

Hannouche
17/004.439

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 mai 1945 Guelma
Faculté des Mathématiques et de l'Informatique et des Sciences de la Matière



Mémoire de projet de fin d'étude

Master « IMD »

Département d'Informatique

Spécialité : Ingénierie des Media.



18/20

Thème:

**Conception et implémentation d'un environnement de
modélisation des systèmes en
RDPMaximalité**

Présenter par :

Mr. MAIZI Nouredine.

Meme.BELAAZ Fatima Zohra.

Sous la direction de :

Mr. BENAMIRA Adel.



- Juin 2012 -

Remerciement

*Nous remercions DIEU le tout puissant qui nous a donné la force,
La volonté et le courage pour accomplir ce modeste travail.*

Nous remercions vivement de tout cœur pour

✧ Mr Benamira Adel ✧

*pour avoir accepté nous encadrer et qui a été notre guide
et nous a pas privés de ses orientations, de ses conseils bénéfiques
et du temps que nous a consacré en dépit de toutes ses occupation.*

Nous exprimons notre profonde reconnaissance

À tous les enseignants de l'université de Guelma

8 mai 1945.

Nous remercions nos familles

Enfin nous remercions tous ceux qui contribué à ce travail par

Leurs remarques, leurs suggestions et leurs soutiens.

Nordine

Fatima

Dédicace

*Je commence ma dédicace au nom du dieu et par le bénisse et le salut
sur Mohamed le messenger de dieu.*

Je dédie ce travail en premier lieu à mes parents que dieu dit sur eux :

« ... و قضي ربك ألا تعبدوا إلا إياه و بالوالدين احسانا ... »

*Qui m'ont aidé durant ces années d'études par leurs soutiens et leurs
prières*

Pour moi que je les nomme « Soleil et Lune ».

Je dédie aussi à mes frères : Fateh, Fouzi, Fares.

A ma belle et la seule sœur dans ce monde : Samia.

A Houria et Nessma.

Ma famille : Maizi et Djebbar.

*A mon partenaire dans ce travail qui est comme une vraie sœur et qui
m'a beaucoup poussé pour travailler*

BelaaZ Fatima Zohra.

A tous nos collègues de la promotion 2012.

Pour tout qui n'ont pas été cité, je leurs dirai : mon cour vous n'oubliez

Jamais.

Noureddine

Dédicace

*Je commence ma dédicace au nom du dieu et par le bénisse et le salut
sur Mohamed le messenger de dieu.*

Je dédie ce travail en premier lieu à mes parents que dieu dit sur eux :

« ... و قضى ربك ألا تعبدوا إلا إياه و بالوالدين احسانا ... »

*Qui m'ont aidé durant ces années d'études par leurs soutiens et leurs
prières*

Pour moi que je les nomme « Soleil et Lune ».

Je dédie aussi à mes frères : Khayreddine, Bilel.

A ma belle et la seule sœur dans ce monde : Chaïma.

A mon marie Zineddine.

Ma famille : Belaaz et ziaya .

*A mon partenaire dans ce travail qui est comme un vrai frère et qui
m'a beaucoup poussé pour travailler*

Maizi Noureddine.

A tous nos collègues de la promotion 2012.

Pour tout qui n'ont pas été cité, je leurs dirai : mon cour vous n'oubliez

Jamais.

Fatima Zohra

Resumé

La vérification formelle des systèmes complexes constitue aujourd'hui un enjeu majeur dans de nombreux domaines de la société humaine, les méthodes formelles sont de plus en plus utilisées dans le but de la modélisation des systèmes concurrents.

Ces méthodes utilisent des modèles formels de spécification dotés de sémantiques bien définies.

Et parmi ces modèles et précisément les modèles sémantiques du parallélisme on trouve les réseaux de pétri.

D'autre part, le modèle des systèmes de transitions étiquetées maximales (STEMs) est un modèle de vrai parallélisme. Ce modèle a été utilisé comme modèle sémantique des algèbres de processus et des réseaux de pétri.

Tout ces théorique on été réaliser dans notre projet sous la plateforme JAVA avec les outils JGraphX et JDom.

Mots Clés : Vérification formelle, RDP, sémantique de maximalité, STEM, java, JGraphics, JDom.

Sommaire

Introduction générale	9
-----------------------------	---

Chapitre 01 : Vérification formelle.

I - Introduction	11
II – La vérification formelle	11
III - Fondement de la vérification formelle	12
III.1 - Un langage de description formelle du système	12
III.2 - Un modèle sémantique du parallélisme	12
III.3 - Un langage de spécification	13
III.4 - La vérification	13
IV - Les approche des méthodes de vérification formelle	14
IV.1 - Les vérifications basées sur les équivalences	14
IV.2 - Les vérifications basées sur le model-checking	15
V - Conclusion	16

Chapitre 02 : RDP et la sémantique de maximalité.

I - Introduction	17
II – Les réseaux de pétri	17
II.1 - Concepts de base	17
II.1.1 - Définition	17
II.1.2 - Exemples de réseau de pétri	19
II.1.3 - Représentation formelle	20
II.1.4 - Représentation graphique	21
II.1.5 - Représentation matricielle	22

II.2 - Fonctionnement de réseau de pétri	23
II.2.1 - Transition validée	23
II.2.2 - Franchissement ou tir	23
II.2.3 - Séquence de Franchissement	25
II.3 - Propriétés de réseau de pétri	26
II.3.1 - Conflits et parallélisme	26
II.3.2 - Réseau propre (réinitialisable)	27
II.3.3 - Réseau vivant (sans blocage)	27
II.3.4 - Réseau borné	27
II.3.5 - Réseau conforme	28
II.3.6 - Machine à états finie	28
II.4 - Réseaux de pétri avec une structure particulière	29
II.5 - Les familles et les extensions de RdP	31
II.5.1 - Réseaux de pétri autonomes	32
II.5.2 - Réseaux de pétri non autonomes	33
III - La sémantique de maximalité	36
III.1 - Principe de la sémantique de maximalité	36
III.2 - Sémantiques autour de la maximalité	37
IV - Réseau de pétri via la sémantique de maximalité	37
IV.1 - Liaison des jetons	40
IV.2 - Successions de franchissement	40
IV.3 - Identification des jetons consommés	42
V - Conclusion	43

Chapitre 03 : Conception.

I - Introduction	44
II - Conception	44
II.1 - Conception de l'éditeur graphique	45
II.1.1 - Creation de RDP	45
II.1.2 - L'ouverture et l'enregistrement d'un RDP	46
II.1.3 - Traitement d'un RDP	47
II.2 - Conception du simulateur	48
II.2.1 - Vérification pour franchissement	48
II.2.2 - Franchissement et mettre à jour	50
III - Conclusion	50

Chapitre 04 : Etude des cas.

