

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire de Master

Option : Modélisation Mathématique et Evaluation des Performances des Réseaux

Evaluation des Performances d'un Modèle de Réseau de Petri Stochastique

Présenté par : SADI Lakhdar

Soutenu devant les jurys composé de :

Dr Djerroud Lamia	MCA	Présidente	à l'UAMB - Bejaia.
Dr Bernine Nassima	MCB	Encadreur	à l'UAMB - Bejaia
Dr Djabri Rabah	MCB	Examinateur	à l'UAMB - Bejaia.
Dr Benouaret Zina	MCB	Examinatrice	à l'UAMB - Bejaia.

Année Universitaire 2021 – 2022

Remerciement

Initialement, ce mémoire n'aurait pas été réalisé sans la bénédiction du Bon Dieu qui m'a permis de m'instruire et qui a récompensé mes prières.

Je souhaite avant tout remercier mon directeur de mémoire madame Nassima Bernine pour le temps qu'elle a consacré à m'apporter les outils méthodologiques indispensables à la conduite de cette recherche. Son exigence m'a grandement stimulé.

Je tiens également à remercier les membres du jury composés de Docteur Djerroud Lamia pour l'honneur qu'elle me fait en présidant le Jury de ma soutenance,

J'exprime toute ma gratitude à Docteur Djabri Rabah et Docteur Benouaret Zina d'avoir accepté l'évaluation de mon travail.

Je remercie également toute l'équipe pédagogique de l'université de Bejaia et les responsables de ma formation.

Dédicaces

Je dédie ce travail à tous ceux qui m'ont soutenu tout au long de mon cursus universitaire :

A mes parents

A mes frères

A mes sœurs

A mes amis et tous ceux qui lisent ce mémoire.

Table des matières

<i>Introduction générale</i>	1
1 Réseaux de Petri	3
1.1 Introduction	3
1.2 Définition d'un Réseau de Petri	3
1.2.1 Structure d'un Réseau de Petri	4
1.2.2 Réseau de Petri marqué	4
1.2.3 Notations matricielles	5
1.3 Interprétation d'un RdP	6
1.3.1 Dynamique des Réseaux de Petri	6
1.3.2 Evolution d'un RdP	7
1.3.3 Sensibilisation et franchissement des transitions	7
1.3.4 Séquence de franchissement	7
1.3.5 Ensemble des marquages accessibles	8
1.3.6 Graphe des marquages accessibles	8
1.4 Propriétés des réseaux de Petri	9
1.5 Réseaux de Petri particuliers	9
1.6 Extensions des RdP	15
1.6.1 Réseaux de Petri à arcs inhibiteurs	15
1.6.2 Réseaux de Petri colorés	16
1.6.3 RdP temporisés et RdP temporels	17
1.7 Réseaux de Petri stochastiques	18
1.7.1 Modèles RdPS markoviens et non markoviens	18
1.7.2 Types de réseaux de Petri stochastiques	21
1.7.3 Réseaux de Petri Stochastiques Généralisés	22

1.7.4	Sémantique des RdPSG :	24
1.8	Conclusion	25
2	Etat de l'art sur l'évaluation de performance	26
2.1	Introduction	26
2.2	Concept d'évaluation de performance	26
2.3	Types d'évaluation des performances	27
2.4	Evaluation des indices de performances	27
2.4.1	Applications à l'évaluation de performances	28
2.5	Techniques de l'évaluation de performances	30
2.5.1	Les mesures directes	30
2.5.2	Méthodes analytiques	31
2.5.3	Simulation	37
2.6	Conclusion	42
3	Analyse d'un modèle de réseau de Petri stochastiques	43
3.1	Introduction	43
3.2	Modélisation	43
3.3	Analyse de modèle	45
3.3.1	Résultats de la simulation	45
3.4	Conclusion	47
	Conclusion générale	48

Liste des figures

1.1	Exemple d'un réseau de Petri.	5
1.2	Exemple de graphe des marquages accessibles d'un RdP.	8
1.3	Graphe d'état.	10
1.4	Graph d'événement	10
1.5	Exemple d'un conflit structurel	11
1.6	Exemple d'un conflit effectif.	12
1.7	Graphe RdP sans conflit	12
1.8	RdP avec choix libre.	13
1.9	RdP simple.	13
1.10	RdP pur.	14
1.11	RdP généralisé.	14
1.12	RdP à capacité.	15
1.13	Représentation d'un arc inhibiteur.	16
1.14	Distribution exponentielle des durées de tir de transition.	20
1.15	Distribution uniforme bornée des durées de tir de transition.	21
1.16	Distribution triangulaire des durées de tir de transition.	21
2.1	Techniques de l'évaluation de performances.	30
2.2	Interface utilisateur graphique de GreatSPN.	39
2.3	Interface utilisateur graphique de TimeNET 4.0.	40
2.4	Logiciel GRIF	41
3.1	Le réseau de petri stochastiques.	44
3.2	Interface utilisateur graphique de GRIF	57
3.3	Barre d'icônes.	57

Liste des tableaux

3.1	Temps moyen de réponse.	46
3.2	Nombre moyen de jetons dans le système.	46
3.3	Fréquence moyenne de franchissement la transition T_1	47
3.4	Fréquence moyenne de franchissement la transition T_2	52
3.5	Fréquence moyenne de franchissement la transition T_3	53
3.6	Fréquence moyenne de franchissement la transition T_4	53
3.7	Fréquence moyenne de franchissement la transition T_5	54

INTRODUCTION GÉNÉRALE

L'évaluation des performances est l'un des sujets les plus étudié en recherche opérationnelle. L'intérêt pour ce sujet vient en partie du fait qu'il nous permet de choisir les meilleurs paramètres du système, pour apporter des améliorations à leurs caractéristiques. Il existe plusieurs approches d'évaluation de performance des systèmes, comme les approches analytiques (files et réseaux de files d'attente, réseaux de Petri ...), et la simulation (GreatSPN, TimeNET, GRIF ...).

Un réseau de Petri (RdP) est un outil graphique de modélisation et d'évaluation des performances de systèmes complexes. Sa puissance expressive permet d'étudier des systèmes composés de sous-systèmes, qui fonctionnent en parallèle, communiquent et partagent des ressources. Les réseaux de Petri ont été largement utilisés dans différents domaines.

Un réseau de Petri stochastique (RdPS) a été introduit pour résoudre certains problèmes d'évaluation des performances impliquant des phénomènes aléatoires. Par conséquent, la transition de RdP contient des temps de passage aléatoires.

L'évaluation des performances est un processus, qui comprend la détermination des caractéristiques du système obtenu, pour ces entrées/sorties dans le cadre de la réalisation des tâches qui leur sont assignées.

Notre travail a pour principal objectif, de modéliser et évaluer les performances d'un réseau de Petri stochastique.

Ce mémoire est organisé en trois chapitres et se termine par une conclusion générale .

Le premier chapitre, portera quelques généralités sur les réseaux de Petri et ces extentions.

Dans le deuxième chapitre, nous allons présenter quelques généralités sur les réseaux de

Petri stochastiques, et les méthodes de résolution des réseaux de Petri stochastiques.

Enfin, le troisième chapitre fera le point sur l'objectif fixé auparavant, nous abordons l'analyse et l'évaluation des performances de réseau de Petri stochastiques, à l'aide du simulateur GRIF qui sera présenté dans ce chapitre .

1.1 Introduction

Les systèmes dynamiques ne peuvent pas être décrits en prenant en compte que leurs états initiaux et finaux. En effet, on doit tenir compte de leur comportement permanent qui est une séquence d'états, pour qu'on puisse parler d'une description bien fondée. Parmi le grand nombre des techniques formelles qui ont, été proposées pour spécifier, analyser et vérifier ce genre de systèmes, les réseaux de Petri sont l'une des plus utilisés.

Historiquement, le réseau est présenté par **Carl Adam Petri** dans sa thèse "Communication avec des Automates" en **Allemagne à Bonn en 1962**. Ce travail a continué à développer par Anatol W. Holt, F [16, 20]. Commoner, M. Hack et leurs collègues dans le groupe de recherche de Massachusetts Institute Of Technology (MIT) dans des années 70s. En 1975, la première conférence de réseau de Petri, et des méthodes relationnels ont été organisés à MIT. En 1981, le premier livre du réseau de Petri a été publié en Anglais par J. Peterson. Aujourd'hui, suivre Petri-Net Newsletter, chaque année il y a des 600 aux 800 d'oeuvres sur les réseaux de Petri qui sont publiés.[16, 20]

1.2 Définition d'un Réseau de Petri

Un réseau de Petri (RdP) est un graphe biparti orienté, composé de deux types de noeuds : les places et les transitions. Graphiquement les places sont représentées par des cercles et les

transitions par des traits ou des rectangles. Les places et les transitions sont reliées par des arcs. Chaque place contient un nombre entier ($n \geq 0$) de jetons[22].

L'état du système modélisé par un RdP est représenté par le marquage du réseau qui est un vecteur, et qui donne la distribution des jetons dans les places du réseau.

1.2.1 Structure d'un Réseau de Petri

Une structure de RdP est donnée par un quadruplet [22] :

$$R = (P, T, Pre, Post). \quad (1.1)$$

Tels que :

- $P = \{ p_1, p_2, \dots, p_n \}$ est un ensemble fini de places,
- $T = \{ t_1, t_2, \dots, t_m \}$ est un ensemble fini de transitions,
- $Pre : P \times T \rightarrow N$ c'est l'application d'incidence avant (places précédentes),
- $Post : T \times P \rightarrow N$ c'est l'application d'incidence arrière (places suivantes),

On note par C la matrice d'incidence du réseau de Petri qui est définie par :

$$C = Post - Pre.$$

1.2.2 Réseau de Petri marqué

Un marquage M_j d'un RdP est une application, $M_j : P \rightarrow N$, qui associe à chaque place $p \in P$ du RdP un marquage $M_j(p)$ qui est le nombre de jetons dans cette place.

Le couple $N = (R; M_0)$ est dit RdP marqué; où :

- R est un réseau de Petri,
- M_0 est le marquage initial.

Exemple de réseau de Petri

La figure (1.1) représente un RdP marqué défini par ses places $P = \{p_1, p_2, p_3, p_4, p_5\}$ et transitions $T = \{t_1, t_2, t_3, t_4, t_5\}$, ainsi que les arcs qui les relient.

Le marquage M du réseau illustré dans la figure est défini par le vecteur :

$$M = (M(p_1), M(p_2), M(p_3), M(p_4), M(p_5)). \quad (1.2)$$

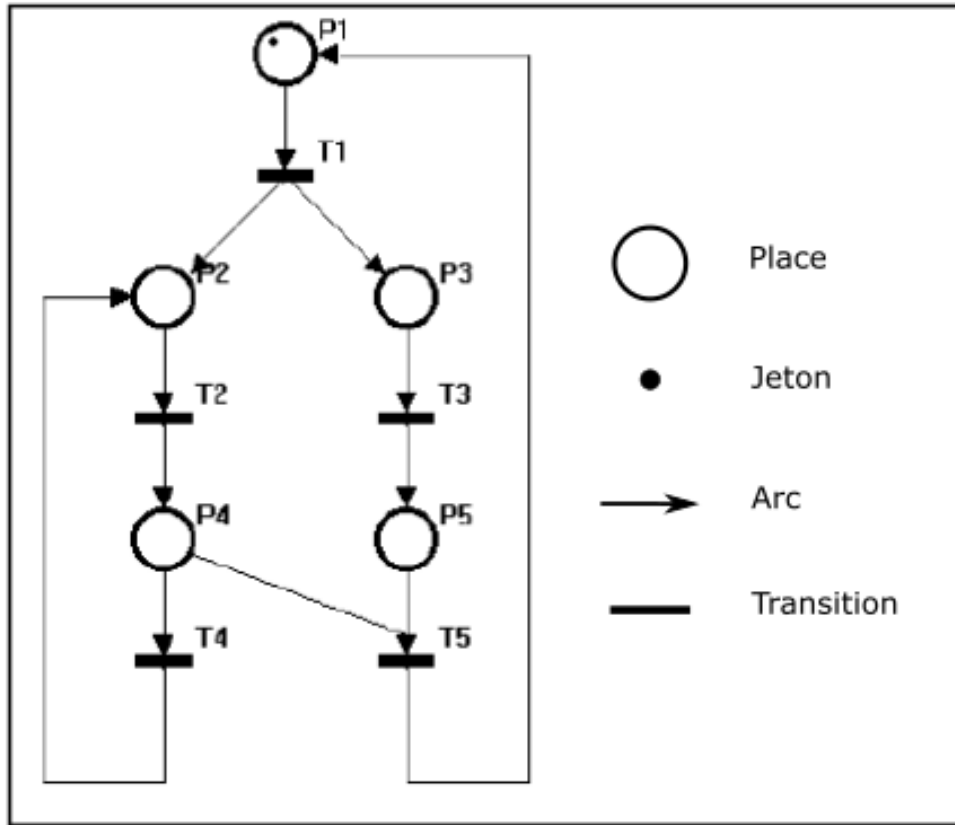


FIGURE 1.1 – Exemple d'un réseau de Petri.

1.2.3 Notations matricielles

Les fonctions Pre, Post et C, peuvent être présentées par des matrices dont le nombre de lignes est égal au nombre de places, et le nombre de colonnes est égal au nombre de transitions.

De l'exemple de la Figure 1.1, on peut illustré les matrices suivantes :

$$M_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$Pre = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Post = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

1.3 Interprétation d'un RdP

1.3.1 Dynamique des Réseaux de Petri

La dynamique d'un RDP correspond à l'évaluation de son marquage au cours du temps (évolution de l'état du système). Il se traduit par un déplacement de marques, ce qui s'interprète comme la consommation/ production de ressources déclenchées par des événements, ou des actions. Déterminer l'évolution d'un RDP correspond, par exemple, à le simuler.

Franchissement d'une transition

Règle de franchissement : Le franchissement d'une transition d'un RDP s'effectue lorsque toutes les places en amont de cette transition contiennent au moins une marque, la transition est validée peut être franchie. Les marques sont alors enlevées à des places en amont et sont déposées dans les places en aval de la transition.

Temps de franchissement : Un temps est associé à chaque transition. Ce temps indique la durée pendant laquelle la marque doit attendre avant de franchir la transition. Pour être plus exact, il s'agit de la durée entre le moment où la transition est sensibilisée et le moment où la transition est franchie. Ce temps est une variable aléatoire qui suit une distribution exponentielle.

Le paramètre de la fonction densité de probabilité associée à la transition t_i est le taux de franchissement noté μ_i .

C'est à dire que la transition temporisée est franchie avec un temps aléatoire de moyenne d_i , avec $d_i = \frac{1}{\mu_i}$.

