Addendum 1

IMPORT/EXPORT OF MODELS

1.1 OVERVIEW.

Model Importation and Exportation functions have been completely rewritten in order to make them easier to use. The exported model is more complete and can be easily moved. Main improvements are:

- assigned models are imported/exported with the model,
- names of models and assigned models are saved,
- one file contains all data corresponding to an exported model and can be easily moved or copied.

1.2 EXPORTATION.

Click on EXPORT and then select the model you want to export (Petri or Markov)(Fig. 1). When this is done, a new file named *Model_Name.petri* or *Model_Name.markov* is created in the EXPORT directory. This file contains:

- the model,
- the associated Markov model if it exists,
- all the corresponding assigned models.

This file can be moved or copied elsewhere in order to import it on another database.

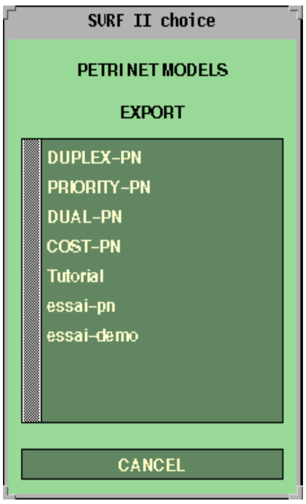

Fig. 1 Export Window.

Filename is used as model name during importation. If you change it, the name of the imported model will change accordingly.
Do not remove the suffix .petri or .markov as it is used by the IMPORT function to recognize the exported models.

1.3 IMPORTATION.

When you select IMPORT, a listing of all Petri Net models or Markov models inside the EXPORT directory is shown (Fig. 2). Select one of them and SURF-2 will import the model(s) and assigned model(s) in your database.

The names are identical with those in the original database except:

- if you have changed the name of the file,
- if there is a model or an assigned model in your database which has already the same name: in this case, model or assigned model is imported under a new name which is *name.suffix* (suffix is a number starting from 1).

Fig. 2 Import Window

Addendum 2

EXPORTATION OF RESULTS

2.1 OVERVIEW.

You can now export your results in a formatted text file that you can edit or import in a spreadsheet. This file contains all the information related to the model folder (names of models, assigned models, measure, ...) and the values of results and of global variables.

2.2 HOW TO.

First, verify that you have defined correctly the IMPORT/EXPORT directory either in your startup file or in the ENVIRONMENT window (MISC Menu). The results file will be saved into this directory under the name *model_folder_name.export*.

Click on the RESULTS menu and select one of the model folders. When the results window shows, just click on EXPORT button (Fig. 1) and a new file containing the results is created (Fig. 2).

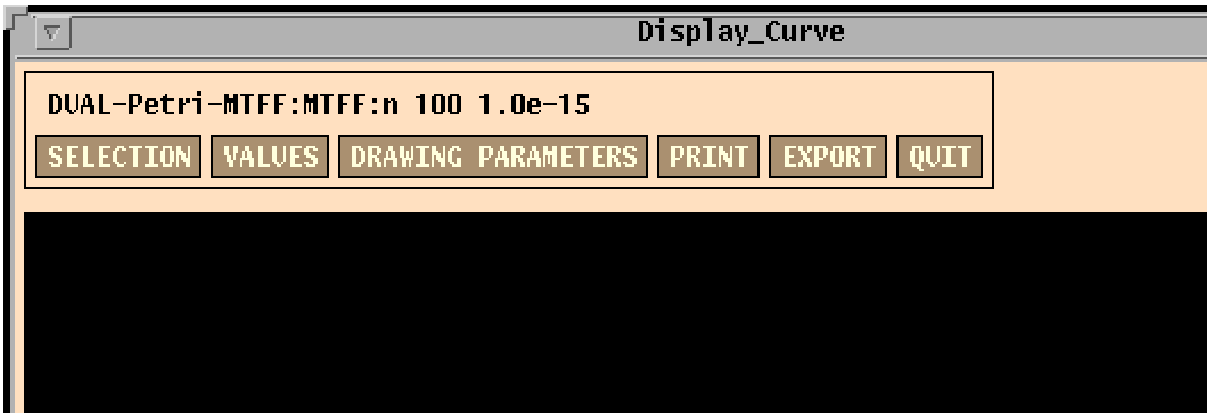

Fig. 1 Results Window

Fig. 2 Message Window

2.3 FILE DESCRIPTION

The file is composed of a set of lines, each line may contain several values/strings separated by tabs.

- first line contains the **name of the model folder** and the **measure**,
- for each assigned model, format is as follows:
  - a line containing the **name of assigned model**, the **name of model**, the **name of partition** and if necessary the **name of reward**,
  - in case of reward measures, a line with the string **"discount"** and its **value**,
  - a line with the **names of global values** and the string **"result"**,
  - a set of lines with **the combinations of global values** and of **result values**.

Here is an example with one assigned model, and one global value:

```
DUAL-Petri-MTFF MTFF
DUAL-NUM        DUAL-PN     Part1
LS      result
1.00000000000000E-04    1.192640692640693E+03
2.50000000000000E-04    1.122222222222222E+03
5.00000000000000E-04    1.033333333333333E+03
```

Addendum 3

MODEL FOLDER PRINTING

3.1 OVERVIEW.

You can now print a model folder and all its content (assigned models, partitions, global values, ...).

3.2 HOW TO.

You just have to click on the new PRINT button (see Fig. 1 below) which is shown on the Model Folder Editor window.

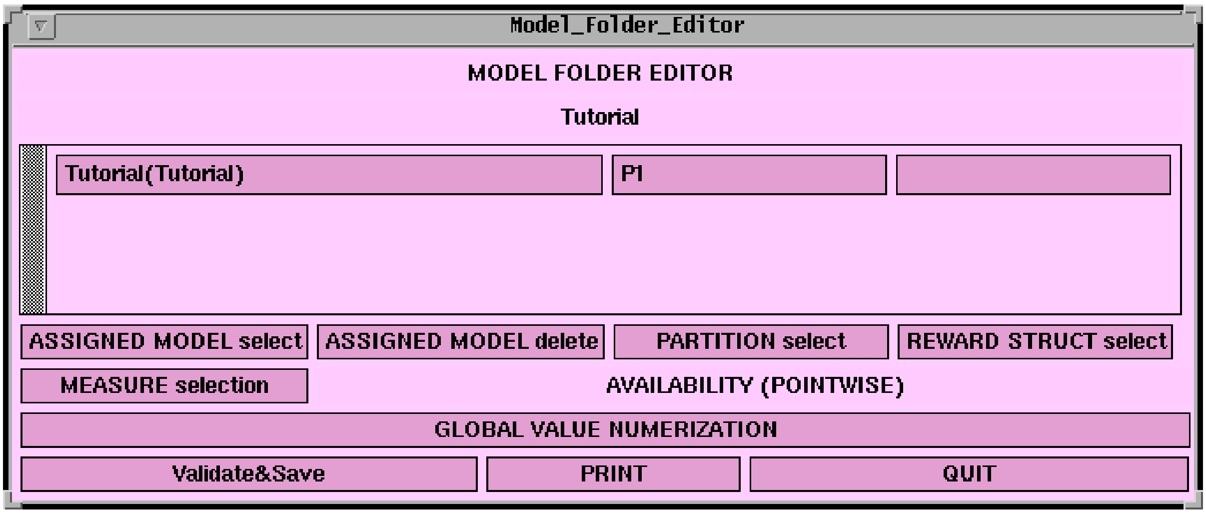

Fig. 1 Model Folder Editor window.

The printout contains the name of the model folder, the name of the measure, the set of assigned models and corresponding partitions, and all the global values.

Addendum 4

EDGES DISPLAY

4.1 OVERVIEW.

In the Model Editor, you can select a node and display only edges coming from or to this node.

4.2 HOW TO.

Select a node (place or transition), and click on mouse third button. All edges but those coming from or to this node will be hidden (Fig. 1). Select another node and click again on third button and edges related to this node will be shown.