I - Introduction	51
II - Réalisation	51
II.1 - L'interface de l'éditeur	51
II.2 - Editeur	52
II.3 - Quelques opérations sur les éléments graphiques	55
III - Etude des cas	57
III.1 - Le modèle quatre saisons	58
III.2 - Le modèle de la machine à remplir et à boucher	59
III.2.1 - Description	59
III.2.2 - Représentation et simulation de modèle	61
III.3 - Le Problème des philosophes	64
III.3.1 - Description	64
III.3.2 - Représentation et simulation de modèle	65
IV- Conclusion	67

Conclusion Générale	68
----------------------------------	----

La bibliographie	69
-------------------------------	----

Liste des Figures:

Figure 1.01 : Vérification formelle	12
Figure 2.01 : Méthode générale de modélisation et d'analyse basée sur les réseaux de pétri	19
Figure 2.02 : Réseau de pétri pour le système de quatre saisons	19
Figure 2.03 : Réseau de pétri pour le système de producteur- consommateur ..	19
Figure 2.04 : Exemple de Graphe (gauche) et Graphe orienté (droite)	21
Figure 2.05 : Représentation graphique d'un réseau de pétri	21
Figure 2.06 : Transition source	21
Figure 2.07 : Transition puits	21
Figure 2.08 : Transition validée (gauche), non validée (droite)	23
Figure 2.09 : Franchissement d'une transition	23
Figure 2.10 : Exemple de l'atelier de coupe	24
Figure 2.11 : Exemple d'évolution d'un RdP	24
Figure 2.12 : Séquence de franchissement	25
Figure 2.13 : Le conflit structurel	26
Figure 2.14 : Le conflit relatif au marquage	26
Figure 2.15 : Le parallélisme sur RdP	26
Figure 2.16 : La vivacité dans les RdP	27
Figure 2.17 : RdP Borné	28
Figure 2.18 : Graphe d'états ou pas	29
Figure 2.19 : Graphe d'événements ou pas	29
Figure 2.20 : Conflit ou pas	29
Figure 2.21 : RdP qui n'est pas simple	30
Figure 2.22 : RdP impur équivalent à un RdP pur	30
Figure 2.23 : Les familles de RdP	31
Figure 2.24 : RdP à arc étiquetés	32
Figure 2.25 : RdP interprété	32
Figure 2.26 : Exemple d'un réseau de pétri marqué	38
Figure 2.27 : Jetons libres et jetons liés dans un marquage	39
Figure 2.28 : Franchissement de la transition t_3	39
Figure 2.29 : Liaison des jetons	40

Figure 2.30 : Successions de franchissements de t1	40
Figure 2.31 : Réseau de pétri avec arc sortant de poids supérieur à un	41
Figure 2.32 : Noms d'événements identifiant les jetons consommés	42
Figure 3.01 : Conception de notre système.....	44
Figure 3.02 : Création de RDP	45
Figure 3.03 : L'ouverture et l'enregistrement de RDP	46
Figure 3.04 : Traitement de RDP.....	47
Figure 3.05 : La Simulation	48
Figure 3.06 : La vérification	49
Figure 3.07 : Colorisation des transitions	49
Figure 3.08 : Franchissement des transitions	50
Figure 4.01 : L'interface de l'éditeur	51
Figure 4.02 : La barre d'outils	52
Figure 4.03 : Création des places	53
Figure 4.04 : Création des transitions	53
Figure 4.05 : Création des arcs IN	54
Figure 4.06 : Création des arcs OUT	54
Figure 4.07 : Suppression d'une place	55
Figure 4.08 : Suppression d'une transition	55
Figure 4.09 : Suppression d'un arc	55
Figure 4.10 : Modification d'une place	56
Figure 4.11 : Modification d'une transition	56
Figure 4.12 : Modification d'un arc	56
Figure 4.13 : Information d'une place	57
Figure 4.14 : Validation du nom de place et transition	57
Figure 4.15 : Exemple d'un RDP de quatre saisons	58
Figure 4.16 : Simulation de l'exemple de quatre saisons	58
Figure 4.17 : Développement de la simulation de l'exemple de quatre saisons.....	59
Figure 4.18 : La machine à remplir et à boucher	60
Figure 4.19 : RDP machine à remplir et à boucher	60

Figure 4.20 : La représentation graphique de modèle	61
Figure 4.21 : La simulation de modèle	61
Figure 4.22 : Franchissement de T1	62
Figure 4.23 : Franchissement de T2	62
Figure 4.24 : Franchissement de T3	63
Figure 4.25 : Franchissement de T4	63
Figure 4.26 : Le problème des philosophes	64
Figure 4.27 : RDP d'un seul philosophe	64
Figure 4.28 : RDP de trios philosophes	65
Figure 4.29 : Simulation de RDP de trios philosophes	65
Figure 4.30 : Franchissement de T1 «le premier philosophe mange»	66
Figure 4.31 : Franchissement de T2 «le deuxième philosophe mange»	66

Liste des abréviations:

BT: Bound Tokens « jeton liée ».

FT: Free Tokens « jeton libre ».

RDP : Réseau De Pétri.

RdPS : Réseaux De Pétri Stochastiques.

XML : Extensible Markup Language.

MIT : Massachussetts Institute Of Technology.

STE : Système de Transition Etiquetée.

STEM: Systèmes de Transitions Etiquetées Maximale.

Introduction Générale

Le développement de systèmes informatiques complexes nécessite une conception préalable qui puisse garantir au mieux que le système que l'on construit satisfera les contraintes nécessaires à son bon fonctionnement. Pour ce faire, on commence par spécifier formellement le système.

Les spécifications formelles restent difficiles à écrire car les systèmes à développer actuellement sont d'une assez grande complexité. Il est donc nécessaire de proposer des méthodes facilitant la démarche d'analyse et de spécification des systèmes complexes.

Les réseaux de pétri ont été développés en tant que formalisme opérationnel pour la spécification des systèmes concurrents. Ils donnent une représentation dynamique d'un état du système par l'utilisation de jetons. (Système de transition étiqueté « STE »).

Le modèle des systèmes de transitions étiquetées maximales « STEMs » est un modèle de vrai parallélisme. Ce modèle a été utilisé comme modèle sémantique des algèbres de processus et des réseaux de pétri.

L'objectif de ce projet est de développer une interface graphique pour réalisation d'un réseau de pétri et aussi la conception et l'implémentation des ces réseaux avec la sémantique de maximalité.

Pour faciliter la lecture de ce mémoire, on l'a divisé en quatre chapitres :

- ❖ **Le premier chapitre :** on peut dire que ce chapitre est un chapitre introduction ou on a parlé de la vérification formelle son fondement et ces méthodes.

- ❖ **Le deuxième chapitre :** dans ce chapitre nous avons donné une vue globale sur les réseaux de pétri concernant la définition, concept de base et ces différentes propriétés et aussi on parler de la sémantique de maximalité son principe et aussi d'appliquer les différentes sémantique sur des exemples.
- ❖ **Le troisième chapitre :** ce chapitre concerne la conception, ou on a présenté les différentes fonctionnalités sous forme des boites noires et détaillé chaque une.
- ❖ **Quatrième chapitre :** ce chapitre est une présentation de l'utilisation et aussi une étude des cas ou on a testé notre application sur des différents exemples.

Nous finissons cette mémoire avec une conclusion général qui résume tous les résultats obtenu lors de ce travail.

Ce mémoire.



Chapitre 01

Vérification formelle

I - Introduction :

La vérification formelle des systèmes complexes constitue aujourd'hui un enjeu majeur dans de nombreux domaines de la société humaine. En effet, l'employant des méthodes de spécification et vérification formelle, assistées par des outils informatiques performants rendre l'analyse de ces systèmes fiable et garantie, en plus, un bon compromis coût performances est assuré. La principale limitation de la vérification formelle basée sur les modèles est connue sous le nom de l'explosion combinatoire du graphe d'états. De multiples travaux sont en cours en vue de maîtriser cette explosion.

II - La vérification formelle: [BEL10]

Durant la dernière décennie, les méthodes formelles sont devenues une activité indispensable intégrée au processus de conception des systèmes complexes à caractère critique, telles que les protocoles de télécommunication, les systèmes répartis et les architectures multiprocesseurs.

En effet, la complexité de ces systèmes rendant leur analyse classique (prototypage et test) extrêmement ardue, leur fiabilité ne saurait être garantie autrement qu'en employant des méthodes de spécification et vérification formelle, assistées par des outils informatiques performants.

Une approche de vérification offrant un bon compromis coût performances est la vérification basée sur les modèles (model-checking), connue aussi sous le nom de la vérification énumérative.