1.3.2 Evolution d'un RdP

L'évolution du marquage par franchissement des transitions dans un RdP, traduit l'évolution du système modélisé dans ces différents états, après l'occurrence de certains événements [10].

Proposition 1.1 Chaque place d'un réseau de Petri peut avoir deux états : elle est marquée ou non marquée.

1.3.3 Sensibilisation et franchissement des transitions

Définition 1.1 [10] Une transition t est dite sensibilisée (franchissable, tirable, activée, validée) pour un marquage M si et seulement si :

$$\forall p \in P : M(p) \geq Pre(p, t); \text{ et on note } M(t > . \quad (1.3)$$

Si t est franchissable pour le marquage M , le franchissement de t donne le nouveau marquage M' tel que :

$$\forall p \in P; M'(p) = M(p) - Pre(p, t) + Post(p, t); \text{ et on note } M(t > M'. \quad (1.4)$$

1.3.4 Séquence de franchissement

Définition 1.2 Une suite finie de transitions $s = \langle t_1, t_2, \dots, t_k \rangle$ est franchissable à partir du marquage M , s'il existe des marquages M_1, M_2, \dots, M_k , tels que [10, 9] :

$$M_1[t_2 > M_2[t_2 > M_3 > \dots > [t_{k-1} > M_k. \quad (1.5)$$

On dit que s est une séquence de franchissement allant de M_1 à M_k . On note $M_1[S > M_k$.

Définition 1.3 Un marquage M est dit accessible (ou atteignable) depuis le marquage initial M_0 s'il existe une séquence de franchissement s telle que : $M_0(S > M$ voir [29] .

1.3.5 Ensemble des marquages accessibles

Définition 1.4 L'ensemble des marquages accessibles, notée $A(M_0)$, est l'ensemble des marquages atteints, à partir du marquage initial M_0 par une séquence de franchissement[11] .

1.3.6 Graphe des marquages accessibles

Définition 1.5 [11] Le graphe des marquages noté $G(R;M_0)$ est un graphe orienté, qui a pour sommets l'ensemble des marquages accessibles $A(M_0)$. Un arc relie deux sommets M_i et M_j , si et seulement si il existe une transition $t \in T$ telle que : $M_i(t > M_j$. Chaque arc est étiqueté par le nom de la transition correspondante.

La figure (1.2) représente le graphe des marquages accessibles de l'exemple précédent (voir la figure(1.1))

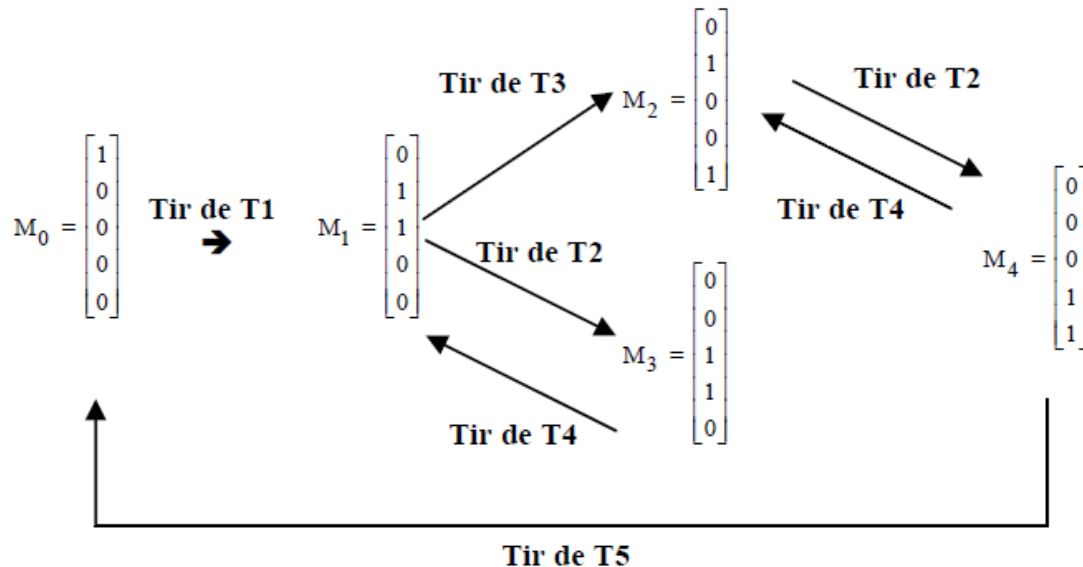


FIGURE 1.2 – Exemple de graphe des marquages accessibles d'un RdP.

1.4 Propriétés des réseaux de Petri

Ces propriétés dépendent à la fois du marquage initial M_0 et de la structure du réseau. Parmi ses propriétés nous citons [4] :

- **Réseau borné** : La bornitude d'un RdP exprime le fait que, le nombre d'états accessibles par le marquage initial M_0 est fini, autrement dit, le nombre de marquages accessibles est fini. Dans le cas contraire, où le RdP est non borné, le nombre d'états est infini, et ceci est dû au fait que certains paramètres de ce système sont non bornés [16].

- **Réseau sans blocage** : Un RdP est dit bloqué, si à une certaine étape son franchissement s'arrête et aucune transition n'est franchissable.

Un marquage M_0 d'un réseau (PN, M_0) est appelé marquage puits(mort), si aucune transition n'est franchissable à partir de M_0 . Un réseau (PN, M_0) est dit sans blocage, si tout marquage accessible depuis M_0 n'est pas un marquage puit. Un réseau RdP est dit sans blocage, s'il existe un marquage initial M_0 tel que (PN, M_0) soit sans blocage.

- **Vivacité** : On dit qu'une transition T_j est vivante pour un marquage initial M_0 si pour tout marquage accessible de M_0 , il existe une séquence de franchissement T^* qui contienne la transition T_j , à partir de M_i .

Un RdP est dit vivant pour un marquage initial M_0 , si toutes ses transitions sont vivantes.

- **Quasi-Vivant** : Lorsque toutes les transitions sont franchissables par une séquence de tirs à partir du marquage initial M_0 .

1.5 Réseaux de Petri particuliers

Graphe d'état

Un réseau de Petri non marqué, est un graphe d'état si et seulement si toute transition a exactement une seule place d'entrée, et une place de sortie.

Exemple : les transitions T_1, T_2, T_3, T_4 et T_5 possède une place d'entrée et une seule place de sortie. Comme il est détaillé dans la figure 1.3 :[24]

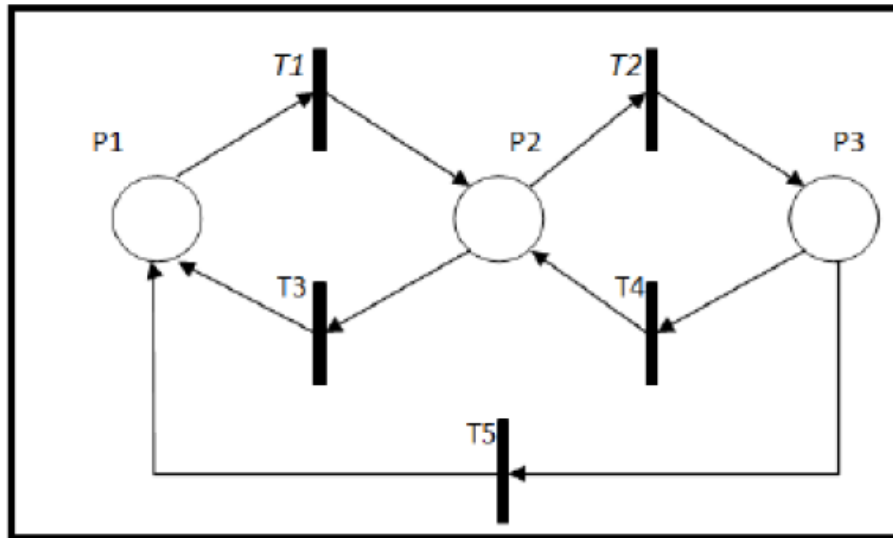


FIGURE 1.3 – Graphe d'état.

Graphe d'événement

Un réseau de Petri est un graphe d'événement, si et seulement si chaque place possède exactement une seule transition d'entrée et une seule transition de sortie, comme il est schématisé ci-dessous (Figure 1.4).un graphe d'évènements est parfois appelé graphe de transition[26].

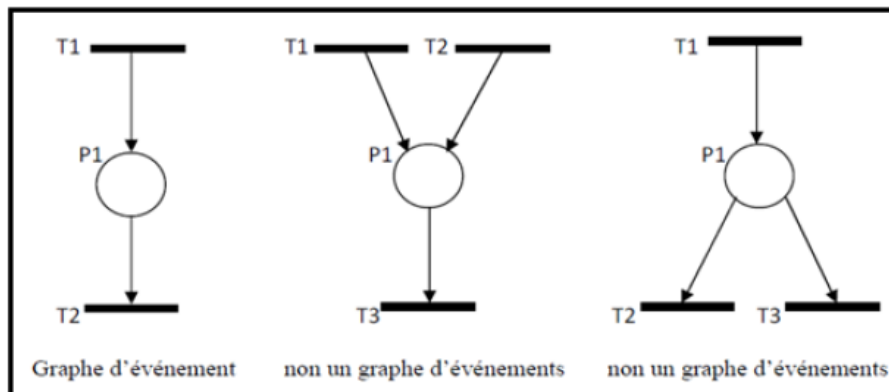


FIGURE 1.4 – Graph d'événement .

RdP conflit

Nous venons de voir que l'évolution d'un réseau de Petri est réalisée par le franchissement de transitions sensibilisées. Pour un marquage donné, plusieurs transitions peuvent être sensibilisées, mais la règle d'évolution impose qu'une seule transition puisse être franchie à la fois. Si les transitions sensibilisées n'ont pas de place en amont commune, le tir de l'une d'entre elles ne remet pas en cause celui des autres. On parle dans ce cas d'un RdP sans conflit (toute place a au plus une transition de sortie). Si les transitions sensibilisées ont une place amont en commun, le tir de l'une d'entre elles peut remettre en question celui des autres. Ces transitions sont en conflit qui conduit à un indéterminisme dans l'évolution d'un réseau lié à la compétition entre ces transitions[12]. Un conflit est l'existence d'une place qui a au moins deux transitions de sortie :

La notion de conflit modélise un choix de décision.

$$[p_i; \{t_1, t_2, \dots, t_n\}] \quad (1.6)$$

Il existe deux types de conflit :

Conflit structurel : Deux transitions t_1 et t_2 sont en conflit structurel, si et seulement si elles ont au moins une place d'entrée en commun[13].

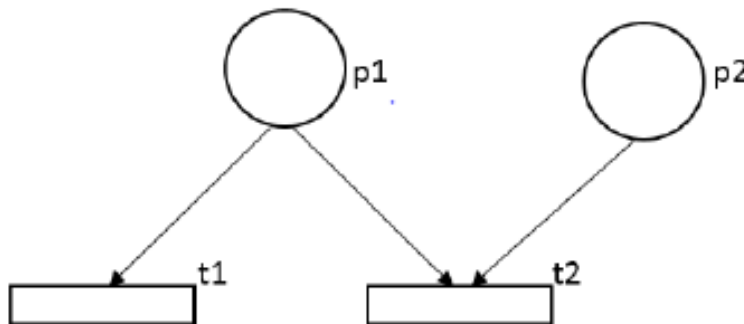


FIGURE 1.5 – Exemple d'un conflit structurel .

Conflit effectif : Deux transitions t_1 et t_2 sont en conflit effectif pour un marquage M , si et seulement si t_1 et t_2 sont en conflit structurel[13].

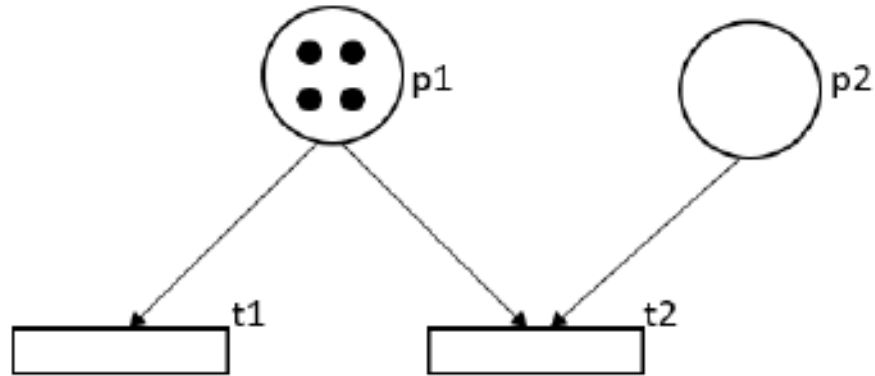


FIGURE 1.6 – Exemple d’un conflit effectif.

RdP sans conflit

Dire qu’un RdP est un réseau sans conflit, si et seulement si chaque place a au plus une transition de sortie. Un RdP avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties. Un conflit est noté : $[P_i, t_1, t_2, \dots, t_n]$; avec t_1, t_2, \dots, t_n étant les transitions de sortie de la place T_n voir [24]. Comme il est montré dans la figure 1.7.

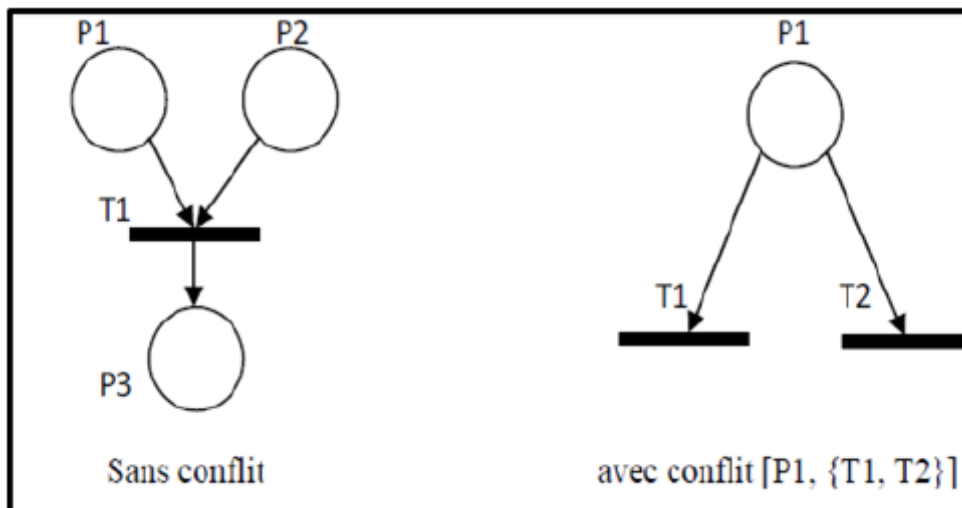


FIGURE 1.7 – Graphe RdP sans conflit .