Click on the second button and all edges are shown (Fig. 2).

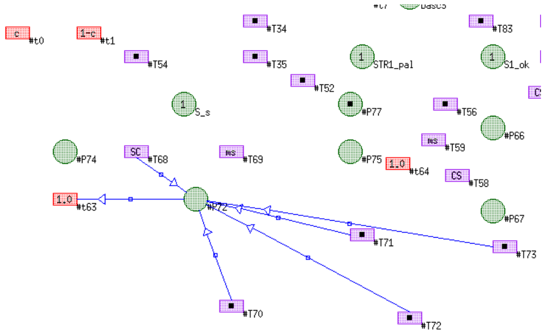

Fig. 1 Only #P72 related edges are shown.

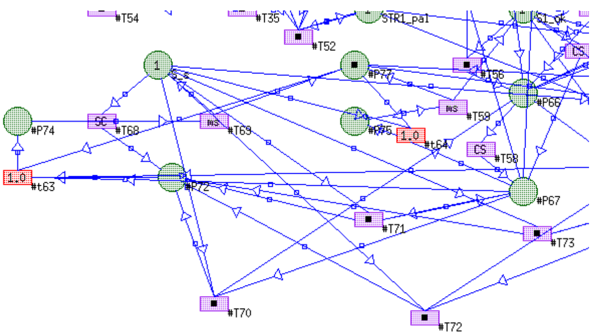

Fig. 2 All edges are shown.

Addendum 5

PREDICATES

5.1 OVERVIEW.

Predicates allow user to add textual firing conditions to transitions. Firing condition in a Petri Net depends on the input places of transitions as well as the weight of the input edges. The transition is fired when the marking of all input places are greater or equal of connecting edges in case of normal edges or when the marking is equal to zero in case of inhibitor edges.

Up to now in SURF-2, these edges and transitions were drawn by user. Thus the conditions were expressed only graphically.

But in large models, the number of edges can be dramatically high making the Petri Net unreadable and very difficult to check and modify. We have thus introduced textual firing conditions we named predicates (actually they are predicates). These predicates enable to define conditions without the need of drawing all. The firing condition is thus an AND function of predicate and existing graphic condition.

A predicate is composed of boolean variables (place names with weight attributes) connected by logical operators NOT, AND, OR (full syntax is described further). When the predicate becomes true and the graphic conditions (marking of places graphically connected) are true, the transition is fired. Boolean variables become true according to their weights and the marking of related places.

Predicates do not fully substitute graphic edges and transitions, they must be added to an existing correct Petri Net.

5.2 HOW TO.

Select Init Object command and click on a transition, the transition attributes window appears (Fig. 1). You can now enter your firing condition in the predicates field according to the syntax described below.

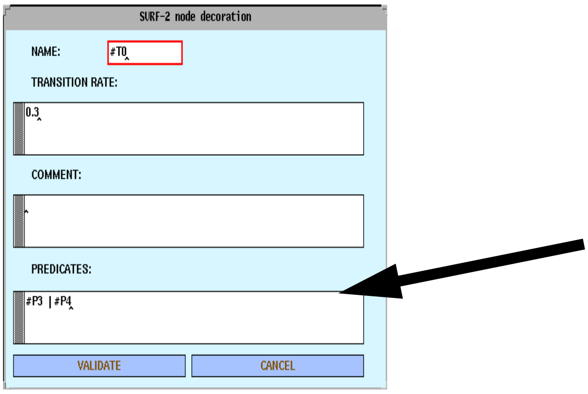

Fig. 1 Transition attributes with predicates.

Transitions containing predicates are highlighted as shown in Fig. 2.

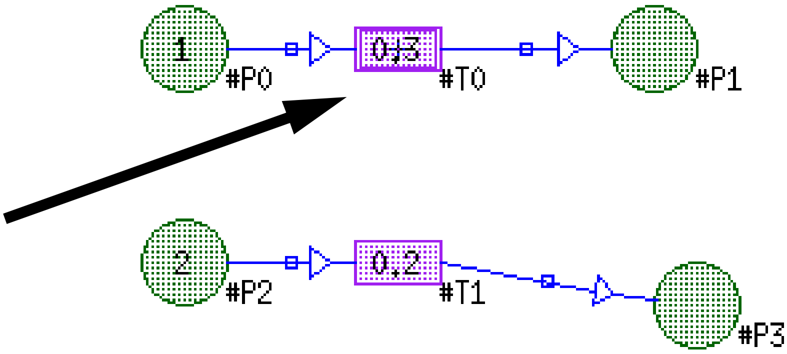

Fig. 2 Transition #T0 contains predicates.

5.3 SYNTAX

A predicate defines the firing condition in terms of a boolean combination of place marking test. The marking test of one place can be:

- "marking of this place is equal to 0",
- "marking of this place is greater or equal than a non-null positive number".

The marking of a place is represented by an optional weight attribute associated with the name of the place (*name*:*weight*). The default value of marking is 1.

P is TRUE when marking of *P* is greater or equal than 1.

P:n, with *n* > 0, is TRUE when marking of *P* is greater or equal than *n*.

P:0 is TRUE when marking of *P* is equal to 0.

Connect operators used in predicates are **&** (AND), **|** (OR) and **~**(NOT).

The priority order of operators is - from highest to lowest - **~**, **&**, **|**.

A predicate must be written as a sum of min-term, the full syntax is as follow:

<predicate> ::= <min-term> [**|** <predicate>]

<min-term> ::= <place-info> [**&** <min-term>]

<place-info> ::= <place-name>

<place-info> ::= <place-name>**:**<marking>

<place-info> ::= **~** <place-name>

<marking> ::= 0...

Note: Parenthesis are not allowed.

Predicate evaluation:

place-infoA & place-infoB

is TRUE if place-infoA and place-infoB are TRUE.

min-termA | min-termB

is TRUE if min-termA is TRUE or min-termB is TRUE

~ place-name

is TRUE if marking of place-name is 0 (= *place-name***:***0)*

Examples:

P1:2 & P2 | P3

is TRUE if marking of P1 is greater or equal than 2 and marking of P2 is greater or equal than 1, or marking of P3 is greater or equal than 1.

~ P1 | P2:4 & P3:3

is TRUE if marking of P1 is equal to 0 or, if marking of P2 is greater or equal than 4 and marking of P3 is greater or equal than 3.

5.4 SEMANTICS

The different components of a predicate do not change the semantics of Petri Nets. They are only a textual way to represent graphic elements, such as edges and transitions. The semantics is described below:

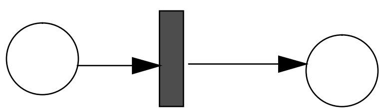

Fig. 3 Original graphic.

place:n

an input bidirectional edge, whose weight is n, between the named place and the selected transition (Fig. 4).

place

an input bidirectional edge whose weight is 1(Fig. 4).

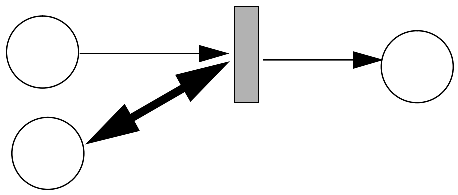

Fig. 4 Simple predicate.

~place or **place:0**

an input inhibitor edge (Fig. 5).

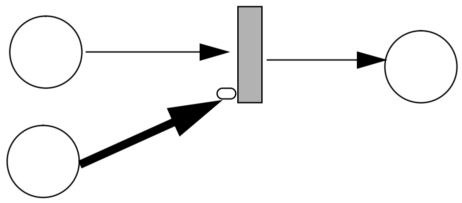

Fig. 5 NOT.

place1:n1 & place2:n2 & ...

a set of input bidirectional edges between named places and selected transition (Fig. 6).