Cette approche consiste à traduire le système, préalablement d'écrite dans un langage appropriée « Description du système », vers un modèle « sémantique », généralement un graphe d'états, sur le quel les propriétés de correction attendues «Spécification» sont vérifiées au moyen d'algorithmes spécifiques (Figure1.01).

Bien que limitée aux applications ayant un nombre fini d'états, la présente approche est particulièrement utile dans les premières phases du processus de conception, permettant une détection rapide et économique des erreurs.

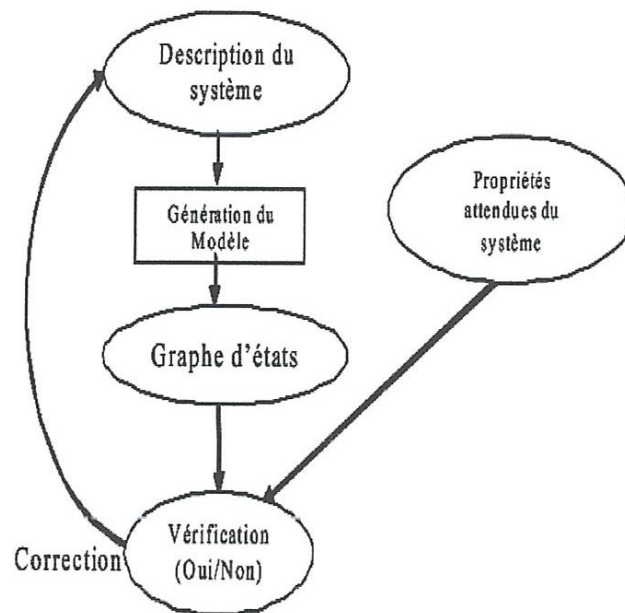


Figure 1.01 : Vérification formelle.

III - Fondement de la vérification formelle : [BEL10]

La vérification formelle fondée précisément sur les quatre éléments suivants :

III.1 - Un langage de description formelle du système :

C'est un formalisme du comportement de systèmes de haut niveau, il possède à la fois une syntaxe et une sémantique bien définie (dit formel). Parmi ces langages, nous pouvons citer les réseaux de pétri, les algèbres de processus (ACP [BK85], CCS [Mil80] [Mil83] [Mil89], CSP [Hoa85],...) et les techniques de description formelle (ESTELLE [90788], LOTOS [BB87] [ISO88], SDL [CCI88]).

III.2 - Un modèle sémantique du parallélisme (modèle de bas niveau) :

Comme son nom l'indique, ce modèle est utilisé pour exprimer la sémantique du parallélisme des langages de description formelle. Nous distinguons deux grandes familles, les modèles

d'entrelacement (Arbres de synchronisation [Mil80], STE [Arn92],...) et les modèles de non entrelacement (dit vrai parallélisme) (RDP [Rei85], SEP [NPW81], STA [Bed87][Shi85a]...)

Leur principale différence réside dans l'adoption ou non de l'hypothèse de l'atomicité temporelle et structurelle des actions (les actions sont de durée nulle et indivisibles).

Les modèles de vrai parallélisme nous permettent d'éviter les difficultés liées aux modèles entrelacés d'une part, et d'une autre part nous permettent d'introduire une méthode de conception formelle par raffinement successif (remplacer une action par un processus).

III.3 - Un langage de spécification :

C'est un formalisme dédié à la description de propriétés attendues du système. Nous trouvons dans la littérature deux types de langages de spécification : spécification logique et spécification comportementale. La spécification logique (HML [HM80], CTL [CE81] [CES83], CTL.* [FH86], ...) est généralement fondée sur des formules d'une logique temporelle modale qui sont interprétées sur le modèle sémantique sous-jacent. Cependant les propriétés du deuxième type sont exprimées dans le même modèle sémantique de celui de la description du système.

III.4 - La vérification :

C'est une relation de satisfaction qui définit la comparaison entre la description du système et sa spécification. Selon le formalisme de la spécification employé, il existe généralement deux types de vérification; le premier type consiste de vérifier les formules logiques sur le graphe d'états du système (dit vérification logique) et l'autre type est une comparaison entre les modèles sémantiques s'effectue au moyen d'une relation d'équivalence ou de préordre [Mil80] [Par81] (dit vérification comportementale).

Pour prévenir ainsi d'éventuels dysfonctionnements, une pléthore de technique ont été développées. Parmi celles-ci, les tests et la simulation sont les plus connus et probablement les plus largement utilisés. Dans les vérifications basées sur les tests, le système à vérifier est testé pour révéler d'éventuelles erreurs sur un ensemble de situations choisies. Faute de temps, il est souvent impossible de tester ce système pour toutes les situations possibles. L'ensemble des cas à tester doit être choisi avec une attention particulière pour couvrir le

maximum de scénarios distincts possible. Néanmoins, il est impossible de garantir que le produit sera exempt de toute erreur.

Une autre technique utilisée pour assurer la validation de systèmes est la méthode formelle. Pour pouvoir être réalisée, la vérification impose une description formelle du système, ainsi qu'une spécification formelle des propriétés.

De nombreux formalismes de spécification dédiés aux systèmes concurrents ont été proposés en littérature tels que les automates communications, systèmes de transitions, algèbres de processus.

Parmi les formalismes les plus utilisés, les réseaux de pétri occupent une place prépondérante et se trouvent être l'outil ayant été le plus largement étudié si l'on considère la diversité des techniques automatiques de vérification qui leur sont associées. Ils offrent une structure très simple et forment toujours le support de nombreuses études théoriques des systèmes concurrents et de leurs propriétés.

IV- Les approches des méthodes de vérification formelle : [BEL10]

Parmi les méthodes de vérification formelle, nous pouvons distinguer ces deux approches suivantes :

IV.1 - Les vérifications basées sur les équivalences :

Cette approche consiste à vérifier l'équivalence (le plus souvent par rapport à une relation de bisimulation) entre le modèle de description de l'implémentation et une spécification qui décrit ce que l'on attend de cette implémentation, les deux utilisant le même formalisme de description. Si les deux modèles sont équivalents cela prouve que l'implémentation est conforme à la conception.

IV.2 - Les vérifications basées sur le model-checking :

Cette dernière approche et la plus utilisées, est basée sur les modèles, permet une vérification simple et efficace et elle est complétement automatisable. La vérification basée sur les modèles ou model-checking est surtout applicable pour les systèmes ayant un espace d'états fini.

Les algorithmes de vérification dans cette méthode utilisent l'ensemble des états que le système peut atteindre pour prouver la satisfaction ou la non-satisfaction des propriétés. Cependant, le problème majeur de ce type de vérification est la taille souvent excessive de l'espace d'états. En effet elle peut être exponentielle par rapport à la taille de la description du système. Une des causes principales de cette explosion combinatoire du nombre d'états et le fait que l'exploration est réalisée en prenant en compte tous les entrelacements possible d'événement concurrents.

Pour pallier à ce problème de l'explosion combinatoire, différentes solutions dont l'objectif est de réduire la taille de l'espace d'états, ont été proposées dans la littérature. Elles sont généralement basées sur :

- L'utilisation de structure de données particulières.
- L'exploitation de l'ordre partiel sur les occurrences des événements.
- L'exploitation de symétries du système.
- L'exploitation de la modularité.

V - Conclusion :

On a parlé dans ce chapitre des différentes méthodes et formalismes des vérifications formelles et parmi les formalismes les plus utilisés on trouve les réseaux de pétri qui dispose d'une définition formelle qui permet de produire des spécifications exemptes d'ambiguïté, chaque construction des modèles possède une sémantique parfaitement définie et aussi les systèmes de transitions étiquetées dispose de plusieurs avantage mais malgré tout ca il n'offre pas un vrai parallélisme pour cela ils ont développé un autre outil qui sont les systèmes de transitions étiquetées maximale STEMs qui ouvre un vrai parallélisme garce a la sémantique de maximalité qui on va la détaille dans le chapitre suivant.