RdP à choix libre

un RdP à choix libre, est un réseau dans lequel pour tout conflit $[P_i, T_1, T_2, \dots, T_n]$ aucun des transitions T_1, T_2, \dots, T_n ne possède une autre place d'entrée que P_i voir [26]. Comme il est exprimé dans la figure 1.8.

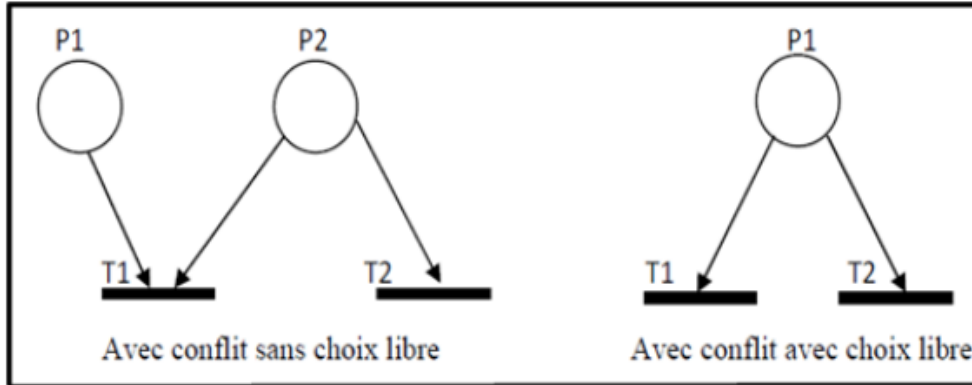


FIGURE 1.8 – RdP avec choix libre.

RdP simple

les réseaux de Petri simples sont des RdP ordinaires tels que chaque transition a au plus une place d'entre qui peut être reliée a d'autré transition (Toute transition appartient à un seul conflit au plus)[26]. Tel qu'il est présenté dans la figure 1.9.

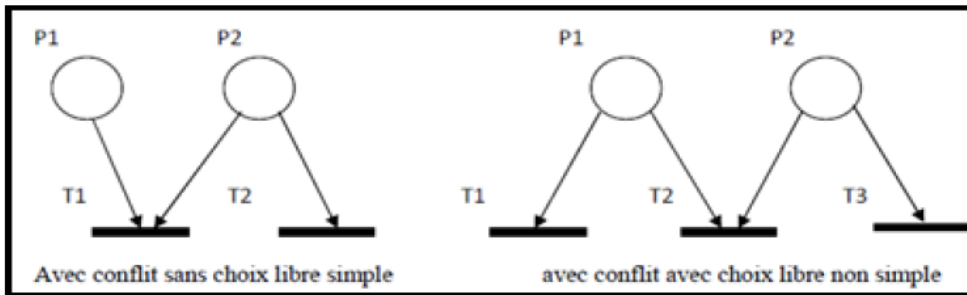


FIGURE 1.9 – RdP simple.

RdP pur

dans un RdP une transition est dite pure, si elle ne possède aucune place, qui est à la fois place d'entrée et place de sortie. Dans un RdP une transition est dite impure si elle possède une place qui est à la fois place d'entrée et place de sortie. Si les transitions du RdP sont impures, le RdP est impur voir la figure 1.8 .[26]

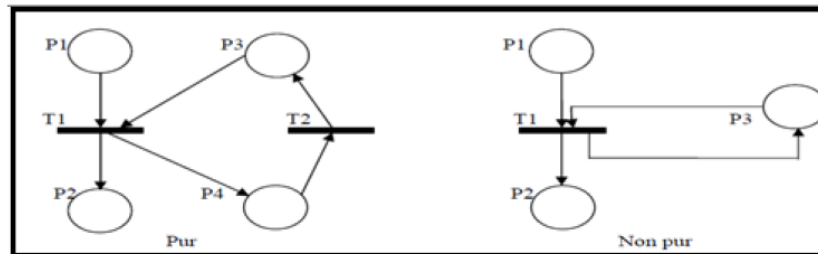


FIGURE 1.10 – RdP pur.

RdP généralisés

Un RdP généralisé est un RdP dans lequel des poids (nombres entiers strictement positifs) sont associés aux arcs. Si un arc (P_i, T_j) a un poids K : la transition T_j n'est franchie que si la place P_i possède au moins K jetons. Le franchissement consiste à retirer K jetons de la place P_i . Si un arc (T_j, P_i) a un poids K : le franchissement de la transition rajoute K jetons à la place P_i . Lorsque le poids n'est pas signalé, il est égal à un. Comme il est exprimé dans la figure suivante 1.9 .[24]



FIGURE 1.11 – RdP généralisé.

RdP a capacité

Un RdP à capacité est un RdP dans lequel des capacités (nombre entiers strictement positifs) sont associées aux places. Le franchissement d'une transition d'entrée d'une place P_i , dont la capacité est $\text{cap}(P_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i , qui est plus grand que $\text{cap}(P_i)$ [24].

La figure 1.10 montre que le franchissement de T_1 conduit à 3 jetons dans P_2 , d'où T_1 ne peut plus être franchie.

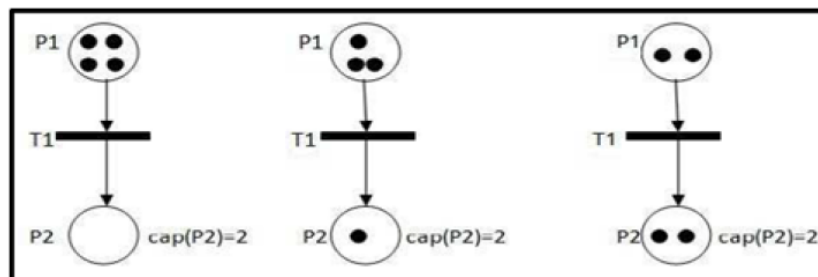


FIGURE 1.12 – RdP à capacité.

1.6 Extensions des RdP

Le concept classique de RdP a été largement développé par beaucoup d'auteurs depuis les années 1970, grâce à une intégration particulière du temps et des aspects aléatoires dans le modèle d'origine. Les paragraphes suivants sont consacrés à quelques extensions des réseaux de Petri.

1.6.1 Réseaux de Petri à arcs inhibiteurs

Un arc inhibiteur est un arc orienté qui part d'une place P pour aboutir à une transition t . Son extrémité est marquée par un petit cercle (voir figure 1.13).

L'arc inhibiteur entre la place P et la transition t , signifie que la transition t n'est validée que si la place P ne contient aucune marque. Le franchissement de t consiste à retirer une marque dans chaque place d'entrée de t à l'exception de P , et à ajouter une

marque dans chaque place de sortie de t . On utilise aussi les expressions "test à zéro" et "RdP étendus".

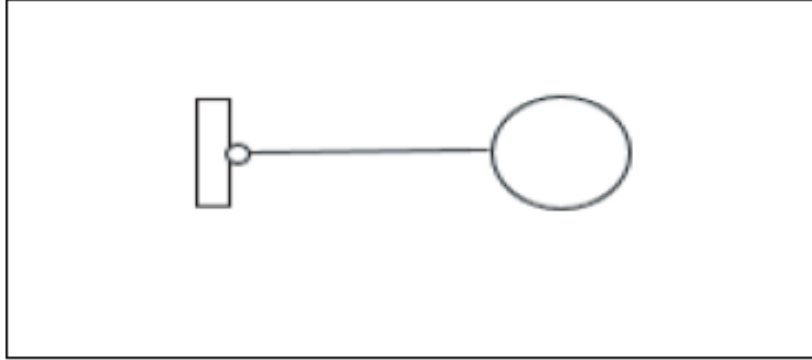


FIGURE 1.13 – Représentation d'un arc inhibiteur.

1.6.2 Réseaux de Petri colorés

Pour un réseau de Petri de base, on ne distingue pas les différents jetons. Dans un réseau de Petri coloré, on associe une valeur à chaque jeton. Pour plusieurs outils associés aux réseaux de Petri colorés, les valeurs des jetons sont typées, et peuvent être testées et/ou manipulées avec un langage fonctionnel [16].

Définition 1.6 Un RdP coloré est un 6-uplet, $R = (P; T; C_{oul}; C_{sec}; W; M_0)$; où [19] :

- P est un ensemble fini de places ;
- T est un ensemble fini de transitions ;
- C_{oul} est un ensemble fini de couleurs ;
- C_{sec} est la fonction qui à chaque place et à chaque transition associe un sous ensemble de C_{oul} :

$$C_{sec} : P \cup T \rightarrow P(C_{oul}); \quad (1.7)$$

- W est la fonction d'incidence $C = \text{Post} - \text{Pre}$:

$$W(p; t) : C_{sec}(t) \times C_{oul}(p) \rightarrow \mathbb{N}; \quad (1.8)$$

- M_0 est le marquage initial, pour chaque place et pour chaque couleur possible dans cette place. Il associe un nombre de jetons.

$$M_0(p) : C_{sec}(p) \rightarrow \mathbb{N}. \quad (1.9)$$

1.6.3 RdP temporisés et RdP temporels

Les réseaux de Petri temporisés introduisent la notion de temps dans le parcours du réseau, qui permettent de décrire un système à évènement discret, dont le fonctionnement dépend du temps.

La temporisation représente la durée minimale de franchissement, où le temps de séjour minimum d'un jeton dans la place. Il existe deux sortes de RdP temporisés : RdP T-temporisés et des RdP P-temporisés.

Réseau de Petri T-temporisé

Les réseaux de Petri T-temporisés fonctionnent de la manière suivante : lorsqu'un jeton franchit une transition, celui-ci est réservé pendant un certain temps. C'est seulement après ce temps d'attente, que le jeton franchira effectivement la transition.

Réseau de Petri P-temporisé

Une temporisation (valeur rationnelle positive) est associée à chaque place, qui représente la durée de séjour minimale d'une marque dans cette place. On notera par d_i la temporisation de place p_n [24].

Le RdP P-temporisé est défini par le couple $((PN; M_0); d)$ tel que :

- (PN, M_0) : Réseau marqué.
- $d : P \rightarrow Q^+$ est la fonction de temporisation ($d_i =$ temps que passent les jetons dans la place p_i).

L'état de disponibilité d'un jeton permet de spécifier la durée nécessaire pour effectuer une opération sur un produit donné.

Réseaux de Petri temporels

Pour la modélisation et l'analyse des systèmes de communication. Une transition dans ce RdP peut être franchie, seulement lorsqu'une durée de temps comprise dans

l'intervalle du temps associé à la transition, ou bien à la place s'est écoulée depuis l'instant de sa validation. La transition t_i doit rester sensibilisée durant au moins d_{imin} unités de temps, et au plus d_{imax} unités de temps avant d'être franchie. Par contre les jetons peuvent à tout moment être consommés par une autre transition[24].

1.7 Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques ont été introduits par Florin en 1978, pour résoudre certains problèmes d'évaluation liés à la fiabilité des systèmes informatiques. Pour ces problèmes impliquant des phénomènes aléatoires, les transformations des réseaux de Petri incluent des temps de passage aléatoires, distribués de façon exponentielle. Cette distribution exponentielle permet d'exploiter les propriétés mathématiques des processus de Markov[3].

Définition 2.1 Un réseau de Petri stochastique est le couple $(R; \Lambda)$ avec :

- $R = (P; T; A; M_0)$ est un réseau de Petri;
- Λ est une fonction qui à chaque transition t , on associe un taux de franchissement $\lambda_t = \Lambda_t$.

Dans les RdPS la durée de sensibilisation est une variable aléatoire θ , avec une distribution de probabilité, dans le cas de distribution exponentielle :

$$P_\theta = P[\theta \leq x] = 1 - e^{-x}. \quad (1.10)$$

La fonction $P_\theta(x)$ décrit la probabilité pour que la durée de sensibilisation soit inférieure ou égale à x .

Avec les RdPS, on pourra par exemple, calculer le temps de bon fonctionnement entre deux défaillances, le temps de réparation ou dans certains cas la durée opérationnelle d'une machine, les taux de production, l'évolution des stocks, etc.[6]

1.7.1 Modèles RdPS markoviens et non markoviens

Modèles RdPS markoviens : On s'intéresse dans cette partie, à la description et à l'exploitation d'une sous-classe de processus stochastiques à savoir les processus marko-

viens, aussi appelés processus sans mémoire, c'est-à-dire que l'état futur dépend uniquement de l'état présent, et non du passé. Toutes les transitions d'un RdPS markovien sont temporisées avec un temps aléatoire distribué selon une loi exponentielle. La loi exponentielle modélise la durée de vie d'un phénomène sans mémoire (ou sans vieillissement, ou sans usure). La probabilité P que le phénomène dure au moins $s + t$ sachant qu'il a déjà duré t est identique à la probabilité de durer s à partir de son initialisation. L'absence de mémoire représente une propriété importante de la distribution exponentielle (figure 2.1) que l'on peut traduire mathématiquement par l'équation :

$$\forall s, t \geq 0 \mathbf{P}(t' > s + t | t' > t) = P(t' > s) \quad (1.11)$$

où t' modélise la durée de vie d'un système et $P(t' > s + t | t' > t)$ est la probabilité conditionnelle de $(t' > s + t)$ sachant que $(t' > t)$.

La distribution d'une variable aléatoire exponentielle de paramètre $\lambda > 0$ est donnée par :

$$f(t) = \begin{cases} \lambda e^{-\lambda t}, & \text{si } t \geq 0 \\ 0, & \text{si } t < 0 \end{cases} \quad (1.12)$$

$f(t)$ est la fonction de densité de probabilité caractérisant la durée de franchissement des transitions, ainsi sa fonction de répartition $F(t)$ si $t > 0$ est calculée comme suit :

$$F(t) = \int_{-\infty}^t f(x) dx = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t} \quad (1.13)$$

Un RdPS markovien est caractérisé par une politique de choix par compétition, une absence de mémoire et une politique de service infini. Pour plus de détails sur la loi exponentielle ainsi que ses propriétés [25], voir la définition 2.1.