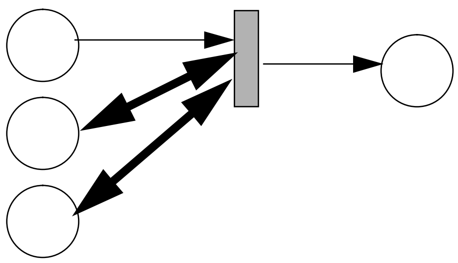

Fig. 6 AND.

place1:n1 & place2:n2 & ... | place11:n11 & ...

the first min-term represents what is described above, the second min-term is a new transition, whose rate is equal to selected one's, a set of bi-directional edges between named places and this transition, and output edges from this transition to output places of selected transition. New output edges have the same weights as existing ones (Fig. 7).

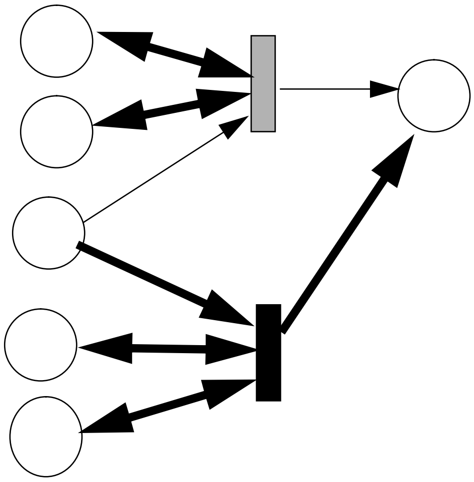

Fig. 7 OR.

Predicates do not create new tokens nor delete existing ones. The marking of referenced places in the predicate do not change when transition is fired.

The actual firing condition of a transition is "graphic condition AND textual condition". Both must be TRUE to fire it.

Do not add a predicate, to a transition:

- *which does not have any input edges,*
- *which has only input bidirectional edges.*

Addendum 6

COMPUTATION

6.1 COMPUTATION METHODS.

There are 3 computation methods in SURF-2: CGS, Integration, Randomisation

6.1.1 CGS

CGS method is used for asymptotic and mean measures. It is a bounded method whose limits are the maximum number of steps and the accuracy defined by the SURF2_WIZARD variable in surf startup script. Default values are 200 and 1e-15.

CGS method may not converge in some cases, if values are too small or the difference between them is too large. An error message is then displayed.

First solution consists in increasing the number of steps (you should not decrease accuracy), but it does solve the problem in a few number of cases. A better method would be to change the parameters of your model and of your model folder.

6.1.2 Integration

Integration method is used to compute pointwise measures. It is generally faster than randomisation for large differential systems. Based on CGS, it is also bounded by the same parameters: accuracy and number of steps.

Same problem may arise than in CGS, method may not converge. Solutions are identical.

But in case of no-convergence of integration method, SURF-2 automatically switches to randomisation method and tries to redo the computation.

6.1.3 Randomisation

Randomisation method is generally used is small differential systems. Slower than integration method, it is more robust and can solve some sytems unsolved by integration. Unfortunately the computation is not bounded and when method cannot succeed to give results or when elapsed time seems too long, user has to stop the computation manually.

An information is then displayed about the step (time) computation stopped. User can then split the computation interval according to theses indications.

Do not abort the computation too early as the the randomisation process can be long.

6.2 RESULTS

Most results are probabilities and must be in the interval [0,1], but in some cases, due to the inaccuracy of floating point computation, results can be greater than 1 or lower than 0. In this case, a warning message is displayed. Be careful, because results are automatically rounded to 0 or 1 by SURF-2.

If the measures are such low or high, it means that your parameterized model is too or not enough dependable and that you have to change it.

Addendum 7

INVARIANTS

7.1 USER-DEFINED INVARIANTS.

User-defined place invariants can be explicitly defined in the Petri Net model. Select the INV button (See Chapter 6, Fig 6.25) and enter the invariants you want to be verified following syntax shown in Appendice 1. The place invariants are verified by the verify function (See Chapter 9).

It has to be to noticed that all invariants (place and transition invariants) are automatically detected and displayed by the verify model function. But in large models, where number of invariants can be high, it may be helpful for model validation to define a priori some invariants which must be in the model.

Here is an example of user-defined place invariants:

```
inv1:m(P1)+2*m(P2)+m(P3)=5;
inv2:3*m(P2)-m(P4)=2;
```

7.2 VERIFY INVARIANTS

The Model verify function shows all transition and place invariants of the assigned model. The result is divided in three parts shown below.

First part consists in all place invariants

```
*** MARKING INVARIANTS :
m(mach_OK)  + m(Wait)  + m(Workshop) = 4
m(Free_Tech)  + m(Workshop) = 1

All the places are there.
```

Second part consists in all transition invariants

```
*** FIRING INVARIANTS :
[  Failure  Repair  RendezVous  ]

All the transitions are there
```

Third part consists in user-defined invariants.

```
Invariant inv :   ok !

Invariant inv2 :   not invariant !
```

Appendix 4

THE MATHEMATICAL EXPRESSIONS OF THE DEPENDABILITY MEASURES

1. Characterization of the basic dependability measures

1.1 Notation

E is the set containing the states of the Markov chain describing the model behavior, where **|E|** is the cardinal number of **E**.

Concerning the expression of the measures, we assume the knowledge of:

- the transition rates between states of the graph modeling the system behavior: λ_{ij}, $\forall$ i, j $\in$ {1, 2, …, |E|}, i   j,
- the initial distribution in the states: p_{0i}, $\forall$ i $\in$ {1, 2, …, |E|}
- the associated reward structure (when it has been defined):

  - yield rates proportional to the time spent in each state of the graph: y_i , $\forall$ i $\in$ {1, 2, …, |E|},

  - bonus acquired for each transition between the states of the graph: b_{ij}, $\forall$ i, j $\in$ {1, 2, …, |E|}, i   j.

Subsequently, X is subset the states of the graph for which the measure will be calculated, and Y the complementary subset ; X and Y define a partition of E. The measures are significant only if X corresponds to a set of connected states (ergodic Markov subchain). These measures are expressed using the following matrix notation.

Transition rates matrix: $\Lambda = [\ \lambda_{ij}\ ] = \begin{bmatrix} \Lambda_{xx} & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_{yy} \end{bmatrix}$

λ_{ij} : events /unit of time ; $\lambda_{ij} \in \Re^{+}$ (i j) ;$\lambda_{ii} \in \Re^{-}$ et $\lambda_{ii} = - \sum_{j=1,\, j\neq 1}^{|E|} \lambda_{ij}$

thus : $\sum_{j=1}^{|E|} \lambda_{ij} = 0$

Initial probability vector[1] : $\mathbf{P}_0 = [\, p_{0i} \,] = [\, \mathbf{P}_{X0} \;\; \mathbf{P}_{Y0}]$

p_{i0} : probability ; $0 \le p_{i0} \le 1$; $\sum_{i=1}^{|E|} p_{i0} = 1$;

otherwise: $\mathbf{P}_0 = \mathbf{P}_0$ so that $p_{10} = 1$ and $p_{j0} = 0$, $j \in \{2, 3, \ldots, |E|\}$

Yield rate vector (sojourn time in each state) : $\mathbf{Y} = [\, y_i \,] = [\, \mathbf{Y}_X \;\; \mathbf{Y}_Y]$

y_i : reward/unit of time; $y_i \in \Re$

Bonus matrix (transitions between states): $\mathbf{B} = [\, b_{ij} \,] = \begin{bmatrix} \mathbf{B}_{XX} & \mathbf{B}_{XY} \\ \mathbf{B}_{YX} & \mathbf{B}_{YY} \end{bmatrix}$

b_{ij} : reward ; $b_{ij} \in \Re$; $b_{ii} = 0$ (matrix with a zero diagonal)

Discount rate (cost analysis of the reward — discounted reward) : γ

γ : depreciation/unit of time ; $\gamma \in \Re^+$; $s(0) = \exp(-\gamma t)\, s(t)$.