Chapitre 02

RDP et la sémantique de maximalité

I - Introduction :

Ce chapitre est divisé sur trois parties principales dans la première on va parler des réseaux de pétri comme la définition, ces différentes représentations, ces propriétés et aussi ces différentes structure et dans la deuxième partie on va parler sur la sémantique de maximalité ces différentes formules et son principe, et la dernière partie c'est la partie de RDP via la sémantique de maximalité c'est la partie la plus importantes dans notre travail ou on va étudié des différentes exemples et les règles générales des franchissements et des différents traitements.

II - Les réseaux de pétri :

II.1 - Concepts de base :

Historiquement le réseau est présenté par Carl Adam pétri dans sa thèse "Communication avec des Automates" en Allemagne _a Bonn en 1962. Ce travail a continue à être développé par Anatol W. Holt, F. 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 pétri et des méthodes relationnels ont été organisés à MIT. En 1981 le premier livre du réseau de pétri a été publié en Anglais par J. Peterson. Aujourd'hui, suivre Pétri-Net Newsletter, chaque année il y a des 600 aux 800 d'œuvres des réseaux de pétri sont publiés [Tra05].

II.1.1 - Définition :

Le réseau de pétri est un outil graphique et mathématique pour modéliser et analyser les systèmes discrets, particulièrement les systèmes concurrents, parallèles, non-déterministes, etc. En étant rôle d'outil graphique, il nous aide à comprendre facilement le système modélisé, et plus il nous permet de simuler les activités dynamiques et concurrentes. Avec le rôle d'outil mathématique, il nous permet d'analyser le système modélisé grâce aux modèles de graphes, aux équations algébriques, etc.

Informellement, le réseau est un graphe biparti ayant 2 types des nœuds. Certains étant les transitions représentent les actions élémentaires du système, et les autres étant les places représentent ses ressources. Pour modéliser et analyser les applications industrielles, qui sont de plus en plus compliqués, plusieurs types de réseau sont représentés. Selon la proposition des Monika Trompedeller en 1995, les réseaux sont classifiés en 3 niveaux [Tra05]:

Niveau 01: Les réseaux caractérisés par les places qui peuvent représenter des valeurs booléennes. Un exemple de ce type est les Réseaux de Condition/Événement.

Niveau 02: Les réseaux sont caractérisés par les places qui peuvent représenter des valeurs de nombre entier. Un exemple de ce type est les Réseaux de Place/Transition.

Niveau 03: Les réseaux sont caractérisés par les places qui peut représenter des valeurs a niveau élevé, c'est-à-dire, une place est marquée par un multi ensemble de jetons structurés. Un exemple de ce type est les Réseaux colorées.

Le formalisme des RdPs a été proposé comme un outil mathématique permettant la modélisation des systèmes dynamiques à événements discrets. Ce formalisme bénéficie d'une riche panoplie de résultats théoriques, de techniques d'analyse et d'outils. L'analyse d'un réseau de Pétri peut révéler des caractéristiques importantes du système concernant sa structure et son comportement dynamique. Les résultats de cette analyse sont utilisés pour évaluer le système et en permettre la modification ou l'amélioration.

La figure 2.01 montre la méthode générale basée sur le formalisme des réseaux de pétri pour la modélisation et l'analyse des systèmes.

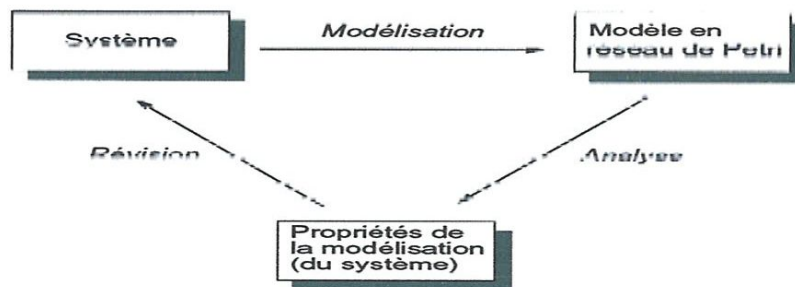


Figure 2.01 : Méthode générale de modélisation et d'analyse basée sur les réseaux de pétri.

II.1.2 - Exemples de réseau de pétri :

❖ Système de quatre saisons :

On considère un système d'écrivain quatre saisons. S'il est printemps et l'événement "début d'été" se produit, la saison changea l'été. Ensuite, si l'événement "début de automne" se produit, la saison est automne, etc. Le système modélise le changement séquentiel entre les saisons [Tra05].

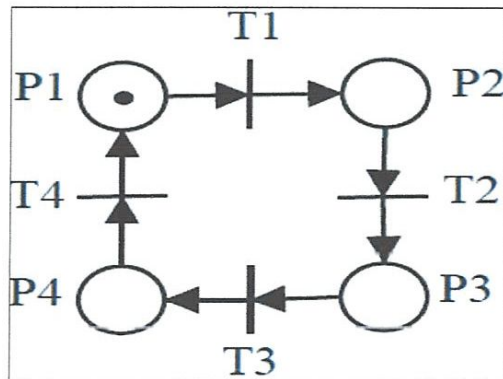


Figure 2.02 : Réseau de pétri pour le système de quatre saisons.

❖ Système de producteur-consommateur :

On décrit le système d'un producteur et un consommateur utilisant un tampon comme leur boutique pour échanger les items (ou les marchandises). Le producteur produit des items (représenté par des jetons dans le tampon). Le consommateur peut les acheter en supprimant des items dans le tampon. Un franchissement d'une transition supprime un jeton de l'entrée et ajoute un jeton à chaque place de sortie à la fois. Cela correspond à échangeement des marchandises dans la réalité [Tra05].

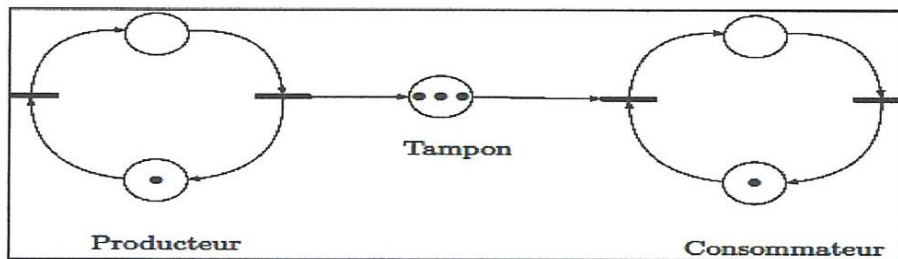


Figure 2.03 : Réseau de pétri pour le système de producteur- consommateur.

II.1.3 - Représentation formelle :

Un réseau de pétri (RDP) est la donnée d'un ensemble fini de places, d'un ensemble fini de transitions et d'une fonction dite fonction de poids. Ceci définit la structure statique du système. L'état de celui-ci se modélise à l'aide d'un marquage que l'on fait évoluer en franchissant des transitions, ce qui correspond à exécuter les actions qui leur sont associées. Formellement, un réseau de pétri peut être représenté par un quadruplet $R = (P, T, W, m_0)$ ou :

- ✓ P est un ensemble fini non vide de places : $P = \{p_1, p_2, \dots, p_n\}$
- ✓ T est un ensemble fini non vide de transitions. $T = \{t_1, t_2, \dots, t_n\}$
- ✓ $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ est la fonction de poids.
- ✓ $m : P \rightarrow \mathbb{N}$ est la fonction de marquage.