Modèles RdPS non markoviens

Les RdPS non markoviens permettent d'imiter le comportement d'un système en traitant des modèles généralement plus proches de la réalité. Contrairement aux modèles markoviens, une prise en compte de la mémoire peut être considérée. Pour un RdPS non markovien toutes les transitions sont temporisées avec une durée aléatoire distribuée

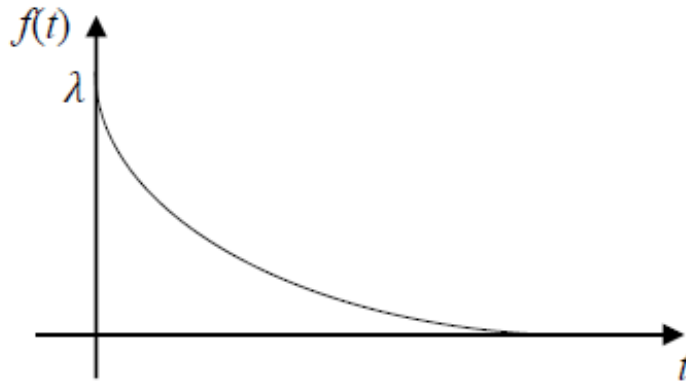


FIGURE 1.14 – Distribution exponentielle des durées de tir de transition.

selon une loi quelconque. Dans la suite, toute loi à support borné et dont la densité de probabilité est connue peut être utilisée. En particulier, les lois uniformes bornées, triangulaires symétriques et exponentielles tronquées sont considérées. La définition analytique de leur densité de probabilité est supposée être connue. Soit une variable aléatoire Y susceptible de prendre toutes les valeurs d'un intervalle fini $[a, b]$, sans privilégier aucune région de $[a, b]$ (on parle d'événements équiprobables). La probabilité que Y prenne une valeur appartenant à l'intervalle $[m, n] \subset [a, b]$ est proportionnelle à la longueur de $[m, n]$, d'où :

$$\mathbf{P}(m \leq Y \leq n) = \frac{m - n}{b - a} \quad (1.14)$$

où

$$f(y) = \begin{cases} \frac{1}{b-a}, & \text{si } a \leq y \leq b \\ 0, & \text{sinon} \end{cases} \quad (1.15)$$

La fonction $f(y)$ représente la densité de probabilité qui définit le comportement aléatoire (stochastique) de la variable aléatoire Y et permet ainsi de caractériser sa loi de probabilité (distribution). Dans ce travail, cette variable aléatoire représente les durées de franchissement des transitions d'une loi uniforme bornée (figure 2.1).

Une loi triangulaire (figure 2.2) est définie de la même manière avec l'équation suivante :

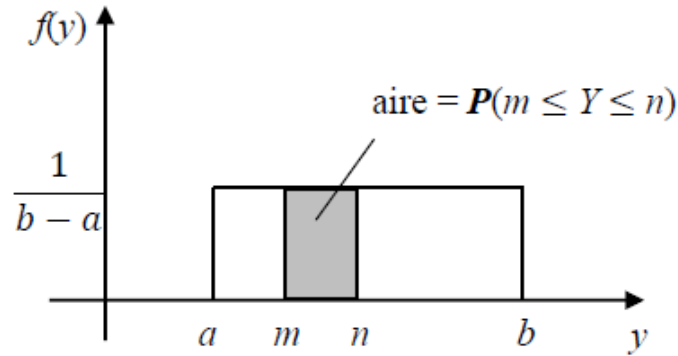


FIGURE 1.15 – Distribution uniforme bornée des durées de tir de transition.

$$f(y) = \begin{cases} \frac{2(y-a)}{(m-a)(b-a)}, & \text{si } a \leq y \leq m \\ \frac{2(b-y)}{(b-m)(b-a)}, & \text{si } m \leq y \leq b \\ 0, & \text{sinon} \end{cases} \quad (1.16)$$

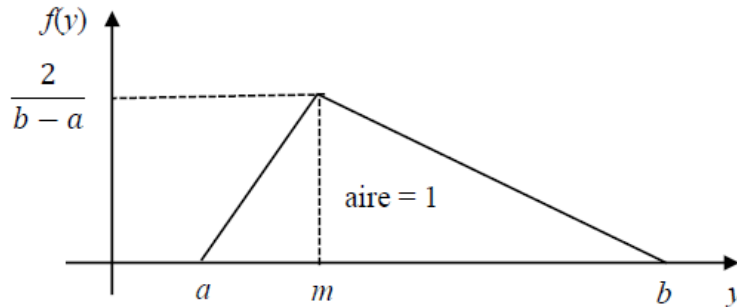


FIGURE 1.16 – Distribution triangulaire des durées de tir de transition.

1.7.2 Types de réseaux de Petri stochastiques

De nombreuses classes de RdPS sont proposées pour l'analyse des performances des systèmes de production. Les caractéristiques des différentes classes de RdPS se situent essentiellement dans la nature des transitions utilisées. Initialement un RDPS a toutes ses transitions temporisées avec un temps aléatoire qui est distribué avec une loi exponentielle, mais nous retrouvons d'autre types de transitions :

- **Les réseaux de Petri stochastique étendus :** Le réseau se compose uniquement de transitions temporisées aléatoires. Le temps est distribué avec une loi quelconque. Le processus stochastique sous-jacent au graphe des marquages est avec quelques restrictions un processus semi-markovien.
- **Les réseaux de Petri stochastiques déterministes :** Cette classe de réseau contient des transitions immédiates, des transitions à temporisations déterministes et des transitions à temporisations stochastiques distribuées avec une loi exponentielle.
- **Les réseaux de Petri stochastiques régénérateurs markoviens :** C'est une généralisation des réseaux de Petri stochastiques déterministes. Le réseau comporte des transitions immédiates, des transitions à temporisations déterministes et des transitions à temporisations stochastiques distribuées avec une loi quelconque. Le processus sous-jacent est un processus Markovien.
- **Les réseaux de Petri stochastiques généralisés :** Le réseau se compose de transition avec une temporisation nulle (transition immédiate) et de transitions avec une temporisation aléatoire distribuées exponentiellement. Le processus stochastique sous-jacent au graphe des marquages est un processus de Markov à temps continu.

1.7.3 Réseaux de Petri Stochastiques Généralisés

Les réseaux de Petri stochastiques généralisées RdPSG ont été introduits par Ajmone, afin de limiter certaines restrictions. Les réseaux de Petri stochastiques généralisées sont une extension des réseaux de Petri stochastiques autorisant deux classes de transitions[1] :

- des transitions instantanées à temporisation nulle (transition immédiate) qui sont franchies immédiatement dès qu'elles sont sensibilisées.
- des transitions temporisées à qui correspondent des variables aléatoires déterminant la durée de franchissement.

Les taux de transition peuvent dépendre du marquage. S'il existe plusieurs transitions franchissables, les transitions instantanées sont franchies avant les autres (car elles sont prioritaires par rapport aux transitions temporisées) .

Les transitions instantanées représentent des activités du système qui s'achèvent en un temps négligeable par rapport aux durées précédentes. Les états qui ne sont pas associées à une transition instantanée sont appelés des états tangibles (qui sensibilisent seulement les transitions temporisées). Sinon, si au moins une transition immédiate est sensibilisée on parle d'un état évanescent, donc un réseau de Petri stochastique généralisé présente deux types d'états :

- Les états tangibles, pour lesquels l'ensemble des transitions sensibilisées sont des transitions temporisées.
- Les états évanescents, pour lesquels au moins une transition sensibilisée est une transition immédiate.

Définition 2.2 Un RdPSG est un 8-uplet $\langle P ; T ; \text{Pre} ; \text{Post} ; \text{Inh} ; \text{pri} ; W ; M_0 \rangle$ où :

- P est l'ensemble fini des places ;
- T est l'ensemble fini des transitions temporisées et des transitions immédiates ;
- Pre, Post , et $\text{Inh} : P \times T \rightarrow \mathbb{N}$ sont les fonctions d'incidence avant, d'incidence arrière et d'inhibition respectivement ;
- $\text{pri} : T \rightarrow 0, 1$ est la fonction de priorité qui associe à chaque transition temporisée la valeur 0 et à chaque transition immédiate la valeur 1 ;
- $W : T \rightarrow \mathbb{R}^+$ est une fonction qui associe à chaque transition temporisée un taux de franchissement et à chaque transition immédiate un poids ;
- $M_0 \times P \rightarrow \mathbb{N}$ est le marquage initial du réseau, à chaque application d'incidence est associée une matrice $|P| \times |T|$.

Les avantages des RdPSG sont nombreux :

- leur puissance de modélisation et en particulier leur aptitude à prendre en compte les différents types de dépendances qui caractérisent le comportement d'un système :
la concurrence, le parallélisme et la synchronisation,
- leur puissance d'analyse des propriétés du système et de validation du modèle établi,
- leur aptitude à évaluer des mesures du comportement du système en particulier

des indices de performances.

1.7.4 Sémantique des RdPSG :

La politique d'exécution d'un RdPSG est composée de trois éléments :

1. La politique de choix

Elle consiste à la définition de la prochaine transition à tirer quand plusieurs transitions sont simultanément sensibilisées. Pour le choix de la transition à tirer, on distingue deux techniques :

- La compétition (modèle concurrentiel, politique de course, "race Policy") :
 - (i) une transition activée réserve tous les jetons dont elle a besoin afin que ces jetons ne soient pas disponibles pour d'autres transitions. Ensuite, (ii) elle attend que son temps d'exécution s'écoule. Enfin, (iii) elle se déclenche immédiatement selon la règle de déclenchement des réseaux de Petri.
- La présélection : (i) une transition activée attend que son intervalle de temps de déclenchement soit écoulé. Ensuite, (ii) elle se déclenche immédiatement selon la règle de déclenchement des réseaux de Petri, à condition qu'elle soit toujours activée à ce moment-là, c'est-à-dire que les jetons requis ne soient pas déjà consommés par une autre transition.

2. La politique de mémoire : Supposons que chaque transition temporisée soit associée à une minuterie. Lorsqu'une transition temporisée est activée, le temps correspondant est fixé à une valeur initiale et commence à se décrémenter. Lorsque le minuteur atteint la valeur zéro, la transition correspondante se déclenche. Étant donné un ensemble de transitions activées à un certain marquage, il existe trois politiques concernant la manière dont le déclenchement d'une transition doit influencer les temporisateurs des autres transitions :

- La réinitialisation ("resampling") : A chaque tir, les temporisateurs de toutes les transitions temporisées sont rejetés (mécanisme de redémarrage).
- La mémoire de la dernière sensibilisation ("enabling memory") : A chaque tir, les temporisateurs de toutes les transitions temporisées désactivées sont redémarrés. Quant aux transitions temporisées qui sont toujours activées, leurs

temporisateurs correspondants conservent leur valeur actuelle (mécanisme de continuation),

- La mémoire de toutes les sensibilisations (age memory) : A chaque tir, les temporisateurs de toutes les transitions temporisées conservent leur valeur actuelle (mécanisme de continuation).

1.8 Conclusion

Dans ce chapitre, nous avons présenté quelques généralités sur les réseaux de Petri, définissons leurs différentes règles d'évolutions. Les réseaux de Petri sont particulièrement bien adaptés à la description, les aspects dynamiques ou comportementales d'un système. Parmi les extensions des RdPs qui sont utilisées pour l'évaluation des performances est RdPS. Dans le chapitre suivant, nous avons opté pour l'évaluation des performances.

CHAPITRE 2

ETAT DE L'ART SUR L'ÉVALUATION DE PERFORMANCE

2.1 Introduction

L'évaluation des performances s'intéresse au calcul des indices de performances d'un système. Les indices de performances que l'on souhaite obtenir sous forme de grandeurs quantitatives peuvent se présenter sous différents ordres. On peut par exemple citer :

- Dans les réseaux de communication, le paramètre de performance le plus important est le temps de réponse (délai d'acheminement) qui mesure le temps qui sépare l'émission d'un message de sa réception par le destinataire.
- Dans les systèmes de production, le paramètre de performance le plus retenu est le débit en produit fini.
- Dans un domaine plus pratique (celui d'un guichet d'un organisme), on peut distinguer deux paramètres importants. Pour l'utilisateur, celui-ci s'intéresse au temps d'attente, mais pour la direction de l'organisme, c'est le nombre de clients qui importe le plus.

2.2 Concept d'évaluation de performance

Pour faire l'évaluation de performances, on doit disposer de deux éléments :

- **Un système** : c'est l'entité dont on évalue les performances. En gros, un système

est considéré comme étant un ensemble de ressources partagées entre différentes tâches. La caractéristique commune pour de tels systèmes est la présence de temps d'attente pour l'accès à ces ressources partagées.

- **Une charge :** la charge du système représente généralement le trafic en entrée, qui pourrait être l'ensemble de messages servis par un dispositif du réseau, le nombre de tâches à exécuter par un processeur. Le trafic est généralement décrit par des lois probabilistes (Poisson, Exponentielle,...).

2.3 Types d'évaluation des performances

Il existe deux approches d'évaluation pour un système, l'approche qualitative et l'approche quantitative.

- **Analyse qualitative :** consiste à vérifier les propriétés de la vivacité, la bornitude, les états d'accueil, ...etc.

- **Analyse quantitative :**

concerne le calcul des mesures que l'on veut effectuer sur un système informatique permettant de quantifier ces performances : débit, temps moyen de réponse, taux d'utilisation de ses ressources, etc.

2.4 Evaluation des indices de performances

Une fois le modèle est obtenu, on vérifie ses propriétés qualitative pour déduire son ergodicité pour faire de l'analyse quantitative. Si le modèle est ergodique, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique[15]. Plusieurs indices de performances peuvent être calculés. Parmi ces indices les plus importants on cite :

- **Fréquence moyenne de franchissement d'une transition :** On appelle fréquence moyenne (débit moyen) de franchissement d'une transition t_i , le nombre moyen de tirs de t_i en une unité du temps. Elle est calculée par :

$$\bar{\lambda}(t_j) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \pi_j. \quad (2.1)$$

où : $E(t_i)$ est l'ensemble des marquages où la transition t_i est franchissable. $\lambda_i(M_j)$ est le taux de franchissement de t_i en M_j .

- **Nombre moyen de marques dans une place :** Le nombre moyen de marques dans une place p est calculé en appliquant la formule :

$$n(p) = \sum_{i: M_i \in E} M_i(p) \pi_i; \quad (2.2)$$

où :

$M_i(p)$ est le nombre de jetons dans la place p pour le marquage M_i . E est l'ensemble des marquages accessibles.

- **Le temps moyen de séjour d'une marque dans un sous-réseau :** jeton passe dans une partie S (sous-réseau) d'un RdPSG à l'état stationnaire, peut être calculé en utilisant la formule de Little :

$$E[T] = \frac{E[T]}{E[\beta]} \quad (2.3)$$

où : $E[T]$: est le nombre moyen de jetons dans S . $E[\beta]$: est le taux d'arrivée effectif des jetons dans S .