Transient-state probabilities matrix: $\Omega(t) = [\, \omega_{ij}(t)] = \begin{bmatrix} \Omega_{XX}(t) & \Omega_{XY}(t) \\ \Omega_{YX}(t) & \Omega_{YY}(t) \end{bmatrix}$

$\omega_{ij}(t)$: probability to be in the state **j** at **t** time, the process being in state **i** at the initial time 0),

$0 \le \omega_{ij}(t) \le 1$; $\sum_{j=1}^{|E|} \omega_{ij}(t) = 1$

Transient-state probabilities vector: $\mathbf{P}(t) = [\, p_i(t)] = [\, \mathbf{P}_X(t) \;\; \mathbf{P}_Y(t)] = \mathbf{P}_0 \; \Omega(t)$

$p_i(t)$: probability ; $0 \le p_i(t) \le 1$; $\sum_{i=1}^{|E|} p_i(t) = 1$

Asymptotic-state probabilities vector: $\mathbf{P} = \lim_{t \to} \mathbf{P}(t) = [\, p_i] = [\, \mathbf{P}_X \;\; \mathbf{P}_Y]$

p_i : probability ; $0 \le p_i \le 1$; $\sum_{i=1}^{|E|} p_i = 1$

Mean sojourn times vector: $\mathbf{T} = [\, t_i] = [\mathbf{T}_X \;\; \mathbf{T}_Y] t_i$: unit of time ; $t_i \in \Re^+$

1. All the considered vectors are line-vectors.

Sojourn times in-state matrix in interval [0;t] : (t) = [$\sigma_{ij}(t)$] $= \int_0^t \Omega(u)du$

$$= \begin{bmatrix} \Sigma_{XX}(t) & \Sigma_{XY}(t) \\ \Sigma_{YX}(t) & \Sigma_{YY}(t) \end{bmatrix}$$

$\sigma_{ij}(t)$: unit of time (expected sojourn time in state **j** in the interval [0,t], the process being in state **i** at time 0) ; $0 \le \sigma_{ij}(t) \le t$ and $\sum_{i=1}^{|E|} \sigma_{ij}(t) = t$

Sojourn-times-in-state vector for a given interval: $\mathbf{L}(t) = [l_i(t)] = \mathbf{P}_0(t) = \int_0^t \mathbf{P}(u)du$

$$= [\ \mathbf{L}_X(t)\ \ \mathbf{L}_Y(t)]$$

$l_i(t)$: unit of time (expected sojourn time in state **i** in the interval [0;t]);

$0 \le l_i(t) \le t$ and $\sum_{i=1}^{|E|} \mathbf{l}_i(t) = t$

Identity vector (elements equal to 1) : $\mathbf{1} = [\ \mathbf{1}_X\ \ \mathbf{1}_Y\ ]$

Zero vector (elements equal to 0) : $\mathbf{0} = [\ \mathbf{0}_X\ \ \mathbf{0}_Y\ ]$

Summation vector : $\mathbf{U} = [\ \mathbf{1}_X\ \ \mathbf{0}_Y\ ]$ (elements equal to 1 for X, elements equal to 0 for Y)

$\mathbf{U}' = [\ \mathbf{0}_X\ \ \mathbf{1}_Y\ ]$ (elements equal to 0 for X, elements equal to 1 for Y)

Identity matrix: $\mathbf{I} = [\ i_{ij}] = \begin{bmatrix} \mathbf{I}_{XX} & \mathbf{I}_{XY} \\ \mathbf{I}_{YX} & \mathbf{I}_{YY} \end{bmatrix}$ $i_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \ne j \end{cases}$

$\mathbf{I}_{XY} = 0$; $\mathbf{I}_{YX} = 0$ and $\mathbf{I}_{XX}$, $\mathbf{I}_{YY}$ identity matrix.

Laplace transform: $F(s) = \mathcal{L}\ \{F(t)\}$

Inverse of the Matrix M : M^{-1}

Transposed matrix $\mathbf{M}$ (or transposed vector of $\mathbf{V}$) : $\mathbf{M}^T$ ($\mathbf{V}^T$)

1.2. States probabilities

The Chapmann-Kolmogorov equation can be written as:

$$\frac{d}{dt}\,\mathbf{P}(t) = \mathbf{P}(t)\,\Lambda \quad (1.1)$$

The use of Laplace transform enables to write (1.1) in the following form:

$$\mathbf{P}(s) = L\,\{\mathbf{P}(t)\} = \mathbf{P}_0\,(s\,\mathbf{I} - \Lambda)^{-1} \quad (1.2)$$

1.2.1. Pointwise states probabilities

a) Case of an ergodic Markov chain

Solving (1.1) leads to: $\mathbf{P}(t) = \mathbf{P}_0 \exp(\Lambda\, t) = \mathbf{P}_0\, \Omega(t)$(1.3)

The instantaneous probability to be in the states of subset **X** with respect to vector $\mathbf{P}_0$ is thus given by:

$$\mathbf{P}_X(t) = \mathbf{P}_X(t)\,(\mathbf{1}_X)^T = \mathbf{P}_0 \exp(\Lambda\, t)\,(\mathbf{U})^T \quad (1.4)$$

$\mathbf{U}$ is a summation vector whose the $|\mathbf{X}|$ first elements are equal to 1, the other being equal to 0.

$(\bullet)^T$ designates the transposition operation.

b) Case of a Markov chain with an absorbing state

Consider that **Y** is a subset of absorbing states. This case corresponds to the computation of the probability to be in the states of **X** before absorbtion by the states of subset **Y**, with respect to vector $\mathbf{P}_0 = [\,\mathbf{P}_{0X}\ \ \mathbf{0}_Y\,]$. From (1.1) and from the partition of Λ we can write:

$$\mathbf{P}_X(t) = \mathbf{P}_{X0} \exp(\Lambda_{XX}\, t) = \mathbf{P}_{X0}\, \Omega_{XX}(t) \quad (1.5)$$

$$\mathbf{P}_X(t) = \mathbf{P}_X(t)\,(\mathbf{1}_X)^T = \mathbf{P}_{0X} \exp(\Lambda_{XX}\, t)\,(\mathbf{1}_X)^T \quad (1.6)$$

1.2.2. Asymptotic states probabilities

a) Case of an ergodic Markov chain

From relation (1.1), according to an asymptotic behavior, we have:

$$\lim_{t\to} \frac{d}{dt}\,\mathbf{P}(t) = \mathbf{0} = \mathbf{P}\,\Lambda \quad (1.7)$$

$\mathbf{0}$ is the null vector.

Moreover $\mathbf{P}$ being a stochastic vector,

$$\sum_{i} p_i = \mathbf{P}\ \mathbf{1}^T = \mathbf{1} \quad (1.8)$$

$\mathbf{1}$ is the identity vector.

To make the computation simpler, relations (1.7) and (1.8) can be combinated as follows:

$$\mathbf{0}_m = \mathbf{P}\ \Lambda_m$$

$$\mathbf{P} = \mathbf{0}_m\ (\ \Lambda_m)^{-1} \quad (1.9)$$

where $\mathbf{0}_m$ is the modified null vector for which element m only is equal to 1,

Λ_m is the modified transition rate matrix where the m^{th} column is replaced by vector $\mathbf{1}^T$,

$(\bullet)^{-1}$ is the matrix inversion operation.

