library of models; second, small modifications to the model can be done with little effort.

Special attention was paid to the use of low complexity algorithms

Table 3: Influence of the size of the transition digraphs in the processing times.

n: number of states
d: average degree
F: fill-in factor

			CPU times in seconds		
n	d	F	Transition Digraph Generation	Evaluation Preparation	Evaluation
7	2.857	1.250	0.49	0.43	0.68
24	4.041	1.784	0.80	0.68	0.78
53	4.415	2.291	2.19	1.50	0.97
94	4.585	2.842	5.43	3.47	1.32
147	4.680	3.188	12.1	6.57	1.82
249	4.763	3.829	32.6	15.7	3.05
479	4.835	4.775	118.	46.6	6.68
653	4.861	5.656	213.	88.8	11.0

in all levels. This was mandatory for the large size of the MPs for which the tool was intended. The code was written in Fortran 77 and Pascal, and about 6000 lines were necessary for the whole system.

Table 3 shows CPU times in VAX 750 for a series of transition digraphs obtained using the generalized model presented in (Cif82). The values for the evaluation level are for constant measures.

5 CONCLUSIONS

In this paper we have presented a tool for the modeling and evaluation of complex fault-tolerant computing systems. These systems require powerful specification methodologies. By choosing a high abstraction level, simplicity and wide scope of application have been obtained. Production rules have two advantages. First, they are easily derived from the knowledge of the system to be modeled; second, by tailoring their structure and using an embedded high level language, powerful and natural syntax is achieved. This is not only important to specify models for complex systems, but also to introduce approximations.

Sparse solutions to numerical problems are mandatory to process large size models. By taking advantage of the properties of the matrices safe numerical methods have been implemented that are efficient for medium and large size transition digraphs and can deal with high stiffness ratios, usually encountered in the practice. For the evaluation of constant measures, cancellations have been eliminated.

Acknowledgements

This research was supported in part by the "Comisión Asesora de Investigación Científica y Técnica" (CAYCIT) and the "Comisió Interdepartamental per la Recerca e Innovació Tecnològica" (CIRIT).

REFERENCES

(Arl83) J. Arlat and J.C. Laprie, *Performance-related Dependability Evaluation of Supercomputer Systems*, Proc. FTCS-13, Milano, Italy, June 28-30, 1983, pp. 276-283.

(BaFe82) A. Barr and E.A. Feigenbaum, *The Handbook of Artificial Intelligence, vol I* , HeirisTech Press, 1982, pp. 190-199.

(Bey81) B. Beyaert, G. Florin, P. Lonc and S. Natkin, *Evaluation of Computer Systems Dependability using Stochastic Petri Nets*, Proc. FTCS-11, Portland, Maine, USA, June 24-26,1981, pp. 79-81.

(Carr86) J.A. Carrasco, *Modelación y Evaluación de la Tolerancia a Fallos de Sistemas Distribuidos con capacidad de reconfiguración*, Doctoral dissertation, in preparation.

(Cif82) J. Ciferskỳ, *Generalized Markov Model for Reliability Evaluation of Functionally Degradable Systems* , Proc. FTCS-12, Santa Monica, California, USA, June 22-24, 1982, pp. 275-278.

(Cos81) A. Costes, J.E. Doucet, C. Landraut and J.C. Laprie, *SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems* , Proc. FTCS-11, Portland, Maine, USA, June 24-26, 1981, pp. 72-78.

(Dahl74) G. Dahlquist and Å. Björk, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

(Dow78) T. Downs and B.J. Parkinson, *Eliminating a Numerical Accuracy Problem in Mean Life Calculations*, IEEE Trans. on Reliability, vol. R-27, no. 4, October 1978, pp. 286-288.

(Gear71) C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

(Gust76) F. Gustavson, *Finding the Block Lower Triangular Form of a Sparse Matrix*, in "Sparse Matrix Computations", Academic Press, New York, 1976.

(Jenn77) A. Jennings, *Matrix Computations for Engineers and Scientists*, John Wiley and Sons, London, 1977.

(Kauf81) L. Kaufman, B. Gopinath and E.F. Wunderlich, *Analysis of Packet Network Congestion Control Using Sparse Matrix Algorithms*, IEEE Trans. on Communications, vol. COM-29, no. 4, April 1981, pp. 453-465.

(Mak82) S.V. Makam and A. Avižienis, *ARIES 81: A Reliability and Life-Cycle Evaluation Tool for Fault-Tolerant Systems*, Proc. FTCS-12, Santa Monica, California, USA, June 22-24, 1982, pp. 267-274.

(Mor80) J. Moreira de Souza, *A Unified Method for the Benefit Analysis of Fault-Tolerance*, Proc. FTCS-10, Kyoto, Japan, October 1-3, 1980, pp. 201-203.

(Nah84) J.M. Nahman, *Iterative Method for Steady State Reliability Analysis of Complex Markov Systems*, IEEE Trans. on Reliability, vol. R-33, no. 5, December 1984, pp. 406-409.

(NgAv80) Y.W. Ng and A. Avižienis, *A Unified Reliability Model for Fault-Tolerant Computers*, IEEE Trans. on Computers, vol. C-29, no. 11, November 1980, pp. 1002-1011.

(Pag80) A. Pagès, *Fiabilité des Systèmes*, Ed. Eyrolles, Paris, 1980.

(Rose72) D.J.Rose, *A Graph-Theoretic Study of the Numerical Solution of Sparse Positive Definite Systems of Linear Equations* , in "Graph Theory and Computing", Academic Press, New York, 1972.

(Tar76) R.E. Tarjan, *Graph Theory and Gaussian Elimination* , in "Sparse Matrix Computations", Academic Press, New York, 1976.