2.4.1 Applications à l'évaluation de performances

Les réseaux de Petri stochastiques peuvent être appliqués à l'évaluation de performances à l'aide d'approches :

- formelles au travers d'analyses qualitatives (bornitude, vivacité) et quantitatives (probabilités des états, temps de séjour, temps d'absorption, temps de premier passage, ...) d'un graphe de marquage probabilisé. Des outils tels que **GreatSPN** [5] ou encore SPNP fournissent un éditeur et des algorithmes d'analyse. A titre d'illustration et sans être exhaustif, modélise un protocole de réseaux industriels de terrain à l'aide d'une classe spécifique de réseaux de Petri stochastiques (les

STPN pour Stochastic Timed Petri Nets) introduisant la modélisation du temps aléatoire sous la forme d'intervalles pour exhiber les performances en termes de temps de réponse dans un contexte critique à l'aide d'un graphe d'états probabilisé; utilise l'analyse de graphes pour déterminer les performances d'un système logistique modélisé avec un réseau de Petri stochastique enrichi par la notion de lots; dans le domaine de la gestion de production, l'analyse de GSPN permet d'évaluer des taux moyens de production ou de stocks[14].

- basées sur la simulation de Monte-Carlo, notamment pour les réseaux comportant des distributions de probabilités non exponentielles et combinant des transitions temporisées aléatoires, à durée fixe et/ou immédiates; des outils de simulation, tels que **TimeNET**[27], et dans une certaine mesure CPN Tools (avec les précautions d'usage mentionnées à la section précédente) facilitent la mise en oeuvre de ces approches basées sur la simulation. A titre d'illustration, utilise la simulation de réseaux de Petri stochastiques pour évaluer la procédure de backoff et l'espace inter-frames sur des réseaux WiFi.
- Les nombreuses applications des réseaux de Petri stochastiques à l'évaluation de performance, que soit dans le domaine des réseaux de communications, des systèmes informatiques ou encore de la gestion de production, montrent l'intérêt significatif que peut accorder à ces modèles la communauté scientifique mais aussi industrielle. Ces modèles permettent en effet la modélisation des tâches d'un système dont la durée d'exécution ne peut être modélisée qu'à travers une durée non déterministe. En cela, ils répondent parfaitement aux besoins exprimés pour l'évaluation de performances en phase d'avant-vente. En revanche, l'analyse formelle de ces modèles peut être confrontée à des phénomènes d'explosion combinatoire pour des modèles de grande taille et impose, dans la plupart des cas, le recours à des distributions de probabilités exponentielles permettant de se ramener dans un cadre markovien ou semi-markovien. Si le recours à d'autres types de distributions (uniformes, Weibull, Erlang, ...) est nécessaire ou si les modèles à analyser sont de grande taille (plusieurs centaines de places), le recours à la simulation de Monte-Carlo peut s'avérer indispensable[21].

2.5 Techniques de l'évaluation de performances

Il existe différentes techniques pour l'évaluation de performance d'un système :

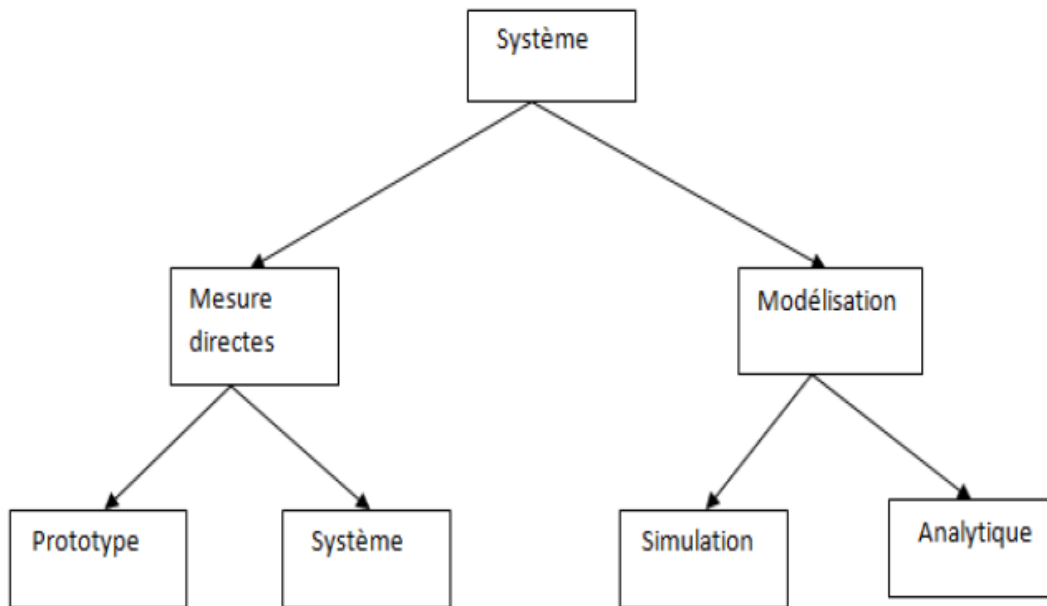


FIGURE 2.1 – Techniques de l'évaluation de performances.

2.5.1 Les mesures directes

Cette méthode est la seule à pouvoir offrir l'image réelle de l'état d'un système réel en tenant compte de toutes ses caractéristiques. Mais elle a des inconvénients dans le fait que les résultats obtenus à un moment donné ne permettent pas toujours de prévoir le comportement du système dans d'autres conditions et générer les coûts elles sont souvent très difficiles à appliquer.

2.5.2 Méthodes analytiques

Chaîne de Markov Les Chaîne de Markov sont des processus stochastiques très utilisés dans l'évaluation de performance des systèmes P2P de nature discrète plus précisément des systèmes dynamiques. Ce type de processus permet souvent de caractériser les états stationnaires du système , c'est-à-dire les états possibles du système à un instant choisi "aléatoirement" loin dans le temps. Les probabilités d'état stationnaire du système représentent aussi la proportion de temps que le système a passé dans chacun de ces états au cours d'une très longue observation du système. Si les probabilités sont bien définies, on dit que le système admet un état stationnaire. Le système tend alors vers cet état au bout d'un certain temps.

Description

Il y'a deux types de Chaîne de Markov.

Chaîne de Markov à Temps Discret (CMTD)

Soit $\{ f(X_n), n = 0, 1, 2, \dots \}$ un processus stochastique à temps discret. Le processus stochastique, $\{ (X_n), n = 0, 1, 2, \dots \}$ est une Chaîne de Markov à temps discret, si pour tout $n=0,1,2, \dots$ et quelque soient les $n + 1$ états $i_0, i_1, \dots, i_{n-1}, i, j$ de E tel que :

$$P[X(n + 1) = j; X(n) = i, \dots, X(0) = i_0] = 0 \quad (2.4)$$

La propriété de Markov est vérifiée :

$$P[X(n + 1) = j/X(n) = i, \dots, X(0)] = i_n = P[X(n + 1) = j/X(n) = i] = p_{ij} \quad (2.5)$$

La probabilité $p_{ij}(n)$ s'appelle la probabilité de transition de l'état i vers l'état j à l'instant n . La propriété de Markov signifié que l'état futur de la chaine à l'instant $n + 1$ ne dépend que de son état présent à l'instant n , et non des états passés par lesquels elle est passée. Une Chaîne de Markov $\{ (X_n), n=0,1,2, \dots \}$, est dite homogène si pour tout $n=0,1,2, \dots$ pour tout états i, j de E , la probabilité,

$$P[X(n + 1) = j/X(n) = i] = p_{ij}, \quad (2.6)$$

est indépendante de l'instant n . La probabilité p_{ij} s'appelle la probabilité de transition de l'état i vers l'état j . La matrice $P = (p_{ij}) / i, j \in E$, dont les éléments sont des probabilités de transition p_{ij} lorsque i et j décrivent l'espace d'états E , s'appelle la matrice des transitions.

$$P = \begin{pmatrix} p_{00} & p_{10} & \cdots & p_{n0} & \cdots \\ p_{01} & p_{11} & \cdots & p_{n1} & \cdots \\ \vdots & \cdots & \cdots & \cdots & \cdots \\ p_{0n} & p_{1n} & \cdots & p_{nn} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Une Chaîne de Markov à temps discret est totalement définie par sa matrice des transitions [6].

Chaîne de Markov à Temps Continu (CMTC)

Un processus stochastique $\{ X(t), t \geq 0 \}$ est dit Chaîne de Markov à temps continu, s'il vérifie les trois conditions suivantes :

1. (a) L'espace d'états S est dénombrable ;
- (b) Le temps d'observation est de nature continue ;
- (c) Le processus $X(t)$ vérifie la propriété de Markov ;

Une Chaîne de Markov homogène à temps continu est définie par ses probabilités de transition :

$$P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, \dots, X(t_0) = i_0] = P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, t_0 < t_1 < \dots < t_n, n \in N] \quad (2.7)$$

et aussi définie par :

$$p_{ij} = P[X(s+t) = j / X(s) = i] s \geq 0. \quad (2.8)$$

C'est à dire que les probabilités $P[X(t_n) = j / X(t_n=i)]$ ne dépendent pas des instants d'observations t_n et t_{n-1} , mais uniquement de la durée $t_n - t_{n-1}$ qui sépare les deux observations [6].

Système d'attente et Réseaux files d'attente

Description

Les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie contemporaine. On les rencontre dans les domaines d'activité les plus divers (guichet de poste, trafic routier, central téléphonique, atelier de réparation, ...). L'étude mathématique des phénomènes d'attente constitue un champ d'application important des processus stochastiques. On parle de phénomènes d'attente chaque fois que certaines unités appelées " client " se présentent d'une manière aléatoire à des " stations " afin de recevoir un service dont la durée est généralement aléatoire [23].

Processus d'arrivée

L'arrivée des clients à la station sera décrite à l'aide d'un processus stochastique de comptage $N(t)$, $t \geq 0$ (On notera indifféremment Nt) ou bien $N(t)$. Si A_n désigne la variable aléatoire mesurant l'instant d'arrivée du n^{ieme} client dans le système, on aura ainsi : $A_0 = 0$ (par convention) et $A_n = \inf t, N_t = n$ Si T_n désigne la variable aléatoire mesurant le temps séparant l'arrivée du $(n-1)^{ieme}$ client et du n^{ieme} client, on a alors : $T_n = A_n - A_{n-1}$.

Temps de service

La distribution du temps de service la plus simple à étudier est la distribution exponentielle. Cependant, la propriété " sans mémoire " de la loi exponentielle fait que celle-ci n'est généralement pas très réaliste pour modéliser les phénomènes réels. On est donc souvent obligé de recourir à d'autres distributions de service.

Structure et discipline de la file

— **Nombre de serveurs :** Une station peut disposer de plusieurs serveurs en parallèle. Soit C le nombre de serveurs. Dès qu'un client arrive à la station, soit il y a

un serveur libre et le client entre instantanément en service, soit tous les serveurs sont occupés et le client se place dans la file en attente de libération d'un des serveurs. La plupart du temps les serveurs sont supposés identiques (ils possèdent donc la même distribution) et indépendants les uns des autres.

- **Capacité de la file** : La capacité de la file à accueillir des clients en attente de service peut être finie ou infinie. Soit K la capacité de la file (incluant le ou les clients en service). Une file à capacité illimitée vérifie $K = +\infty$. Lorsque la capacité de la file est limitée et qu'un client arrive alors que cette dernière est pleine, le client est perdu.
- **Discipline de service** : La discipline de service détermine l'ordre dans lequel les clients sont rangés dans la file et y sont retirés pour recevoir un service. Les disciplines les plus courantes sont :
 - **FIFO** (first in, first out) PAPS (premier arrivée, premier servi) : c'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée.
 - **LIFO** (last in, first out) DAPS (dernier arrive, premier servi). Cela correspond à une pile, dans laquelle le dernier client arrive (donc posé sur la pile) sera le premier traité (retiré de la pile).
 - **FIRO** (first in, Random out) les clients sont servis de manière aléatoire.

Notation de Kendall La notation de Kendall permet de décrire le système à files d'attente par une suite de paramètres. Pour identifier un système à files d'attentes, le formalisme suivant a été proposé et est unanimement adopté : $A/B/n/K/N$

A : La première lettre identifie la loi du processus des arrivées, B : la seconde le processus des services, avec dans les deux cas les conventions :

M : loi "sans mémoire" (arrivées poissoniennes, service exponentiel) ;

D : loi déterministe ;

E_k : loi "Erlang- k "

H_k : loi "hyper exponentielle" d'ordre k ;

GI : loi générale, les variables successives étant indépendantes ;

G : loi générale, sans hypothèse d'indépendance.

n : Le chiffre qui suit donne le nombre de serveurs,

K et N : les lettres qui suivent (facultatives) identifient la taille de la file d'attente et la taille de la population source (si ces valeurs sont omises, elles sont supposées infinies).

Quelques mesures de performance

L'analyse théorique d'un modèle de files d'attente a pour objet de saisir quantitativement et qualitativement le fonctionnement du système en question. Pour cela il faut définir les critères et les mesures susceptibles d'atteindre cet objectif afin d'être capable de déterminer à l'avance les performances du système, les effets d'un changement ou encore d'identifier les paramètres les plus sensibles. Pour un système constitué d'une seule file d'attente, les principales mesures de performances sont.

1. N_s : Le nombre moyen de clients dans le système,
2. T_s : Le temps moyen de séjour dans le système,
3. N_q : Le nombre moyen de clients en attente,
4. W_s : Le temps d'attente moyen d'un client dans le système,
5. W_q : Le temps d'attente moyen d'un client dans la file,
6. Le taux d'occupation $\rho = \lambda / \mu$ d'un ou des serveurs.

Ses facteurs sont liés de la manière suivante :

- $N_s = \lambda W_s$
- $N_q = \lambda W_q + 1/\mu$
- $W_s = \lambda / \mu$
- $N_s = \lambda / \mu + N_q$

Il est important de remarquer que ces grandeurs sont aléatoires, leur espérance et distribution dépendent a priori de l'instant t où elles sont considérées. Nous devrions donc parler du nombre de clients dans le système à l'instant t . Ce n'est que sous l'hypothèse de stationnarité du système que ces mesures de performance deviennent indépendantes du temps lors de l'étude d'une file, on cherche à estimer l'espérance en régime stationnaire des grandeurs précédentes.

Modèle d'attente Markovien

Les modèles markoviens de files d'attente sont des systèmes où les deux quantités stochastiques principales " le temps des inter-arrivées " et " la durée de service " sont des variables aléatoires indépendantes, exponentiellement distribuées. La propriété " sans mémoire " de la loi exponentielle facilite l'analyse de ces modèles.