The asymptotic probability to be in the states of subset **X** is then given by:

$$\mathbf{P}_X = \mathbf{P}_X\ (\mathbf{1}_X)^T = \mathbf{P}\ \ (\mathbf{U})^T \quad (1.10)$$

b) Case of a Markov chain with an absorbing state

Consider that Y is a subset of absorbing states; in the asymptotic behavior, we have:

$$\mathbf{P}_X = \mathbf{P}_X\ (\mathbf{1}_X)^T = 0$$

$$\mathbf{P}_Y = \mathbf{P}_Y\ (\mathbf{1}_Y)^T = \mathbf{1} \quad (1.11)$$

This case deserves a practical interest only if we consider the probability to be in a subset of absorbing states of **Y** ; let **Z** be this subset and **H** the complementary subset. We then have the following partitions:

Transition rates matrix:

$$\Lambda = \begin{bmatrix} \Lambda_{XX} & \Lambda_{XY} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \Lambda_{XX} & \Lambda_{XZ} & \Lambda_{XH} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Asymptotic-state probabilities vector: $\mathbf{P} = [\ \mathbf{0}\ \ \mathbf{P}_Y] = [\ \mathbf{0}\ \ \mathbf{P}_Z\ \ \mathbf{P}_H]$

In this case, considering the form and the partition of Λ , relation (1.1) can be written as:

$$\frac{d}{dt}\mathbf{P}_X(t) = \mathbf{P}_X(t)\,\Lambda_{XX}$$

$$\frac{d}{dt}\mathbf{P}_Y(t) = \mathbf{P}_X(t)\,\Lambda_{XY} \Rightarrow \frac{d}{dt}\,[\,\mathbf{P}_Z(t)\;\;\mathbf{P}_H(t)\,] = \mathbf{P}_X(t)\,[\,\Lambda_{XZ}\;\;\Lambda_{XH}\,]$$

Using the Laplace transform leads to:

$$\mathbf{P}_X(s) = \mathbf{P}_{0X}\,(s\,\mathbf{I}_{XX} - \Lambda_{XX})^{-1}$$

$$s\,\mathbf{P}_Y(s) = \mathbf{P}_{0X}\,(s\,\mathbf{I}_{XX} - \Lambda_{XX})^{-1}\;\Lambda_{XY}$$

that is :

$$s\,[\mathbf{P}_Z(s)\;\;\mathbf{P}_H(s)] = \mathbf{P}_{0X}\,(s\,\mathbf{I}_{XX} - \Lambda_{XX})^{-1}\;[\Lambda_{XZ}\;\;\Lambda_{XH}]$$

The theorem of the final value: $\mathbf{F} = \lim_{t\to} \mathbf{F}(t) = \lim_{s\to 0} L\,\{\frac{d}{dt}\,\mathbf{F}(t)\} = \lim_{s\to 0} s\,\mathbf{F}(s) - \mathbf{F}(0)$, allows, for example, to write:

$$\mathbf{P}_Z = -\,\mathbf{P}_{0X}\,(\Lambda_{XX})^{-1}\,\Lambda_{XZ} \quad (1.12)$$

$\mathbf{P}_Z$ is the vector of the asymptotic probabilities to be in the subset **Z** with respect to $\mathbf{P}_{0X}$. Considering that $\mathbf{P}_Z$ corresponds to the asymptotic probability of presence in **Z**, we further have:

$$P_Z = \mathbf{P}_Z\,(\mathbf{1}_Z)^T = -\,\mathbf{P}_{0X}\,(\Lambda_{XX})^{-1}\,\Lambda_{XZ}\;(\mathbf{1}_Z)^T \quad (1.13)$$

1.3. Mean sojourn Times

To obtain a significant measure, when computing the **mean sojourn time in the states of subset X**, we consider that Y is a subset of absorbing states[2]. The vector of the mean sojourn time in the states of X before absorbtion by the states of Y is defined as:

$$\mathbf{T}_X = \int_0^\infty \mathbf{P}_X(t)\,dt = \mathbf{lim}_{s\to 0}\,\mathbf{P}_X(s) \quad (1.14)$$

and thus, from relation (1.2):

$$\mathbf{T}_X = -\,\mathbf{P}_{0X}\,(\Lambda_{XX})^{-1} \quad (1.15)$$

$$\mathbf{T}_X\,\Lambda_{XX} = -\,\mathbf{P}_{0X}$$

2. In the opposite case, the mean sojourn time would be infinite.

The mean sojourn time in the states of subset **X** with respect to vector $\mathbf{P}_{0X}$ is thus given by:

$$T_X = \mathbf{T}_X (\mathbf{1}_X)^T = - \mathbf{P}_{0X} (\Lambda_{XX})^{-1} (\mathbf{1}_X)^T \quad (1.16)$$

$(T_X)^{-1} = \rho_X$ can be interpreted as the mean absorbing rate by subset **Y** (flow from X to Y). When **Y** is partitioned into two subsets **Z** and **H** (see paragraph 1.2.2.b.), this rate can be split up into two rates $\rho_{X/Z}$ and $\rho_{X/H}$, defined as:

$$\rho_{X/Z} = (T_{X/Z})^{-1} = P_Z (T_X)^{-1} \quad (1.17)$$

$$\rho_{X/H} = (T_{X/H})^{-1} = P_H (T_X)^{-1}$$

where, P_Z and P_H represent the absorbing probabilities by **Z** and **H** respectively, and so that:

$$\rho_X = \rho_{X/Z} + \rho_{X/H}$$

Relations (1.13) and (1.16) allow, for example, to write:

$$P_Z = \mathbf{T}_X \Lambda_{XZ} (\mathbf{1}_Z)^T \quad (1.18)$$

From (1.17), **the mean sojourn time in X before absorbtion by Z [$T_{X/Z}$]** is:

$$T_{X/Z} = \frac{T_X}{P_Z} = \frac{\mathbf{T}_X(\mathbf{1}_X)^T}{\mathbf{P}_Z(\mathbf{1}_Z)^T} = \frac{\mathbf{T}_X}{\mathbf{P}_Z(\mathbf{1}_Z)^T}(\mathbf{1}_X)^T = \frac{\mathbf{P}_{0X}(\Lambda_{XX})^{-1}}{\mathbf{P}_{0X}(\Lambda_{XX})^{-1}\Lambda_{XZ}}(\mathbf{1}_X)^T \quad (1.19)$$

Vectors $\mathbf{T}_{X/Z}$ and $\mathbf{T}_{X/H}$ can also be introduced to characterize the mean sojourn times in the states of **X** before absorbtion by **Z** and **H**;, respectively. It is quite easy to deduce $T_{X/Z}$ from relation (1.19) and from the following definitions:

$$T_{X/Z} = \mathbf{T}_{X/Z} (\mathbf{1}_X)^T$$

$$T_{X/H} = \mathbf{T}_{X/H} (\mathbf{1}_X)^T$$

1.4. Reward

The **reward W(t) calculated in the interval [0, t]** takes into account the cumulative time effect on the whole states of the model using the **reward structure** settled by the vector of **yield rates** in relation to the presence in the states $\mathbf{Y} = [\, y_i \,]$ and by the matrix of **bonuses** associated to transitions $\mathbf{B} = [\, b_{ij} \,]$.

In the case of a cost analysis, where the notion of reward involved is a **discounted reward**, it is quantitatively different to gain or to spend an amount of money immediately or later on; we thus assume that an amount of money invested at time zero with the intention to yield interests, would earn more money after **N** periods of time.

The model of depreciation of money used in SURF-2 is an exponential model detailed hereafter:

Assuming that s(t) corresponds to an amount of money at time t, the value of this money at time zero taken as a reference, is given by:

$$s(0) = \exp(-\gamma t)\, s(t) \quad (1.20)$$

where γ is the discount rate.