Le marquage initial est donné par m_0 . $m(p) = k$ signifie que la place contient k marques (jetons). On dit aussi que le marquage de p est k .

$W(p, t) = k$ signifie que la transition t utilise k jetons dans la place p .

De façon duale, $W(t, p) = k$ signifie que la transition t crée k jetons dans la place p .

[Jarras95]

Exemple :

Soit le réseau de pétri $R = (P, T, W, m_0)$ ou :

$P = \{p_1, p_2, p_3\}$ et $T = \{t_1, t_2, t_3, t_4\}$

Définissons W et m_0 de la façon suivante :

$W(p_1, t_2) = 1$. $W(p_2, t_1) = 1$. $W(p_2, t_3) = 3$. $W(p_3, t_4) = 1$.

$W(t_1, p_1) = 1$. $W(t_2, p_2) = 1$. $W(t_3, p_3) = 1$. $W(t_4, p_2) = 3$.

$m_0(p_1) = 0$; $m_0(p_2) = 3$; $m_0(p_3) = 0$:

Pour tous les couples (x, y) non spécifiés ci-dessus, $W(x, y) = 0$.

Par exemple : $W(p_1, t_3) = W(t_3, p_1) = 0$.

II.1.4 - Représentation graphique :

Un graphe peut être défini par un ensemble d'éléments appelés nœuds ou sommets et un ensemble de relations appelées arrêtes ou arcs. Un exemple est donné dans la figure les nœuds sont constitués par les carrés étiquetés A, B, C et D; les arcs sont constitués par les droites étiquetées 1 (qui relie le sommet A au sommet B), 2 (qui relie le sommet A au sommet C), 3 (qui relie le sommet B au sommet C) et 4 (qui relie le sommet C au sommet D). Les arcs peuvent être orientés : on parle alors de graphe orienté. Les sommets peuvent être de plusieurs types [GG06].

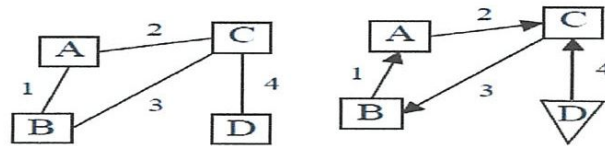


Figure 2.04 : Exemple de Graphe (gauche) et Graphe orienté (droite).

Un Réseau de pétri est un graphe orienté comprenant deux types de sommets :

- ✓ les places sont représentées par des cercles. ○
- ✓ les transitions par des rectangles (ou traits). □ ou —
- ✓ Ils sont reliés par des arcs orientés → : Un arc relie soit une place à une transition, soit une transition à une place mais jamais une place à une place ou une transition à une transition.

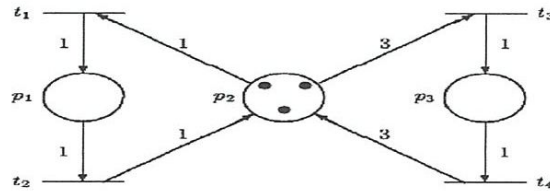


Figure 2.05 : Représentation graphique d'un réseau de pétri. [Jarras95]

✚ Cas particuliers :



Figure 2.06 : Transition source.

« pas de place en entrée de la transition »



Figure 2.07 : Transition puits.

« pas de place en sortie de la transition »

II.1.5 - Représentation matricielle : [ElMansouri]

Une représentation matricielle d'un RdP est offerte afin de simplifier les tâches d'analyse et de vérifications effectuées sur un modèle RdP. Agir sur une représentation graphique d'un modèle RdP est une tâche délicate en le comparant avec une représentation matricielle.

Soit Un réseau de pétri $R = (P, T, W)$ avec $P = \{p_1, p_2, \dots, p_m\}$ et $T = \{t_1, t_2, \dots, t_n\}$.

On appelle matrice des pré-conditions *pré*, la matrice $m \times n$ à coefficients dans N telle que $pré(i, j) = W(p_i, t_j)$ indique le nombre de marques que doit contenir la place p_i pour que la transition t_j devienne franchissable.

De la même manière on définit la matrice des post-conditions *post*, la matrice $n \times m$ telle que $post(i, j) = W(t_j, p_i)$ contient le nombre de marques déposées dans p_i lors du franchissement de la transition t_j . La matrice $C = post - pré$ est appelée matrice d'incidence du réseau (m représente le nombre de places d'un réseau de Pétri et n le nombre de transitions.)

Le marquage d'un réseau de Pétri est représenté par un vecteur de dimension m à coefficients dans N . La règle de franchissement d'un réseau de pétri est définie par :

$$M'(p) = M(p) + C(p, t).$$

Exemple :

La représentation matricielle du réseau de la figure 2.05 est comme suit :

$$\begin{array}{l}
 \begin{array}{c} \text{pré} \\ \text{pré} \end{array} = \begin{array}{c} \begin{array}{cccc} & t_1 & t_2 & t_3 & t_4 \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \\
 \end{array}
 \qquad
 \begin{array}{c} \text{post} \\ \text{post} \end{array} = \begin{array}{c} \begin{array}{cccc} & t_1 & t_2 & t_3 & t_4 \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \\
 \end{array}
 \end{array}$$

$$C = \begin{array}{c} \begin{array}{cccc} & t_1 & t_2 & t_3 & t_4 \\ \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} & \begin{pmatrix} 1 & \Leftrightarrow 1 & 0 & 0 \\ \Leftrightarrow 1 & 1 & \Leftrightarrow 3 & 3 \\ 0 & 0 & 1 & \Leftrightarrow 1 \end{pmatrix} \end{array} \\
 \end{array}$$

Le marquage initial :

$$m_{ii} = \begin{array}{c} \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}
 \end{array}$$

II.2 - Fonctionnement de réseau de pétri :

L'évolution d'un RdP correspond à l'évolution des on marquage (évolution de état du système): il sera duit par un déplacement de marques ce qui s'interprété comme la consommation/production de ressources déclenchée par des événements sou des actions.

Déterminer l'évolution d'un RdP correspond en fait à le simuler, terme plus généralement utilisé en modélisation.

II.2.1 - Transition validée :

Une transition est dite validée si toutes les places en amont (c'est-à-dire en entrée) de celle-ci possèdent au moins une marque. Une transition source est par définition toujours validée.

Deux exemples sont présentés dans la Figure 2.08. [GG06].

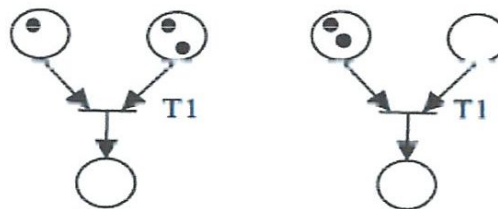


Figure 2.08 : Transition validée (gauche), non validée (droite).

II.2.2 - Franchissement ou tir :

Si la transition est validée, on peut effectuer le franchissement de cette transition: on dit alors que la transition est franchissable. Le franchissement consiste à :

- retirer une marque dans chacune des places en entrée de la transition;
- ajouter une marque a chacune des places en sortie.



Figure 2.09 : Franchissement d'une transition.

Lorsqu'une transition est validée, ce la n'implique pas qu'elle sera immédiatement franchie. Cela veut dire que les conditions nécessaires à son franchissement sont effectivement réunies: l'action et/ou l'événement associé à la transition sont alors possibles. Dans l'exemple de l'atelier de coupe (Figure 2.10), pour que la machine de coupe commence la commande (franchissement de la transition T2), il est nécessaire que la machine soit disponible (une marque dans la place P1) et qu'au moins une commande soit en attente (au moins une marque dans la place P2). Cela ne veut pas dire que la machine va commencer à traiter immédiatement la commande.