Système M/M/1 :

Pour le système d'attente le plus simple, M/M/1, le ux des arrivées est poissonnien, de paramètre λ et la durée de service est exponentielle, de paramètre μ . La capacité d'attente est illimitée, la discipline de service est FIFO et il y a une seule station de service. La condition de stabilité (équilibre) de ce système est : $\rho = \lambda / \mu < 1$. **Système M/M/ $+\infty$:** On considère un système composé d'un nombre illimité de serveurs indépendants les uns des autres. Dès qu'un client arrive, il rentre instantanément en service. Cette le particulière ne comporte donc pas de file d'attente. Le processus d'arrivée des clients est poissonnien de taux λ et les temps des services sont exponentiels de taux μ_i , $i = 1, +\infty$. **Système M/M/1/K :** Le modèle M/M/1/K correspond au cas capacité de K clients, K représente le nombre maximum de clients dans le système, c'est-à-dire en attente et en service. Si un client arrive et trouve devant lui K client dans le système celui la sera perdu, donc la probabilité de perte est égale à la probabilité de trouver K clients dans le système au régime stationnaire, on a : $p_{perte} = \rho^k (1 - \rho) / (1 - \rho^{k+1})$

Modèle d'attente non-Markovien

En s'écartant de l'hypothèse d'exponentialité de l'une de deux quantités stochastiques " le temps des inter-arrivées " et " les durées de service " ou en introduisant des paramètres supplémentaires spécifiques au n'aura plus de processus Markovien. Ce qui rend l'analyse du modèle analytique très délicate, voire impossible. C'est pourquoi, souvent on se ramène à un processus Markovien.

Système G/G/1 : Pour d'écrire l'évaluation du système G/G/1, on a besoin de caractériser, en plus du nombre de clients dans le système, le temps déjà passé dans le serveur pour le client en service et le temps déjà écoulé depuis l'arrivée du dernier client.

Système M/G/1 : Une file M/G/1 est caractérisée par un processus d'arrivée poissonnien et par des temps de service indépendants et identiquement distribués selon une loi de probabilités " quelconque " (mais ne prenant que des valeurs non négatives).

L'évaluation du nombre de clients dans une file ne peut plus être modélisée par une Chaîne de Markov à temps continu car si, à un instant quelconque $t > 0$, X_t clients sont présents, l'évaluation future de la variable X_t dépend non seulement de la valeur de X_t mais également du temps de service déjà alloué au client occupant le serveur.

Les réseaux de files d'attentes

Un réseau de file d'attente est un ensemble fini s_1, s_2, \dots, s_n de systèmes de file d'attente. Le séjour d'un client dans le système consiste à parcourir une partie ou l'ensemble de tous les systèmes s_i ($i=1, \dots, n$), selon un itinéraire déterministe ou aléatoire. En d'autres termes, un réseau de files d'attente est composé d'un ensemble de stations de service et d'un ensemble de clients. Ce système est caractérisé par :

- Les processus d'arrivées des clients au réseau.
- Les temps de service des clients aux stations.
- Le cheminement des clients d'une station à une autre.
- Les disciplines de service à chaque station.

Les différents types de réseaux

On distingue deux types de réseaux :

Réseaux ouverts : Ces réseaux possèdent une ou plusieurs entrées de l'extérieur. Le nombre de clients pouvant se trouver à un instant donné dans un réseau ouvert n'est pas limité.

Réseaux fermés : Ces réseaux ne possèdent pas d'entrées l'extérieur, le nombre de clients dans ce cas est fini constant. Pour un mécanisme de routage probabiliste, on définit p_{ij} la probabilité qu'un client qui quitte la station i , se rende à la station j .

2.5.3 Simulation

La simulation est une technique largement utilisée dans l'évaluation de performances des systèmes informatiques et réseaux de communication. Il s'agit d'implanter un mo-

dèle simplifié du système à l'aide d'un programme de simulation adéquat. Elle présente l'avantage par rapport aux méthodes analytiques de traduire le comportement du système à l'évaluer d'une manière plus réaliste et représente un moyen utile pour prédire les performances d'un système et les comparer sous différentes configurations. La simulation permet en plus de visualiser les résultats sous formes de graphes faciles à analyser et à interpréter. Elle rend possible l'analyse systématique des systèmes lorsque les solutions analytiques ne sont pas disponibles et l'expérimentation sur le système considéré est impossible ou non pratique [28].

Les étapes de la simulation : Les différentes étapes à suivre pour faire une simulation d'un système sont[7] :

- Formulation du modèle : cette étape consiste à identifier et à analyser le problème, en déterminant ses composantes, leurs relations et les frontières entre le système et son environnement.
- Elaboration du modèle : cette étape consiste à extraire un modèle aussi fidèle que possible du système réel.
- Identification du modèle et collecte de données : la collecte de données est indispensable pour l'estimation des paramètres du modèle. Ceci requiert une connaissance des méthodes statistiques et des tests d'hypothèses.
- Validation du modèle : cette étape consiste à évaluer les performances du modèle puis de les comparer à celles du système réel.
- Exécution de la simulation : pour mettre à l'épreuve le modèle, le concepteur doit effectuer plusieurs exécutions et recueillir les résultats.
- Analyse et interprétation des résultats : une fois les résultats obtenus, le concepteur passe à l'analyse et à l'interprétation de ces résultats pour donner des recommandations et des propositions.
- Conclusion : cette dernière étape consiste à évaluer les perspectives d'exploitation du modèle pour d'autres préoccupations.

Il existe plusieurs simulateurs, parmi lesquels nous avons choisi de présenter trois que nous classifions comme plus connus, utiles et disponibles.

1. GreatSPN

Présentation de GreatSPN : est un progiciel pour la modélisation, la validation et l'évaluation des performances de systèmes distribués utilisant des réseaux de Petri stochastiques généralisés et leur extension colorée : Stochastic Well-formed Nets. L'outil fournit un cadre convivial pour expérimenter des techniques de modélisation basées sur un réseau de Petri chronométré[8, 2] .

Il implémente des algorithmes d'analyse efficaces pour permettre son utilisation sur des applications assez complexes, pas seulement des exemples de jouets[5].

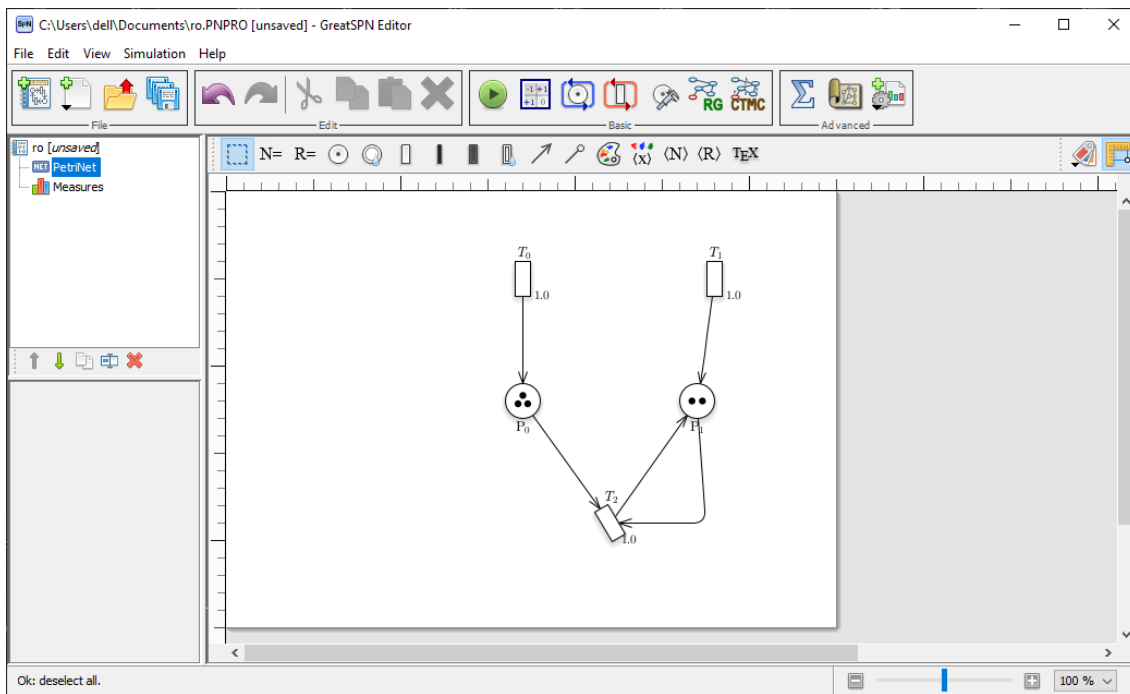


FIGURE 2.2 – Interface utilisateur graphique de GreatSPN.

Fonctionnalités principales de GreatSPN

Les fonctions principales de l'outil peuvent être décrites comme suit :

2. TimeNET

La conception de systèmes techniques non triviaux est plus efficace et moins risquée lorsque l'effet des décisions de conception peut être estimé tôt. L'évaluation des performances et de la fiabilité basée sur un modèle est une aide importante. Dans le domaine des systèmes techniques commandés par ordinateur, de nom-

breuses classes de modèles dans le domaine des systèmes à événements discrets stochastiques ont été proposées[27].

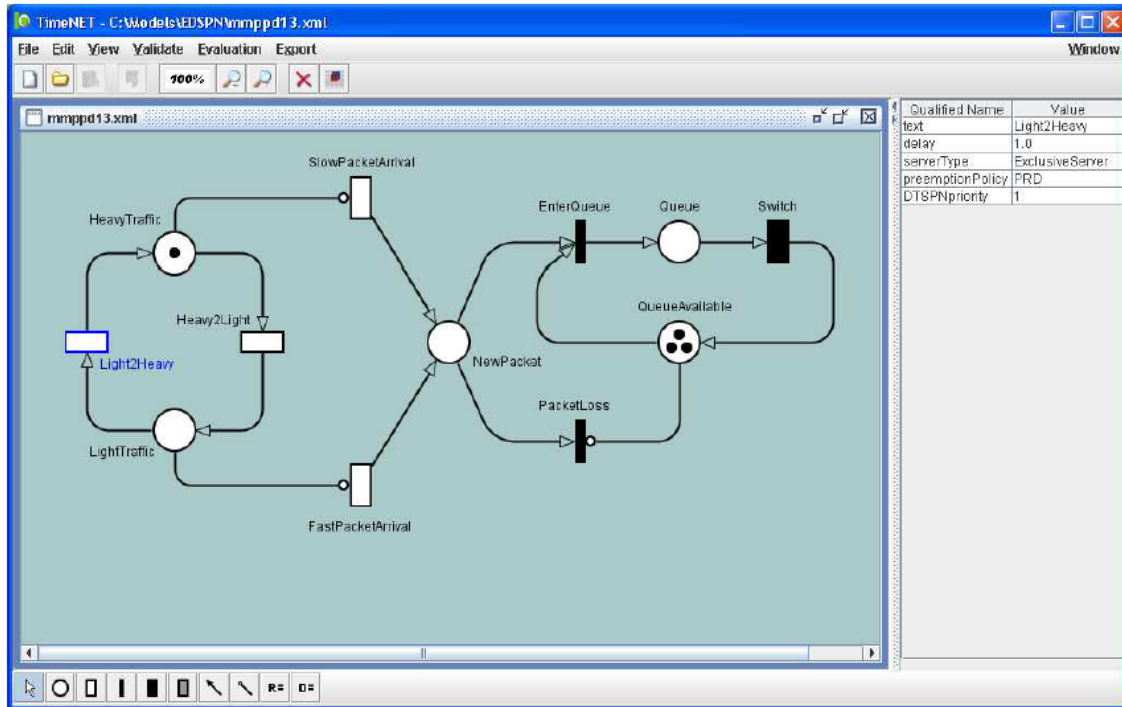


FIGURE 2.3 – Interface utilisateur graphique de TimeNET 4.0.

Les réseaux de Petri stochastiques et leurs variantes sont largement acceptés aujourd'hui comme un outil mature pour décrire des systèmes présentant une concurrence, des restrictions de ressources, une synchronisation, une dépendance temporelle et un comportement probabiliste.

TimeNET est un outil logiciel pour la modélisation et l'analyse de réseaux de Petri stochastiques avec des temps de tir distribués de manière non exponentielle. Il prend en charge la modélisation graphique des réseaux de Petri non colorés et colorés ainsi que des chaînes de Markov. De nombreux algorithmes d'évaluation des performances et d'analyse structurelle sont disponibles ainsi qu'un jeu de jetons interactif.

3. GRIF[13]

GRaphiques Interactifs pour la Fiabilité (GRIF) est une plate-forme logicielle d'analyse des systèmes qui permet de déterminer les indicateurs fondamentaux

de la sûreté de fonctionnement :

- Fiabilité
- Disponibilité
- Performance
- Sécurité

GRIF laisse le choix à l'utilisateur d'opter pour la technique de modélisation la plus adéquate à la résolution du système étudié : blocs diagrammes, arbres de défaillance, graphes de Markov, réseaux de Petri. Des architectures déjà intégrées dans le logiciel facilitent cette modélisation. Développé au sein de TotalEnergies, GRIF bénéficie de plus de 30 ans de Recherche et Développement. Cette plateforme dispose ainsi de moteurs de calcul matures, très performants et aux capacités de modélisation propres à répondre aux besoins de l'ensemble des études fiabilistes[17].

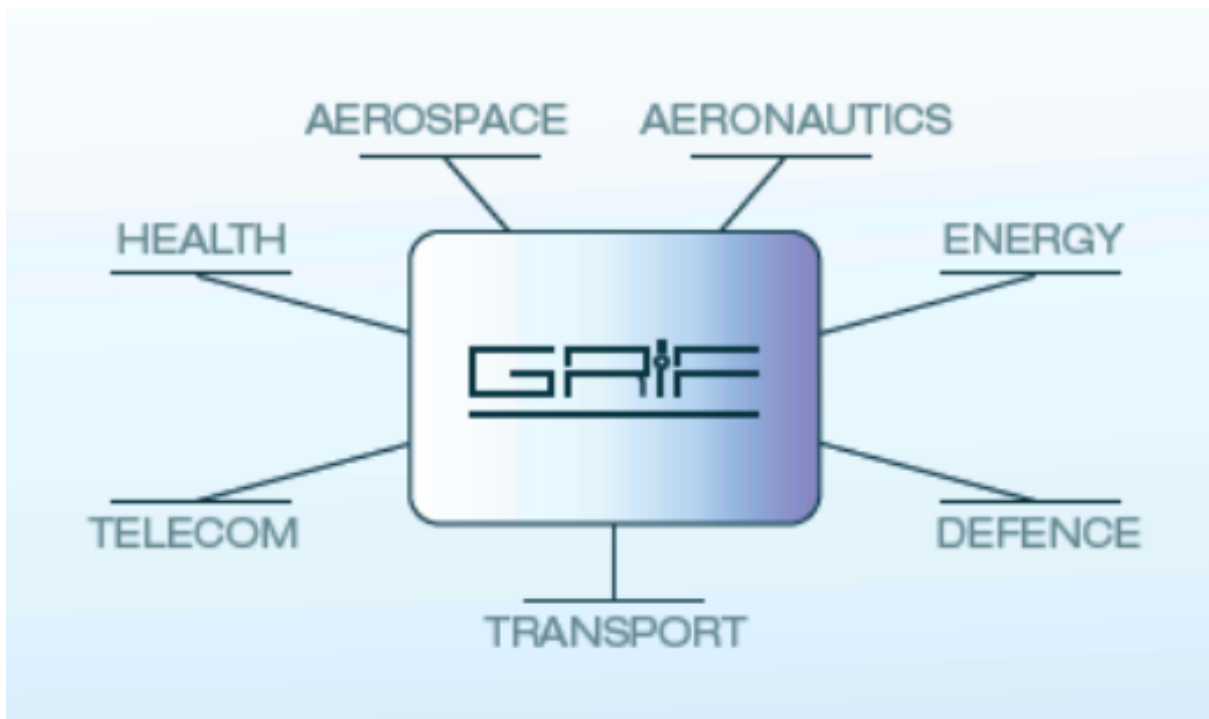


FIGURE 2.4 – Logiciel GRIF

GRIF est structuré en 3 packages

- (a) Le package Booléen comprend 7 modules.
- (b) Le package Simulations comprenant les modules Petri, BStock, Petro et Flex :
Ces modules utilisent le moteur de calcul MOCA-RP basé sur la simulation de Monte Carlo.
- (c) Le package Markovien comprenant le module Markov.