In this case, the reward parameters must account for this depreciation; the yield rates and the bonuses at time **t**, brought back to the initial reference time of the interval, are thus computed as:

$$\mathbf{Y}(t,\gamma) = [y_i(t,\gamma) = y_i\ e^{-\gamma t}] \text{ and } \mathbf{B}(t,\gamma) = [b_{ij}(t,\gamma) = b_{ij}\ e^{-\gamma t}] = \mathbf{B}\, I\ e^{-\gamma t} \quad (1.21)$$

We thus obtain for the cumulative reward $W(t,\gamma)$:

$$W(t,\gamma) = \sum_i y_i \int_o^t e^{-\gamma u} V_i(u)du + \sum_i \sum_{j!=i} b_{ij} \int_0^t e^{-\gamma u} S_{ij}(u)du \quad (1.22)$$

$$W(t,\gamma) = \sum_i y_i \int_o^t y_i(u,\gamma) V_i du + \sum_i \sum_{j!=i} \int_0^t b_{ij}(u,\gamma) S_{ij}(u)du \quad (1.23)$$

where, $V_i(t)$ = 1 if the state variable X(t) = i,

0 otherwise))

and $S_{ij}(t)$ dt = 1 if X(t) = i and X(t +) = j

0 otherwise))

Relation (1.23) is the cumulative reward earned in the [0, t] interval expressed with respect to the initial time zero:

- during the sojourn in each state of the model,
- during each transition between states of the model.

Note that if the discount rate γ is equal to zero, relation (1.23) can be simply written as:

$$W(t,0) = W(t) = \sum_i y_i \int_o^t V_i(u)du + \sum_i \sum_{j!=i} b_{ij} \int_0^t S_{ij}(u)du \quad (1.24)$$

We can thus write (1.24) as follows:

$$W(t) = W(t,\gamma) = \sum_i y_i t_i(t) + \sum_i \sum_{j!=i} b_{ij} n_{ij}(t) \quad (1.25)$$

where $t_i(t)$ is the cumulative time elapsed in state **i** in the interval [0, t], and $n_{ij}(t)$ is the number of transitions occuring from state **i** to state **j**.

Under a matrix form : $W(t) = \mathbf{Y}\,[\mathbf{T}(t)]^T + \mathbf{B}\,[\mathbf{N}(t)]^T\,\mathbf{1}^T$(1.26)

where $\mathbf{Y}$ =[y_i] and $\mathbf{T}(t)$ =[$t_i(t)$] are vectors and $\mathbf{B}$ = [b_{ij}] and $\mathbf{N}(t)$ = [$n_{ij}(t)$] are matrix (with $b_{ii} = 0$ and $n_{ii}(t) = 0$).

1.4.1. Pointwise reward

a) Taking into account the discount rate

The expected pointwise reward in the interval [0, t] [EW(t;γ)] corresponds to:

$EW(t,\gamma) = E[W(t,\gamma)]$

$$= \sum_i [y_i \int_o^t p_i(u) e^{-\gamma u} V_i(u)du + \sum_{j \neq i} b_{ij} \int_0^t p_i(u) \lambda_{ij} p_i(u) e^{-\gamma u} S_{ij}(u) du] \quad (1.27)$$

where $p_i(u)$ is the probability to be in state **i** at time **u**,

and $p_i(u)\ \lambda_{ij}$ is the density probability of a transition from **i** to **j** in the interval[u;u +].

$$EW(t,\gamma) = \sum_i \left\{ \left[\int_0^t p_i(u) e^{-\gamma u} du \right] \left[y_i + \sum_{j \neq i} b_{ij} \lambda_{ij} \right] \right\} \quad (1.28)$$

In a matrix fom: $EW(t,\gamma) = \left[\int_0^t P(u) \exp(\gamma I u) du \right] (\mathbf{Y} + \mathbf{B} \square \Lambda \mathbf{1}^T)^T$ (1.29)

The symbol ❑ corresponds to the matrix operation "product of congruence" defined as:
$\mathbf{Z} = \mathbf{X} \square \mathbf{Y} \Leftrightarrow [z_{ij}] = [x_{ij} \cdot y_{ij}]$.

From relation (1.3) we find:

$$EW(t,\gamma) = \left\{ \mathbf{P}_0 \int_0^t \exp[(\Lambda - \gamma \mathbf{I})u] du \right\} (\mathbf{Y} + \mathbf{B} \square \Lambda \mathbf{1}^T)^T \quad (1.30)$$

EW(t,γ) is thus obtained by multiplying the integration of the exponential function of matrix:

$$\Lambda^* = \Lambda - \gamma\mathbf{I} \quad (1.31)$$

by the vector R of the **reward rates** associated to both state occupency and state transition[3]:

$$\mathbf{R} = \mathbf{Y} + \mathbf{B} \square \Lambda\ \mathbf{1}^T \quad (1.32)$$

hence :

$$EW(t,\gamma) = \left\{ \mathbf{P}_0 \int_0^t \exp(\Lambda^* u)\,du \right\} \mathbf{R}^T = \mathbf{P}_0\ {}^*(t)\ \mathbf{R}^T = \mathbf{L}^*(t)\ \mathbf{R}^T \quad (1.33)$$

with : $\ {}^*(t) = \int_0^t \exp(\Lambda^* u)\,du$

$$\mathbf{L}^*(t) = \mathbf{P}_0\ {}^*(t) = \mathbf{P}_0 \int_0^t \exp(\Lambda^* u)\,du$$

$$\mathbf{R} = (\mathbf{Y} + \mathbf{B} \square \Lambda\ \mathbf{1}^T)$$

b) No discount rate is considered

If the discount rate is equal to 0, we thus have:

$$EW(t) = \left\{ \mathbf{P}_0 \int_0^t \exp(\Lambda u)\,du \right\} \mathbf{R}^T = \mathbf{P}_0\ (t)\mathbf{R}^T = \mathbf{L}(t)\ \mathbf{R}^T \quad (1.34)$$

with : $\ (t) = \int_0^t \exp(\Lambda u)\,du$

$$\mathbf{L}(t) = \mathbf{P}_0\ (t) = \mathbf{P}_0 \int_0^t \exp(\Lambda u)\,du$$

$$\mathbf{R} = (\mathbf{Y} + \mathbf{B} \square \Lambda\ \mathbf{1}^T)$$

In the latter case, we also have:

$$EW(t) = E[W(t)] = E[\mathbf{Y}\ \mathbf{T}(t) + \mathbf{B}\,\mathbf{N}(t)^T\ \mathbf{1}^T] = \mathbf{Y}\,E[\mathbf{T}(t)] + \mathbf{B}\,E[\mathbf{N}(t)^T]\ \mathbf{1}^T \quad (1.35)$$

This can be applied both to an ergodic Markov chain and to a state-absorbing Markov chain.

---

3. By analogy with relation (1.4), **R** can be considered as a "super" summation vector for the calculation of additional measures.

1.4.2. Asymptotic reward

a) *Taking into consideration the discount rate*

The **asymptotic value of the expected reward, [EW(γ)]** is defined as:

$$EW(\gamma) = \lim_{t\to} EW(t,\gamma) = \lim_{t\to} E[W(t,\gamma)] \quad (1.36)$$

From relation (1.33); one can write:

$$EW(\gamma) = \left\{ \mathbf{P}_0 \int_0^{\infty} \exp(\Lambda^* u)\,du \right\} \mathbf{R}^T$$

$$EW(\gamma) = \lim_{s\to 0} \{ P_0 \; (s\,\mathbf{I} - \Lambda^*)^{-1} \} \, \mathbf{R}^T$$

$$EW(\gamma) = - \mathbf{P}_0 \, (\Lambda^*)^{-1} \; \mathbf{R}^T = \mathbf{P}_0 \, (\gamma\,\mathbf{I} \; - \; \Lambda)^{-1} \; \mathbf{R}^T (1.37)$$

b) *No discount rate is considered*

When there is no depreciation of the cumulated reward, the **asymptotic value of the expected reward** tends towards a finite constant value, only in the case of a state-absorbing Markov chain whose the associated yield rate is equal to zero. This value is not bounded in the case of an ergodic Markov chain; it is however possible to evaluate the reward variations rate.