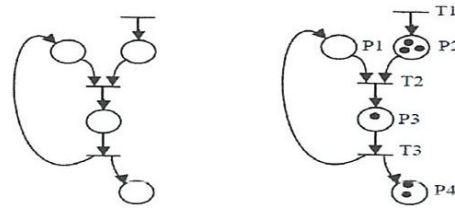


Figure 2.10 : Exemple de l'atelier de coupe.

L'évolution du RdP se fait par le franchissement d'une seule transition à la fois. Quand plusieurs transitions sont simultanément franchissables, on ne peut pas savoir dans quel ordre elles seront effectivement franchies. L'évolution n'est donc pas unique [GG06].

Et voila un exemple d'évolution d'un réseau de pétri :

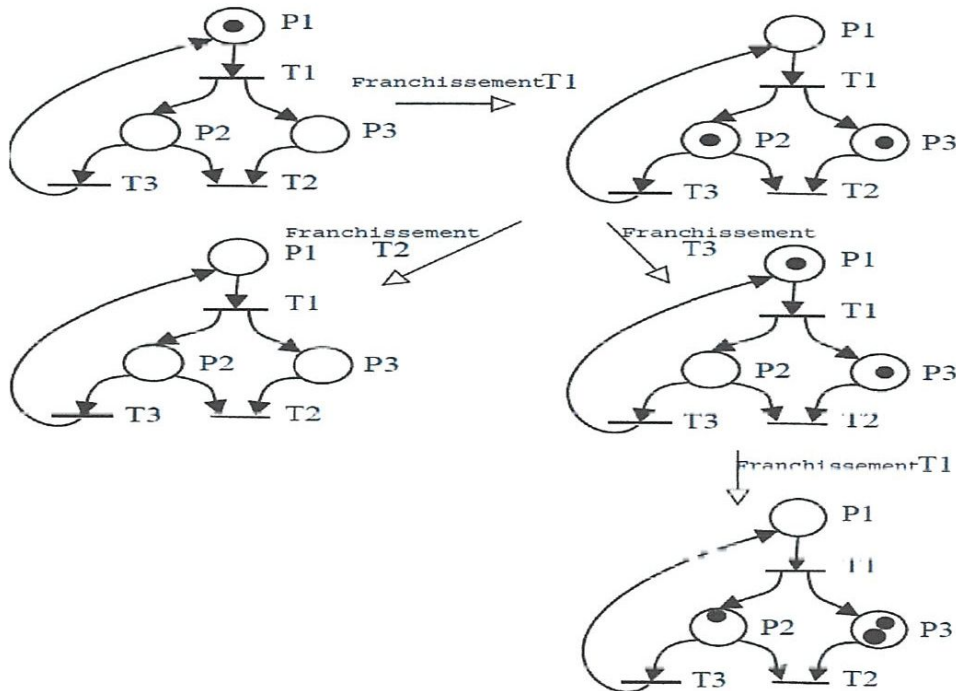


Figure 2.11 : Exemple d'évolution d'un RdP.

II.2.3 - Séquence de Franchissement :

Une séquence de franchissement S est une suite de transitions $T_i T_j \dots T_k$ qui peuvent être franchies successivement à partir d'un marquage donné. Une seule transition peut être franchie à la fois.

On note $M_i [S \rightarrow M_j$: ou $M_i [S > M_j$ à partir du marquage M_i , le franchissement de la séquence S aboutit au marquage M_j . Dans la figure suivante on a $T_1 T_2$ et $T_1 T_3$ sont deux séquences de franchissement :

$$M_0 [T_1 T_2 \rightarrow M_1 \quad \text{et} \quad M_0 [T_1 T_3 \rightarrow M_2 \quad \text{avec} \quad M_1 = [0010]^T \quad \text{et} \quad M_2 = [0001]^T$$

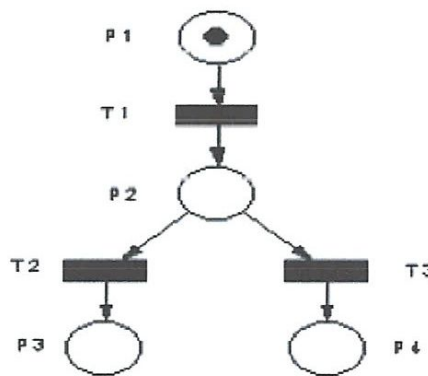


Figure 2.12 : Séquence de franchissement.

II.3 - Propriétés de réseau de pétri : [Geffroy]

II.3.1 - Conflits et parallélisme :

A. Structure potentiellement conflictuelle: $t1$ et $t2$ sont en conflit structurel potentiel pour le partage des jetons de la place p .

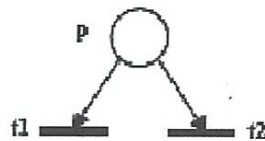


Figure 2.13 : Le conflit structurel.

B. Conflit relatif au marquage: Structure potentiellement conflictuelle + marquage insuffisant $t1$ et $t2$ en conflit pour le partage du jeton : non déterminisme du tir ($t1$ ou $t2$).

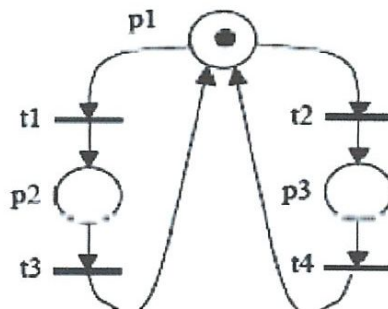


Figure 2.14 : Le conflit relatif au marquage.

C. Parallélisme :

Les deux transitions $t1$ et $t2$ sont tirées en parallèle.

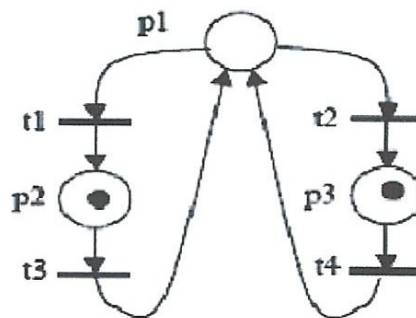


Figure 2.15 : Le parallélisme sur Rdp.

II.3.2 - Réseau propre (réinitialisable) :

SSI \forall marquage M_i accessible depuis M_0 , \exists une séquence de tirs conduisant à M .

II.3.3 - Réseau vivant (sans blocage) :

$M_0 \rightarrow M_k$

$\forall t_i \exists$ séquence de tirs passant par t_i

$M_0 = [1 \ 1 \ 0 \ 0]$

$M_1 = [0 \ 0 \ 1 \ 1]$

$M_2 = [0 \ 1 \ 2 \ 0] \rightarrow$ blocage ! (Voir Figure 2.16)

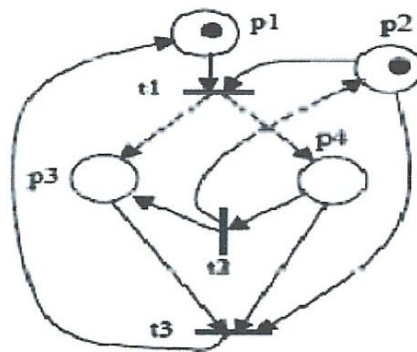


Figure 2.16 : La vivacité dans les Rdp.

II.3.4 - Réseau borné :

Si \forall marquage M_i accessible depuis M_0 , et $\forall p_j \in P \rightarrow M_i(p_j) \leq \text{MAX}$.

Marquages successifs :

$M [0 \ 0 \ 0 \ 1]$

$M [1 \ 1 \ 1 \ 0]$

$M [2 \ 0 \ 1 \ 1]$

$M [4 \ 1 \ 2 \ 0]$

M [5 1 2 1]

M [6 1 3 1] ... croissance infinie

Si MAX = 1 → réseau de pétri Sauf.