2.6 Conclusion

Dans ce chapitre, nous avons introduit les notions d'évaluation de performances. Nous avons présenté également les méthodes d'évaluation des performances telles que la simulation et les méthodes analytiques.

Dans le chapitre suivant, nous appliquerons une méthode simulation par le logiciel GRIF pour modéliser et évaluer les performances d'un réseau petri stochastiques.

CHAPITRE 3

ANALYSE D'UN MODÈLE DE RÉSEAU DE PETRI STOCHASTIQUES

3.1 Introduction

Nous proposons un modèle de réseau de Petri stochastiques, après nous analysons ce modèle avec différentes valeurs puis évaluons ses performances, son temps de réponse, le nombre moyen de jetons dans le système, et la Fréquence moyenne de franchissement des transition.

Les performances de ce modèle sont calculées à l'aide d'un simulateur GRIF.

3.2 Modélisation

On a proposé un modèle de réseau de Petri stochastique, voir la figure 3.1.

Les arrivées aux réseaux suivent dans la distribution de Poisson et les taux d'arrivée sont λ_i , $i \in \{1, 2\}$ et sont indépendants du processus d'échec. Les places P_1 et P_2 sont des places illimités. Nous sélectionnons n jetons pour qu'ils se rendent à la place P_1 , P_2 respectivement. après avoir franchi les transitions T_i , $i \in \{1, 2, 3, 4\}$. Ensuite, ils franchissent la transition T_5 avec un taux μ . La figure 3.3 présente un modèle de réseau de Petri stochastique.

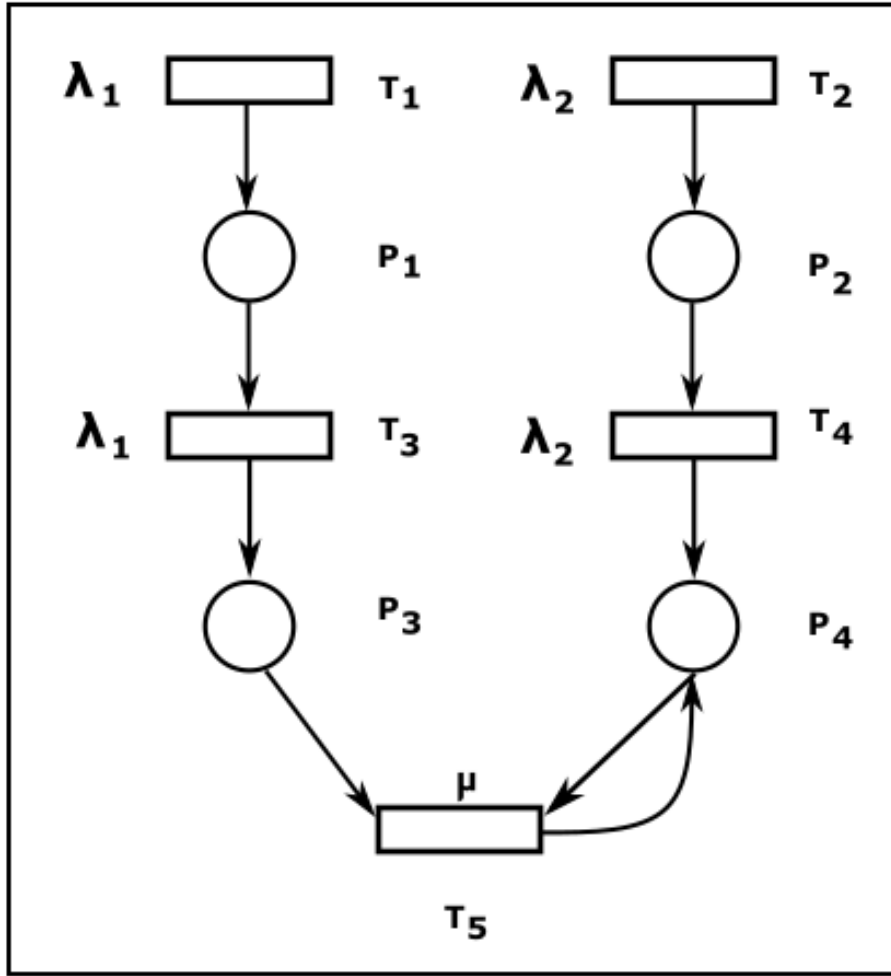


FIGURE 3.1 – Le réseau de petri stochastiques.

- $P = \{ P_1, P_2, P_3, P_4 \}$ un ensemble de quatre places telles que :
 - P_1 : place illimité.
 - P_2 : place illimité.
 - P_3 : place à capacité limité.
 - P_4 : place à capacité limité.
- $T = \{ T_1, T_2, T_3, T_4, T_5 \}$ un ensemble de cinq transitions, telles que :
 - T_1 représente les arrivées suivant une loi exponentielle, avec un taux de franchissement λ_1 .
 - T_2 représente les arrivées suivant une loi exponentielle, avec un taux de franchissement λ_2 .

- T_3 représente les arrivées suivant une loi exponentielle, avec un taux de franchissement μ .
- T_4 représente les arrivées suivant une loi exponentielle, avec un taux de franchissement λ_2 .
- T_5 représente les arrivées suivant une loi exponentielle, avec un taux de franchissement λ_1 .

3.3 Analyse de modèle

On a proposé un modèle de réseau de Petri stochastiques voir la figure 3.1 En utilisant la simulation par le simulateur GRIF, on a calculé les indices des performances.

pour calculer le temps moyen de réponse, nous avons considéré les critères d'entrées sont :

$\lambda_2 = 0.01$, $\mu = 1$ et les différentes valeurs de λ_1 sont :

$\lambda_1 = 0.00001, 0.0001, 0.001, 0.01, 0.1$

pour les différentes valeurs de n sont :

$n = 1, 10, 100, 10^3, 10^4, 10^5, 10^6$.

3.3.1 Résultats de la simulation

Après l'implémentation du modèle dans le simulateur GRIF, nous avons pu simuler notre système selon les différentes valeurs, nous avons récupéré les résultats de simulation associés aux places et aux transitions, qui seront donnés dans le tableau 3.1, tableau 3.2, et le tableau 3.3, pour des autres résultats voir l'annexe 1.

Table 3.1 présente temps moyen de réponse en fonction de n, différentes valeurs de taux d'arrivée λ_1 .

$\lambda_1 \backslash n$	0.00001	0.0001	0.001	0.01	0.1
1	19572.83303	21900.07944	25449.28722	27714.07459	29394.33438
10	19533.75038	25744.56141	21789.18216	27785.13838	29544.214
100	19681.14167	25668.71792	21952.35916	28078.93531	30254.03281
1000	19779.39553	26089.77688	22026.26187	28637.30316	38502.57257
10000	29471.32407	31758.26847	35600.44718	37634.52542	38502.57257
100000	29586.24425	31871.57631	35706.82337	37668.84949	38502.57257

TABLE 3.1 – Temps moyen de réponse.

D'après le tableau 3.1, nous remarquons que le temps moyen de réponse augmente pour chaque augmentation de λ_1 , parce que le nombre des arrivées augmente.

Aussi à chaque fois que n augmente, le temps moyen de réponse augmente.

Table 3.2 présente le nombre moyen de jetons dans le système en fonction de n , différentes valeurs de taux d'arrivée λ_1 .

$\lambda_1 \backslash n$	0.00001	0.0001	0.001	0.01	0.1
1	49.13686091	49.75461151	51.20734133	57.11949155	67.92700487
10	57.59478144	58.22908083	59.91989331	66.3810992	78.11803323
100	150.7327483	151.2353943	153.02793	158.144432	166.9980823
1000	1092.912092	1093.424527	1095.878107	1102.128796	1120.554892
10000	14989.9465	14991.65546	14996.50442	15036.88026	15496.88261
100000	194989.178	194990.9089	194995.8259	195036.7726	195496.8826

TABLE 3.2 – Nombre moyen de jetons dans le système.

D'après le tableau 3.2, nous remarquons que le Nombre moyen de jetons dans le système augmente pour chaque augmentation de λ_1 , parce que le nombre des arrivées augmente.

Aussi à chaque fois que n augmente, le Nombre moyen de jetons dans le système augmente.

Table 3.3 présente la fréquence moyenne de franchissement la transition T_1 (le nombre moyen de tire de T_1 en une unité du temps) en fonction de n , différentes valeurs de taux d'arrivée λ_1 .

$n \backslash \lambda_1$	0.00001	0.0001	0.001	0.01	0.1
1	0.12	0.9	10.14	98.53	1001.94
10	0.13	0.91	10.72	98.81	999.65
100	0.12	0.9	9.54	101.16	1001.72
1000	0.12	0.8	8.6	96.38	1008.21
10000	0.13	0.89	10.3	99.87	1006.26
100000	0.13	0.89	10.3	99.82	1006.26

TABLE 3.3 – Fréquence moyenne de franchissement la transition T_1 .

D'après le tableau 3.3, nous remarquons que la fréquence moyenne de franchissement la transition T_1 augmente par chaque λ_1

Aussi à chaque fois que n augmente, la fréquence moyenne de franchissement la transition T_1 est presque stable.

3.4 Conclusion

Après avoir analysé le réseau de Petri stochastiques, nous avons simulé et évalué les performances de notre modèle avec le simulateur GRIF avec différents valeurs de n et λ_1 , et à partir des résultats obtenus nous avons conclu le nombre limite des arrivées sont augmente pour chaque n , et pour λ_1 .

CONCLUSION GÉNÉRALE

Les méthodes analytiques (réseaux de Petri, réseaux de files d'attente, Chaîne de Markov ...) et la simulation (GreatSPN, TimeNET, GRIF ...) constituent des outils très performants pour la modélisation et l'évaluations, des performances des réseaux de Petri stochastiques. Ceci permettra en particulier, de mesurer plusieurs facteurs de performance.

Dans le cadre de ce travail, nous avons donné une brève description du réseau de petri (RdP). Nous avons défini les différents types selon des critères de classification spécifiques. En particulier, nous avons présenté les différents modèles analytiques et simulation proposés pour l'évaluation des performances de système RdPS.

Le but de notre étude est l'évaluation des performances de RdPS avec le simulateur GRIF.

L'interêt majeur des RdPS est de pouvoir combiner l'analyse qualitative et l'analyse quantitative. Les RdPS permettent l'obtention de résultats des indices de performances aisément. Ils permettent également d'inclure facilement les différentes structures de synchronisation.

L'analyse du système retenu est rendue facile grâce à l'utilisation du logiciel GRIF qui est basé sur des algorithmes construits selon les propriétés des RdPs et la simulation de Monté Carlo.

BIBLIOGRAPHIE

- [1] M. AJMONE M, G. C. et G. B. “A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems”. In : *ACM Transactions on Computer Systems (TOCS)* 2.2 (1984), p. 93-122.
- [2] E. G. AMPARORE et Susanna DONATELLI. “GreatTeach : a tool for teaching (stochastic) Petri nets”. In : *International Conference on Applications and Theory of Petri Nets and Concurrency*. Springer. 2018, p. 416-425.
- [3] Y. ATAMNA. “Réseaux de Petri temporisés stochastiques classiques et bien formés. Définition, analyse et application aux systèmes distribués temps réel”. Thèse de doct. Toulouse, 1994.
- [4] R. Berghoute B. “Identification des systèmes à événements discrets par réseaux de Petri”. Thèse de doct. Université du Havre, 2009.
- [5] S. BAARIR et al. “The GreatSPN tool : recent enhancements”. In : *ACM SIGMETRICS Performance Evaluation Review* 36.4 (2009), p. 4-9.
- [6] F. BAUSE et P. S. K. *Stochastic petri nets*. T. 1. Citeseer, 2002.
- [7] N. BERNINE et al. “Towards a performance analysis of composite web services using Petri nets”. In : *International Journal of Mathematics in Operational Research* 17.4 (2020).
- [8] G CHIOLA et al. *1.7 : Graphical editor and analyzer for timed and stochastic Petri nets Perform. Eval.* 1995.

- [9] A. CHOQUET-GENIET. *Les réseaux de Petri-Un outil de modélisation*. Dunod, 2006.
- [10] R. DAVID et A. Du Grafset aux réseaux de Petri. Hermes, 1992.
- [11] M. DIAZ. *Les réseaux de Petri : modèles fondamentaux*. Hermès science publications, 2001.
- [12] M. DIAZ. *Petri nets : fundamental models, verification and applications*. John Wiley & Sons, 2013.
- [13] GRIF. “Graphical Interface for Reliability Forecasting”. In : (2016).
- [14] I. HATONO, K. Y. et H. T. “Modeling and on-line scheduling of flexible manufacturing systems using stochastic Petri nets”. In : *IEEE Transactions on Software Engineering* 17.2 (1991), p. 126.
- [15] J. HÊCHE et T. M. L. *Recherche opérationnelle pour ingénieurs*. T. 2. PPUR presses polytechniques, 2003.
- [16] A. HETTAB. “De M-UML vers les réseaux de Petri■ Nested Nets”. In : ().
- [17] <http://grif-workshop.fr>. 2022.
- [18] <https://satodev.com/nos-produits/grif-simulation>. 2022.
- [19] Kurt J. *Coloured Petri nets : basic concepts, analysis methods and practical use*. T. 1. Springer Science & Business Media, 1997.
- [20] T. MURATA. “Petri nets : Properties, analysis and applications”. In : *Proceedings of the IEEE* 77.4 (1989), p. 541-580.
- [21] M. AA. NDIAYE. “Évaluation de performance d’architecture de contrôle-commande en réseau dans un contexte incertain d’avant-vente”. Thèse de doct. Université de Lorraine, 2017.
- [22] C. A. PETRI. “Interpretations of net theory”. In : *GMD, Technical Report* (1976).
- [23] Philippe ROBERT. *Réseaux et files d’attente : méthodes probabilistes*. T. 35. Springer Science & Business Media, 2000.
- [24] T. ROQIYA et M. B. “Le Développement d’un Outil de Transformation des Modèles Orientés Aspect vers les Réseaux de Pétri, Basé sur la Transformation De Graphes”. Thèse de doct. Abdelhafid Boussouf University centre-Mila, 2018.