b.1) Case of a Markov chain with absorbing states

We consider **Y** as the subset of absorbing states. In this particular case, it is interesting to evaluate the **asymptotic value of expected total reward, [EW$_X$]** in the **X** subset before absorbtion by **Y** with respect to P_{0X}. This measure is calculated from the partition that corresponds to subset **X** ; from (1.29) we thus have:

$$EW_X = \int_0^{\infty} \mathbf{P}_X(u)\,du\,(\mathbf{R}_X)^T$$

where $\mathbf{R}_X$ is a subvector from $\mathbf{R}$ defined by (1.32) that corresponds to the states of X : $\mathbf{R} = [\ \mathbf{R}_X \;\; \mathbf{0}_Y\ ]$

From (1.14) and (1.15) we thus obtain :

$$EW_X = - \mathbf{P}_{0X} \;\; (\Lambda_{XX})^{-1} \, \mathbf{R}_X{}^T (1.38)$$

Note that relation (1.38) is only a particular case of relation (1.37).

b.2) Case of an ergodic Markov chain

For an ergodic Markov chain, one can show that the expected value of the reward varies asymptotically in linear way when time tends towards infinity. The **reward variation rate [r]** corresponds to the slope of this variation.

$$r = \lim_{t \to} \frac{EW(t)}{t} = \lim_{t \to} \frac{E[W(t)]}{t} \quad (1.39)$$

Considering (1.26), relation (1.30) can be writen as:

$$r = \lim_{t \to} \left[\frac{1}{t} \int_0^t \mathbf{P}(u)du \right] \mathbf{R}^T$$

hence: $$r = \mathbf{P}\ \mathbf{R}^T \quad (1.40)$$

2. On the dependability measures in SURF-2

The dependability measures available in SURF-2 are listed in the table of Fig. 1. What follow in this paragraph, are definitions of the mathematical expressions characterizing these basic measures in relation with the basic measures given in paragraph 1.

Pointwise Measures	Asymptotic Measures	Mean Values
Reliability		MTFF
Safety		MTTF
Maintenability		MUT
Availability	Availability	MDT
Unavailability	Unavailability	MTBF
	Discounted Reward	AC[a]
Reward	Reward	Reward

a.Availability before Catastrophic failure.

Fig. 1: Dependability measures selection

In the table of Figure 1; the reward and discounted reward measures characterize the absence or the presence of a discount rate respectively.

To allow the measures to be computed, the user must associate a semantic to the subsets of states defined in paragraph 1. This definition relies on the notion of **class**. The user must define the **failed-states class** (that provide the improper service: I). The class of states that provide a proper service (P) is thus the complementary state class. Fig. 2 shows this state partition of the model.

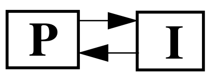

Fig. 2: Definiton of classes

2.1. Pointwise measures

2.1.1. Reliability

The **reliability [R(t)]** corresponds to the probability of a continuous presence in the subset of states **X**; the reliability is expressed from relation (1.6) where the subset **X** corresponds to the **class of states that provide a proper service P**.

$$\mathbf{R}(t) = \mathbf{P}_{0P} \exp(\Lambda_{PP}\ t)\ (\mathbf{1}_P)^T \quad (2.1)$$

Note : $\mathbf{P}_{0P}$ = vector of initial probabilities associated to the states that provide a proper service.

2.1.2. Safety

The **safety [S(t)]** is expressed in a similar way as the reliability; the difference is mainly semantic and is related to the definition of the failed state class defined by the user.

$$S(t) = \mathbf{P}_{0P} \exp(\Lambda_{PP}\ t)\ (\mathbf{1}_P)^T \quad (2.2)$$

2.1.3. Maintainability

The **maintainability [M(t)]** corresponds to the probability of presence in the class **P** when the system was initially down (Improper class). This measure can be obtained by:

- calculating the probability of the continuous presence in class **I**,
- taking the complement to 1.

$$M(t) = 1 - [\ \mathbf{P}_{0I} \exp(\Lambda_{II}\ t)\ (\mathbf{1}_I)^T] \quad (2.3)$$

Note: $\mathbf{P}_{0I}$ = vector of initial probabilities associated to the states that provide an improper service.

2.1.4. Availability

The **availability [A(t)]** is expressed from relation (1.4):

$$A(t) = \mathbf{P}_0 \exp(\Lambda\ t)\ \mathbf{U}^T \quad (2.4)$$

U is a summation vector for which the **|P| first** elements are equal to 1, the other being equal to 0.

2.1.5. Unavailability

The **unavailability [UA(t)]** is simply expressed as:

$$UA(t) = 1 - A(t) = \mathbf{P}_0 \exp(\Lambda\ t)\ \mathbf{Z}^T \quad (2.5)$$

Z is a summation vector whose **|P| first** elements are equal to 0, the other being equal to 1.

2.1.6. Reward and discounted reward

The expected total reward in the [0, t] interval [EW(t)] is expressed from relation (1.33):

$$\mathrm{EW}(t,\gamma) = \mathbf{P}_0 \ {}^*(t)\ \mathbf{R}^T \quad (2.6)$$

with : $\;{}^*(t) = \int_0^t \exp(\Lambda^* u)du = \int_0^t \exp[(\Lambda - \gamma I)u]du$

$$\mathbf{R} = (\mathbf{Y} + \mathbf{B} \ \square \ \Lambda \ \mathbf{1}^T) : r_i = y_i + \sum_{j=1}^{j \neq i, |E|} b_{ij}\lambda_{ij}$$

2.2. Asymptotic measures

2.2.1. Availability

The **asymptotic availability [A]** is derived form expression (1.10):

$$A = \mathbf{0}_{|E|} \ (\Lambda_{|E|})^{-1} \ \mathbf{U}^T \quad (2.7)$$

where $\mathbf{0}_{|E|}$ is the modified zero vector for which only element $|E|$ (the last one) is equal to 1.

$\Lambda_{|E|}$ is the modified transition rate matrix replacing column $|E|$ by vector $\mathbf{1}^T$.

2.2.2. Unavailability

The **asymptotic unavailability [UA]** can be simply expressed as:

$$UA = \mathbf{0}_{|E|} \ (\Lambda_{|E|})^{-1} \ \mathbf{U}'^T \quad (2.8)$$

where $\mathbf{U}' = [\ \mathbf{0}_X \ \ \mathbf{1}_Y\ ]$.

One can also note that: $UA = 1 - A = \mathbf{1} - \mathbf{0}_{|E|} \ (\Lambda_{|E|})^{-1} \ \mathbf{U}^T$

2.2.3. Discounted reward

From relation (1.37), the **asymptotic value of the expected discounted reward [EW(γ)]** can be written as:

$$\mathrm{EW}(\gamma) = -\mathbf{P}_0 \ (\Lambda^*)^{-1} \ \mathbf{R}^T = P_0 \ (\gamma \mathbf{I} - \Lambda)^{-1} \ \mathbf{R}^T \quad (2.9)$$

2.2.4. Reward rate

The **rate of the expected reward (r) for an ergodic Markov chain with an asymptotic behavior** is calculated from (1.40):

$$r = \mathbf{P}\ \mathbf{R}^T \quad (2.10)$$

with $\mathbf{P} = \mathbf{0}_{|E|}\ (\Lambda_{|E|})^{-1}$

$$\mathbf{R} = (\mathbf{Y} + \mathbf{B} \square\ \Lambda\ \mathbf{1}^T) : r_i = y_i + \sum_{j = 1,\ j \neq i}^{|E|} \lambda_{ij}$$

2.3. Mean Time values

The various mean sojourn times are calculated from expression (1.16):

$$\mathbf{M}\bullet\bullet\bullet = -\ \mathbf{P}_{X0}\ (\Lambda_{XX})^{-1}\ (\mathbf{1}_X)^T \quad (2.11)$$

The main difference lies essentially on the choice of vector $\mathbf{P}_{X0}$.