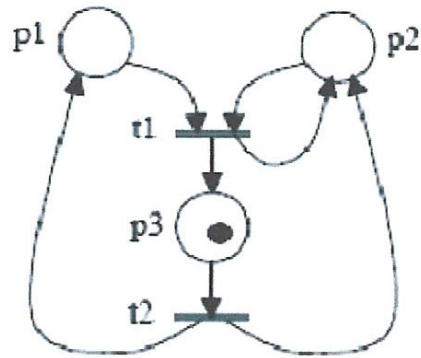


Figure 2.17 : RdP Borné.

II.3.5 - Réseau conforme :

Réseau de pétri vivant et sauf : 1 jeton, sans blocage.

II.3.6 - Machine à états finie :

Chaque transition n'a qu'une place amont et une place aval.

II.5 - Les familles et les extensions des RdPs : [Diaz01]

Trois grandes familles (qui ne sont évidemment pas étanches) peuvent se distinguer. Elles correspondent à trois domaines sémantiques : la sémantique discrète (RdP PT et ses dérivés) pour les comportements qui peuvent se représenter par des graphes finis ou dénombrables d'états. La sémantique continue (RdP T et ses dérivés) pour les comportements qui nécessitent la prise en considération explicite d'un temps dense; la sémantique stochastique (RdP S et ses dérivés) pour les comportements qui incluent des distributions de franchissement et conduisent à des processus stochastiques (chaînes de Markov, etc.).

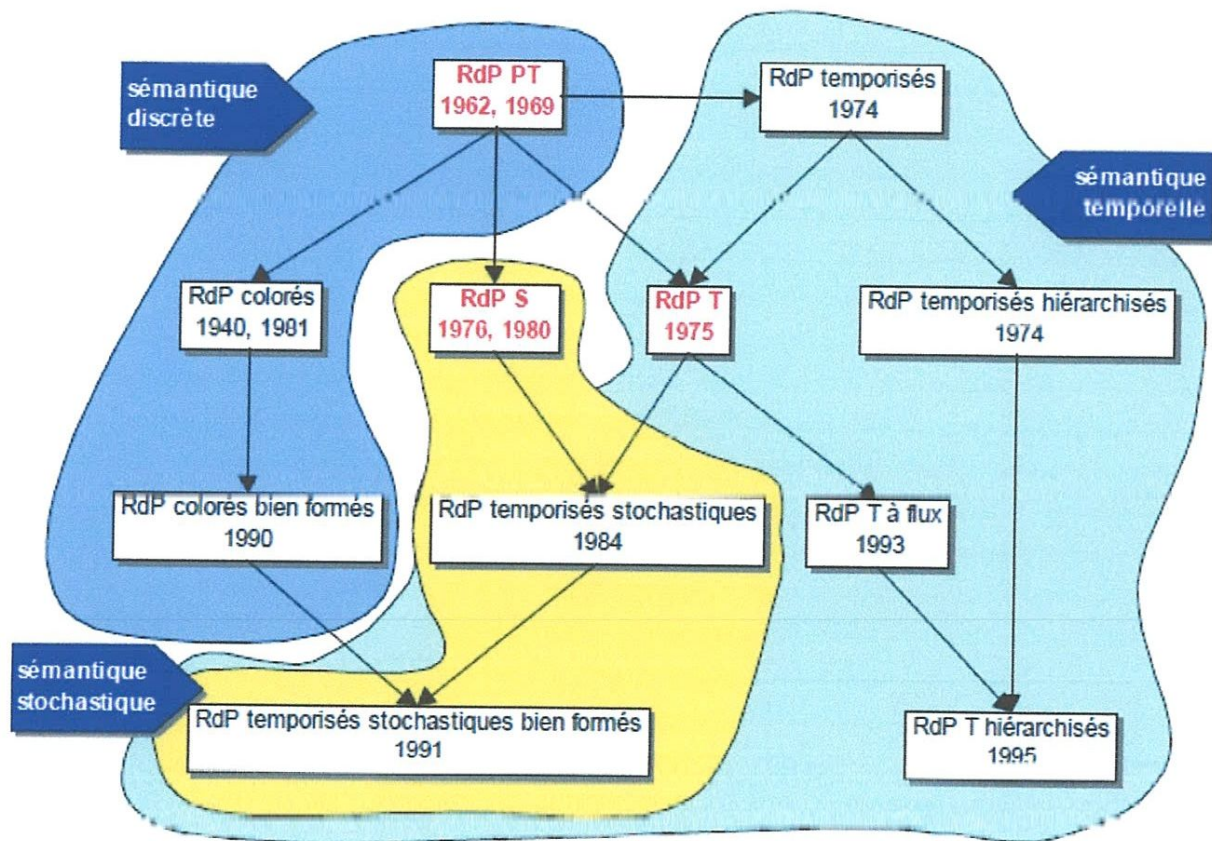


Figure 3.33 : Les familles de RdP.

II.5.1 - Réseaux de pétri autonomes :

Un RdP autonome décrit le fonctionnement d'un système de façon autonome, c'est-à-dire dont l'évolution est régi par ses propres lois. La succession des quarts saisons peut être ainsi modélisée par un RdP autonome.

❖ **Réseaux de pétri ordinaire :**

Tous les arcs ont des poids égaux à 1.

❖ **Réseaux de pétri à arcs étiquetés ou généralisé :**

Un exemple de RdP généralisé et de RdP ordinaire équivalent est présenté Figure 2.24. L'intérêt de l'utilisation de RdP généralisés est évident : il permet d'obtenir des modèles RdP plus concis. Les RdPs généralisés font partie des classes de modèles RdPs appelées abréviations. Ces classes de modèles RdPs permettent de représenter les mêmes systèmes que les RdPs mais avec une représentation graphique beaucoup plus concise. Une seconde classe d'abréviations est les RdPs colorés.

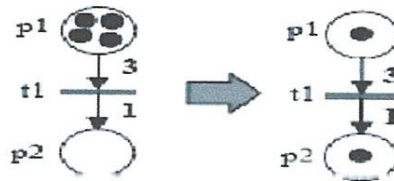


Figure 2.24 : RdP à arc étiquetés.

❖ **Réseau de pétri interprété :**

Tir de la transition si :

- 1) transition sensibilisée
- 2) prédicat vrai ($'ab' = 1$)

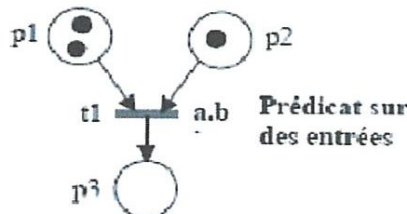


Figure 2.25 : RdP interprété.

❖ Les réseaux de pétri coloré :

Lorsque le nombre d'entités du système à modéliser est important, la taille du réseau de pétri devient rapidement énorme ; et si les entités présentent des comportements similaires, l'usage des réseaux colorés permet de condenser le modèle. Les réseaux de pétri colorés sont des réseaux de pétri dans lesquels les jetons portent des couleurs.

Une couleur est une information attachée à un jeton. Cette information permet de distinguer des jetons entre eux et peut être de type quelconque. [Kurt1 97]

Ainsi, les arcs ne sont pas seulement étiquetés par le nombre de jetons mais par leurs couleurs. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires, qui en plus satisfont les couleurs qui étiquettent les arcs. Après le franchissement d'une transition, les jetons qui étiquettent les arcs d'entrée sont retirés des places en entrée tandis que ceux qui étiquettent les arcs de sortie sont ajoutés aux places en sortie de cette transition.