- [25] S. M. ROSS. *Introduction to probability models*. Academic press, 2014.
- [26] S. SAGGADI. “Optimisation des temps d’attente des système flexibles de production basée sur les réseaux de petri”. Thèse de doct. Boumerdes, Université M’hamed Bougara. Faculté des sciences de l’ingénieur, 2007.
- [27] A. ZIMMERMANN et M. K. “TimeNET 4.0”. In : (2007).
- [28] Enrico ZIO. “Monte carlo simulation : The method”. In : *The Monte Carlo simulation method for system reliability and risk analysis*. Springer, 2013, p. 19-58.
- [29] W. M. ZUBEREK. “Timed Petri nets and preliminary performance evaluation”. In : *Proceedings of the 7th annual Symposium on Computer Architecture*. 1980, p. 88-96.

Table 3.4 présente la fréquence moyenne de franchissement la transition T_2 (le nombre moyen de tire de T_2 en une unité du temps) en fonction de n , différentes valeurs de taux d'arrivée λ_1 .

$n \backslash \lambda_1$	0.00001	0.0001	0.001	0.01	0.1
1	99.44	99.44	99.86	97.3	100.89
10	99.23	99.74	99.21	97.27	101.16
100	99.86	100.98	101.74	100.47	100.92
1000	83.1	84.5	88.99	94.9	101.59
10000	100.89	102.02	100.95	100.99	99.24
100000	100.9	101.91	100.79	100.9	99.24

TABLE 3.4 – Fréquence moyenne de franchissement la transition T_2 .

Table 3.5 présente la fréquence moyenne de franchissement la transition T_3 (le nombre moyen de tire de T_3 en une unité du temps) en fonction de n , différentes valeurs de taux d'rivée λ_1 .

$n \backslash \lambda_1$	0.00001	0.0001	0.001	0.01	0.1
1	1.01	1.29	7.83	87.9	986.87
10	10	10.31	16.95	96.92	992.93
100	100	100.33	106.54	188.66	1088.24
1000	1000	1000.22	1005.78	1087.03	1982.57
10000	10084.99	10000.26	10007	10083.55	10084.99
100000	10123.83	10121.96	10121.55	10117.32	10084.99

TABLE 3.5 – Fréquence moyenne de franchissement la transition T_3 .

Table 3.6 présente la fréquence moyenne de franchissement la transition T_4 (le nombre moyen de tire de T_4 en une unité du temps) en fonction de n , différentes valeurs de taux d'rivée λ_1 .

$n \backslash \lambda_1$	0.00001	0.0001	0.001	0.01	0.1
1	0.01	0.29	6.83	86.91	985.94
10	0	0.31	6.95	86.93	983.02
100	0	0.33	6.54	88.66	988.32
1000	0	0.22	5.78	87.03	982.67
10000	0	0.26	7	88.61	969.85
100000	0	0.26	7	88.61	969.85

TABLE 3.6 – Fréquence moyenne de franchissement la transition T_4 .

Table 3.6 présente la fréquence moyenne de franchissement la transition T_5 (le nombre moyen de tire de T_5 en une unité du temps) en fonction de n , différentes valeurs de taux d'arrivée λ_1 .

$n \backslash \lambda_1$	0.00001	0.0001	0.001	0.01	0.1
1	87.05	87.56	88.41	86.66	89.72
10	88.09	88.66	88.88	85.19	90.17
100	93.01	92.58	90.79	88.12	89
1000	76.72	76.95	79.62	84.06	90.55
10000	90.64	91.34	91.92	91.27	90.44
100000	90.61	91.35	92.01	91.23	90.44

TABLE 3.7 – Fréquence moyenne de franchissement la transition T_5 .

Présentation du logiciel GRIF

Package simulation : module Petri

Le package simulation de GRIF est étudié pour simuler des systèmes dynamiques. Le module de Petri (réseau de Petri) est généré au format Moca-RP qui est le moteur de simulation de Monte Carlo. Puisque Moca-RP simule le comportement du système, il peut fournir toutes les informations sur le réseau de Petri créé [18] :

- La disponibilité des composants et du système
- La fiabilité
- Le temps passé dans les différents états du système (Marche, Panne, Réparation ...)
- Les temps de remise en service après un certain événement,
- Les ressources consommées,
- La production du système ... etc[18].

Présentation de l'interface

Fenêtre principale du module Réseaux de Petri à prédicats[17]

La fenêtre principale est décomposée en plusieurs parties :

- **Barre de titre** : La barre de titre indique le nom du module et le nom du fichier en cours d'édition.

- **Barre de menu** : La barre de menu permet d'accéder à toutes les fonctions de l'application.
- **Barre d'icônes (raccourcis)** : La barre de raccourcis est une barre (horizontale) d'icônes permettant d'accéder plus rapidement aux fonctions usuelles.
- **Barre d'outils** : La barre d'outils (verticale) permet de sélectionner les éléments pour modéliser.
- **Zone de saisie** : Un maximum de place a été laissée à la zone de saisie graphique pour permettre de réaliser le modèle.
- **Modèle** : La liste des modèles se situent en dessous de l'arborescence. Ils sont groupés en deux sous dossier suivant leur lieu d'enregistrement (Répertoire utilisateur ou d'installation).
- **Arborescence** : L'arborescence est "cachée" entre la zone de saisie et la barre d'outil. Elle permet de naviguer dans les pages et groupes du document.
- **Ensemble des tableaux** : Les tableaux sont regroupés dans des onglets "cachés" à droite de la zone de saisie.

Panneau de simulation [18]

Le panneau de la simulation interactive est composé de quatre parties :

- Tout en haut, une barre d'outils regroupe les fonctionnalités permettant de lancer, stopper, configurer et jouer la simulation,
- Juste en dessous, vous est présenté un historique des transitions tirées,
- Puis vient l'échéancier qui contient la liste des transitions tirables ordonnées par date de tire,
- Tout en bas un panneau indique le temps courant et le cycle courant

RdP dans GRIF

Les RdPs sont faciles à construire via une interface graphique intuitive. Dans GRIF les transitions, les arcs et des jetons peuvent être créés facilement. Une fois que le système a été modélisé, le moteur Moca-PN (Monte Carlo Petri Nets) produit de nombreux résultats, par exemple [18] :

- Par rapport aux places : il nous donne le temps de séjours, le nombre moyen de

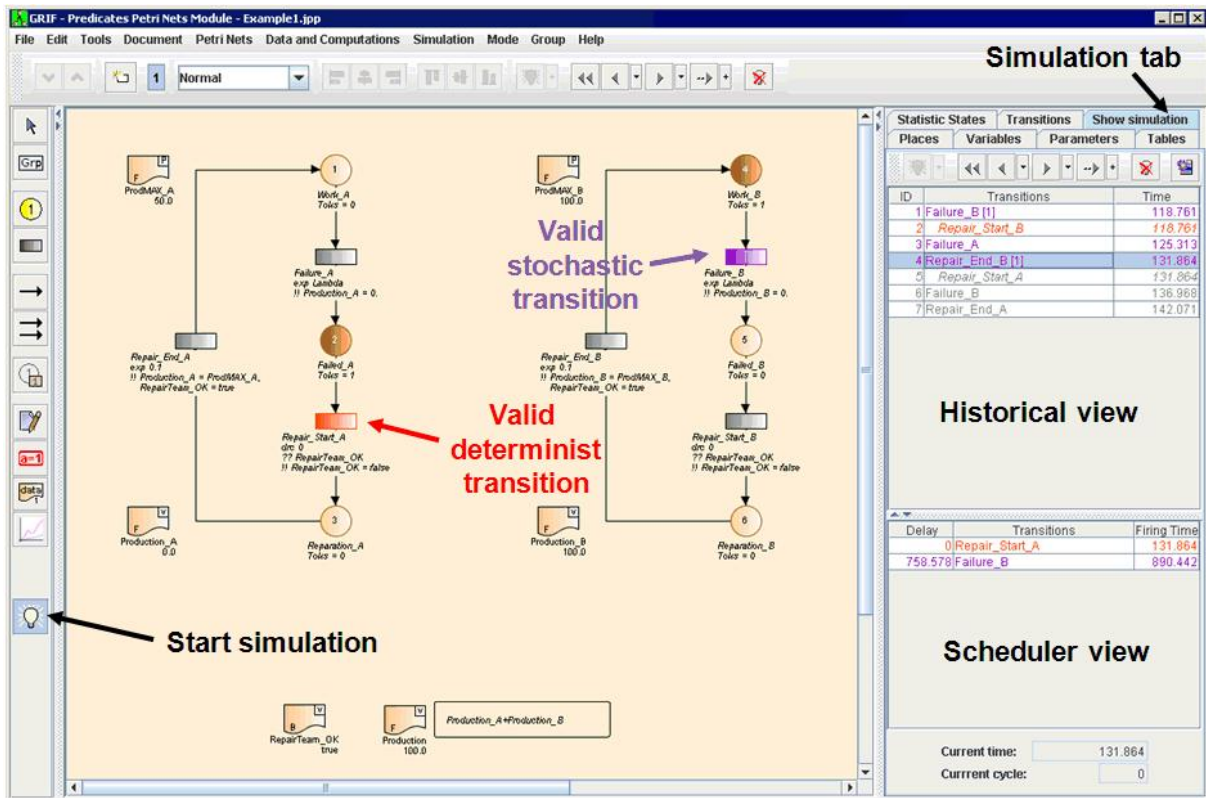


FIGURE 3.2 – Interface utilisateur graphique de GRIF

jetons qui passent dans ces places, il nous donne aussi le nombre moyen de jeton à la fin de l’histoire dans chaque places.

— Pour les transitions, il nous donne le nombre de franchissement de ces transitions.

Création d’un RdP avec GRIF

L’ensemble de symboles graphiques relatifs aux réseaux de Petri est représenté sur la barre d’icônes placées verticalement à gauche de la fenêtre de saisie.



FIGURE 3.3 – Barre d’icônes.

La barre d’outils verticale comporte les éléments suivants :

- Places représentées par des cercles ;

- Transitions représentées par des rectangles ;
- Arcs amont et aval représentés par des flèches ;
- Place répétée (ou Renvoi) pour réaliser des liaisons entre plusieurs parties du même modèle (sur des pages Ou dans des groupes différents) ;
- Commentaire pour ajouter du texte directement sur le graphique ;
- Affichage dynamique pour afficher une valeur d'un élément du modèle ;
- Variables locales pour créer des variables liées uniquement à une partie du modèle ;
- Courbe pour tracer des courbes représentant des calculs sur le modèle ;
- Simulation permettant de passer en mode simulation (mode animation).

Utilisation du Moca-PN

La fenêtre de paramétrage des calculs est accessible de deux manières différentes : soit par le menu Données et calculs, Données Moca soit par Données et calculs Lancer Moca.... La différence entre les deux est que dans le second cas, l'étape de paramétrage est directement suivie par l'étape de lancement des calculs. La fenêtre de paramétrage qui est ainsi ouverte est appelée Lancement des calculs Moca. Cette fenêtre de paramétrage est composée de plusieurs parties [17] :

1. **Titre** : permet de donner un titre au fichier résultat.
2. **Temps de calcul par défaut** :
 - Itération De A à B pas C : les calculs seront effectués pour des valeurs de t allant de A à B par pas de C.
 - Liste de temps : les calculs seront effectués pour les valeurs de t données dans cette liste.
 - Unité : les calculs sont effectués par défaut en heures. Il est possible de spécifier l'unité dans laquelle ont été saisis les temps de calcul. Les résultats seront toujours afficher en heures.
3. **Général** :
 - Nombre d'histoires : Nombre d'histoires (NH) à simuler ;
 - $1_{er}N$ au hasard : Graine du générateur de nombres aléatoires ;

- Temps de calcul maximum : Temps (en secondes) au bout duquel Moca arrêtera de simuler de nouvelles histoires ;
 - Durée automatique de l'histoire : Si cette case est cochée, GRIF va calculer la durée de l'histoire en fonction des temps de calcul de l'ensemble des variables et états statistiques. Sinon l'utilisateur peut spécifier la Durée d'une histoire ;
 - Calcul multi-processeurs : Permet d'activer le calcul multi-processeurs et d'indiquer le nombre d'instances Moca lancées.
4. **Variables :** L'onglet variables rappelle et permet de modifier la configuration du calcul pour chaque variable. Si le document contient des états statistiques, un onglet supplémentaire sera disponible.
 5. **Options de Sortie :** permet de paramétrer la sortie.
 6. On peut choisir la limite pour le nombre de tirs instantanés avant détection d'une boucle.

Résumé

Dans ce mémoire, nous modélisons et calculons les indices d'évaluation de performance d'un réseau de Petri stochastique, En utilisant la méthode simulation par le simulateur GRIF.

L'utilisation des RdPS présente a la fois l'avantage de la modélisation graphique et des résultats mathématiques. Les résultats obtenus par simulation nous ont permis une meilleure compréhension.

Mots clés : Evaluation des performances, Modélisation, GRIF, Simulation, RdPS.

Abstract

In this dissertation, we model and calculate the performance evaluation indices of a stochastic Petri net, using the GRIF simulator method.

The use of RdPS has both the advantage of graphical modeling and mathematical results. The results obtained by simulation have allowed us a better understanding.

Key words : Performance evaluation, Modeling, GRIF, Simulation, RdPS.

المخلص

في هذه الرسالة ، قمنا بنمذجة وحساب مؤشرات تقييم الأداء لشبكة بترى العشوائية باستخدام طريقة المحاكاة بواسطة محاكي GRIF .

يتمتع استخدام RdPS بميزة النمذجة الرسومية والنتائج الرياضية. سمحت لنا النتائج التي تم الحصول عليها عن طريق المحاكاة بفهم أفضل.

الكلمات الرئيسية : تقييم الأداء ، النمذجة ، GRIF ، المحاكاة ، RdPS .