2.3.1. MTFF

Concerning the **Mean Time to First Failure**, $\mathbf{P}_{X0} = \mathbf{P}_{P0} = [1\ 0\ ...\ 0]$; that is the vector used to calculate the reliability:

$$\text{MTFF} = -\ \mathbf{P}_{P0}\ (\Lambda_{PP})^{-1}\ (\mathbf{1}_P)^T \quad (2.12)$$

2.3.2. MTTF

In the case of an ergodic Markov chain; this measure corresponds to the **mean time elapsed in the proper class P (Fig. 2) whereas the system is randomly reset in an asymptotic behavior [Mean Time To Failure]**.

This measure can be obtained from calculating:

- the vector of probabilities of the asymptotic states in the proper class **P**: P_P, from relation (1.9) with $\mathbf{P} = [\mathbf{P}_P\ \ \mathbf{P}_I]$,
- the vector of conditional probabilities to be in the proper class **P**, in the asymptotic behavior: $\mathbf{P}_{CP} = \mathbf{P}_P\ /\ [\mathbf{P}_P\ (\mathbf{1}_P)^T]$,
- the mean sojourn time in proper class **P** before absorbtion by the improper class **I**, using the vector $\mathbf{P}_{CP}$ as the vector of initial probabilities P_{0X} in (2.11).

$$\text{MTTF} = -\ \frac{\mathbf{P}_P}{\mathbf{P}_P \mathbf{1}_P}\ (\Lambda_{PP})^{-1}\ (\mathbf{1}_P)^T \quad (2.13)$$

2.3.3. MUT

In the case of an ergodic Markov chain, this measure corresponds to **the mean time elapsed in the proper class P (Fig. 2) whereas the system is reset after a transition from class I to class P, in an asymptotic behavior [Mean Up Time]**.

This measure can be obtained by calculating:

- the vector of probabilities of the asymptotic states in the improper class **I** : $\mathbf{P}_I$, from relation (1.9) with $\mathbf{P} = [\mathbf{P}_P \ \mathbf{P}_I]$,
- the vector of probabilities of transitions from class **I** to class **P** in the asymptotic behavior: $\mathbf{P}_{TP} = (\mathbf{P}_I \ \Lambda_{IP}) \ / \ [\mathbf{P}_I \ \Lambda_{IP} \ (\mathbf{1}_P)^T]$
- the mean sojourn time in proper class P before absorbtion by the improper class **I**, using the vector $\mathbf{P}_{TP}$ as the vector of initial probabilities $\mathbf{P}_{0X}$ in (2.11).

From

(i) relation 1.7 [$\mathbf{0} = \mathbf{P}\,\Lambda$]

(ii) the induced partition by **P** and **I**:

$$\mathbf{P}_P \, \Lambda_{PP} + \mathbf{P}_I \, \Lambda_{IP} = 0 \quad (2.14)$$

$$\mathbf{P}_P \, \Lambda_{PI} + \mathbf{P}_I \, \Lambda_{II} = 0 \quad (2.15)$$

(iii) the fact that the lines of Λ sum up to zero, in particular knowing that:

$$\Lambda_{PP} \, (\mathbf{1}_P)^T + \Lambda_{PI} \, (\mathbf{1}_I)^T = 0 \quad (2.16)$$

we obtain:

$$\mathrm{MUT} = \frac{\mathbf{P}_P(\mathbf{1}_P)^T}{\mathbf{P}_P\Lambda_{PI}(\mathbf{1}_I)^T} \quad (2.17)$$

2.3.4. MDT

There is a duality between this measure **[Mean Down Time]** and the MUT (permutation of **P** and **I** classes).

$$\text{MDT} = \frac{\mathbf{P}_I(\mathbf{1}_I)^T}{\mathbf{P}_I\Lambda_{IP}(\mathbf{1}_P)^T} \quad (2.18)$$

2.3.5. MTBF

This measure characterizes the mean cycle time between two failures **[Mean Time Between Failure]** that is:

$$\text{MTBF} = \text{MUT} + \text{MDT} \quad (2.19)$$

and thus, from relations (2.17) and (2.18), and the fact that $\mathbf{P}(\mathbf{1})^T = \mathbf{P}_P(\mathbf{1}_P)^T + \mathbf{P}_I(\mathbf{1}_I)^T = 1$:

$$\text{MTBF} = \frac{1}{\mathbf{P}_P\Lambda_{PI}(\mathbf{1}_I)^T} = \frac{1}{\mathbf{P}_P\Lambda_{IP}(\mathbf{1}_P)^T} \quad (2.20)$$

From the asymptotic availability definition (paragraph 1.1) we also have:

$$A = \frac{\text{MUT}}{\text{MTBF}} \quad (2.21)$$

2.3.6. Mean reward

The expected reward (with a null discount rate) cumulated in subset **P**, before absobtion by **I**, **[EWp]**; is calculated from (1.38) as:

$$EW_P = -\mathbf{P}_{P0}\ (\Lambda_{PP})^{-1}\ \mathbf{R}_P^T \quad (2.22)$$

where $\mathbf{R}_P$ is the fraction of vector $\mathbf{R}$ corresponding to the states of P: $\mathbf{R} = [\ \mathbf{R}_P \quad \mathbf{0}_I\ ]$

2.3.7. Examples of specific measures

Here are some examples derived from the use of EW_P. One can note, in particular, that from (2.12) and (2.22):

$$\text{MTFF} = EW_P \quad \text{for } \mathbf{Y}_P = \mathbf{1}_P \text{ and } \mathbf{B}_P = \mathbf{0}_P \quad (2.23)$$

Other interesting measures can be obtained using vector $\mathbf{Y}$ as a parametrable summation vector. This is particularly applicable in the case of safety system, where the states of the graph can be partionned into three different classes (Fig.3):

- class A : Service Accomplishment,
- class B : Benign failure,
- class C : Catastrophic failure.

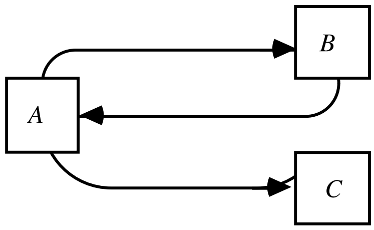

Fig. 3 : Generic graph for a safety system.

The Catastrophic class **C** corresponds to the Improper class when the safety measure S(t) is evaluated. It can easily be verified that expression (2.12) enables to calculate the **mean time before occurrence of a catastrophic failure [Mean Safe Time : MST]**. Moreover, relation (2.22) enables to calculate:

- using $\mathbf{Y}_P = [\ \mathbf{1}_A\ \mathbf{0}_B\ ]$ and $\mathbf{B}_P = \mathbf{0}_P$, the **mean accomplishment time before occurrence of a catastrophic failure [Mean Accomplishment Time : MAT]**,
- using $\mathbf{Y}_P = [\ \mathbf{0}_A\ \mathbf{1}_B\ ]$) and $\mathbf{B}_P = \mathbf{0}_P$, the **mean benign failure time before occurrence of a catastrophic failure [Mean Benign failure Time : MBT]**;

respectively defined as:

$$\text{MAT} = -\,\mathbf{P}_{P0}\ (\Lambda_{PP})^{-1}\ (\mathbf{1}_A\ \mathbf{0}_B)^T \quad (2.24)$$

$$\text{MBT} = -\,\mathbf{P}_{P0}\ (\Lambda_{PP})^{-1}\ (\mathbf{0}_A\ \mathbf{1}_B)^T \quad (2.25)$$

In particular, this allows to deduce the **mean availability before occurrence of a catastrophic failure [Availability before Catastrophic failure : AC] as well as the mean unavailability before occurrence of a catatrophic failure [UnAvailability before Catastrophic failure : UAC]** :

$$\text{AC} = \frac{\text{MAT}}{\text{MST}} = \frac{\text{MAT}}{\text{MAT} + \text{MBT}} \quad (2.26)$$

$$\text{UAC} = 1 - \text{UAC} = \frac{\text{MBT}}{\text{MST}} = \frac{\text{MBT}}{\text{MBT} + \text{MAT}} \quad (2.27)$$