Les réseaux colorés n'apportent pas de puissance de description supplémentaire par rapport aux réseaux de pétri, ils permettent juste une condensation de l'information. A tout réseau de pétri coloré marqué correspond un réseau de pétri qui lui est isomorphe. La relation entre le RdP coloré et le RdP ordinaire vu comme une relation entre le langage de programmation de haut niveau et le code assembleur.

Théoriquement les deux niveaux d'abstraction ont la même sémantique. De plus le langage de haut niveau offre une grande puissance de modélisation par rapport au langage assembleur, car il est bien structuré, bien typé et modulé. [Kurt1 97]

11.5.2 - Réseaux de pétri non autonomes :

L'évolution du RdP va maintenant dépendre de l'occurrence d'événements externes ou du temps. Par simplicité, on se concentrera sur l'extension des RdPs ordinaires, même si cela peut s'appliquer aux RdPs généralisés.

❖ Réseaux de pétri synchronisés :

Dans les modélisations RdPs que nous avons vues précédemment, le fait qu'une transition soit franchissable indique que toutes les conditions sont réunies pour qu'elle soit effectivement franchie. Le moment où se produira le franchissement n'est pas connu. Un RdP synchronisé est un RdP où à chaque transition est associée un événement. La transition sera alors franchie si elle est validée mais quand l'événement associé se produit.

Dans un RdP synchronisé, une transition validée n'est pas forcément franchissable. La transition est validée quand la condition sur les marquages est satisfaite. Elle deviendra franchissable quand l'événement externe associé à la transition se produit : elle est alors immédiatement franchie. Si en fonction du marquage de ses places d'entrée, plusieurs franchissements sont possibles, un seul se produira effectivement, celui dont l'événement associé se produit en premier.

Les notions de bornitude et de vivacité peuvent se généraliser aux RdPs synchronisés.

Un RdP synchronisé est borné si, pour tout marquage stable et instable, il est borné. Une transition T_j est vivante si pour tout marquage accessible, il existe une séquence d'événements externes telle que T_j soit franchie. Si on considère le RdP autonome associé au RdP synchronisé. [GG06]

❖ Réseaux de pétri temporels et temporisés :

Il existe deux principales familles d'extension temporelle des réseaux de pétri : réseaux de pétri temporisés introduits par Ramchandani [RAM 74] et les réseaux de pétri temporels introduits par Merlin [MER 74]. Pour les réseaux de pétri temporisés, les temporisations ont d'abord été associées aux transitions (t-temporisés), puis aux places (p-temporisés).

La temporisation représente alors la durée minimale de tir ou le temps de séjour minimum d'un jeton dans une place (ou durée exacte avec une règle de fonctionnement au plus tôt).

Les RdP t-temporisés et p-temporisés sont équivalents et sont une sous-classe des réseaux de pétri temporels.

Concernant les réseaux de pétri temporels, l'extension temporelle s'exprime sous la forme d'un intervalle associé principalement aux transitions (t-temporel) [MER 74], ou aux places (p-temporel).

En ce qui concerne l'expressivité des réseaux de pétri p-temporels et t-temporels, ces deux modèles sont incomparables. [Franck]

Enfin, les réseaux de pétri t-temporels forment une sous classe des Time Stream pétri nets qui ont été introduits pour modéliser des applications multimédia. [Franck]

Le problème de la bornitude est indécidable pour ce type de RdP et les travaux sur ce modèle reportent les résultats de décidabilité (comme l'accessibilité) sur l'hypothèse de bornitude du réseau. [Franck] La bornitude (et les autres problèmes) sont alors résolus par le calcul de l'espace d'états (quand celui-ci termine).

❖ Réseaux de pétri stochastiques :

Les réseaux de pétri stochastiques (RdPS) ont été développés. En effet, en introduisant une temporisation dans les RdP, le but est de concevoir un modèle unique permettant à la fois une validation qualitative et quantitative de réseaux.

Un RdPS est un RdP temporisé doté d'une mesure de probabilité sur l'espace des trajectoires, c'est à dire que les séquences de franchissement sont mesurables en considérant un espace aléatoire. De façon formelle :

Un RdPS est un couple $S = \langle R, \phi \rangle$, tel que :

- $R = \langle P, T, \text{Pré}, \text{Post}, M_0 \rangle$ est le réseau sous-jacent.
- $\phi : T \rightarrow \mathbb{R}^+$, la fonction qui associe à chaque transition un taux de franchissement fixe.
- M_0 : marquage initial du réseau.

III - La sémantique de maximalité :

III.1 - Principe de la sémantique de maximalité: [Sai96]

La sémantique d'un système concurrent peut être caractérisée par l'ensemble des états du système et des transitions par les quelles le système passe d'un état à un autre. Dans l'approche basée sur la maximalité, les transitions sont des événements qui ne représentent que le début de l'exécution des actions. En conséquence, l'exécution concurrente de plusieurs actions devient possible, c'est-à-dire que l'on peut distinguer exécutions séquentielles et exécutions parallèles d'actions.

Etant donné que plusieurs actions qui ont le même nom peuvent s'exécuter en parallèle (auto-concurrence), nous associons, pour distinguer les exécutions de chacune des actions, un identificateur à chaque début d'exécution d'action, c'est-à-dire à la transition ou à l'événement associé. Dans un état, un événement est dit maximal s'il correspond au début de l'exécution d'une action qui peut éventuellement être toujours en train de s'exécuter dans cet état là. Associer des noms d'événements maximaux aux états nous conduit à la notion de configuration.

- **Définition : [BOU11]**

"Système de Transitions Etiquetées Maximales "

M étant un ensemble dénombrable des noms des évènements, un Système de Transitions Etiquetées Maximales (STEM) de support M est un quintuplet $(\Omega, A, \mu, \xi, \psi)$ avec :

- $\Omega: \langle S, T, \alpha, \beta \rangle$ est un système de transitions tels que :
 - S : est un ensemble d'états qui peut être fini ou infini.
 - T : est un ensemble de transitions qui peut aussi être fini ou infini.
 - α et β sont deux applications de T dans S tels que pour toute transition t de T

On a : $\alpha(t)$ l'origine de la transition et $\beta(t)$ son but.

- $\langle \Omega, A \rangle$ est un système de transitions étiquetées par un alphabet A.

III.2 - Sémantiques autour de la maximalité [BEL10] :

L'étude d'un système de manière formelle nécessite généralement la donnée d'une sémantique précise aux langages et modèles de spécification utilisés. Dans le besoin de Préserver la nature des systèmes à spécifier, nous nous intéressons aux sémantiques dites de vrai parallélisme. Ces sémantiques nous offrent la possibilité de spécifier, dans un contexte de haut niveau, le comportement des systèmes à étudier tout en s'intéressant à la concurrence.

Le choix de cette sémantique est poussé par le fait qu'elle est prouvée nécessaire et suffisante pour le raffinement d'actions et les durées d'actions.

C'est un modèle de vrai parallélisme, il nous permet de déterminer pour chacun de ses états les actions qui sont en train de s'exécuter en parallèle; cela est assuré en associant les informations sur le parallélisme aux états et aux transitions, dont les états contiennent les événements des actions qui sont potentiellement en exécution, et les transitions sont des événements qui ne représentent que le début de l'exécution des actions.

IV - Réseau de pétri via la sémantique de maximalité : [BEL10]

Nous essayons maintenant de donner une sémantique de maximalité en terme de systèmes de transitions étiquetées maximales au modèle des réseaux de Pétri place/transition.

Il est à noter qu'une sémantique de maximalité et une relation de bisimulation de maximalité ont été définies par R. Devillers pour les réseaux de Pétri place/transition en termes de réseaux d'occurrences. Ces derniers sont des structures infinies si le comportement du système correspondant est infini. Par conséquent, cette approche est intéressante du point de vue théorique. L'avantage de l'utilisation de modèle des systèmes de transitions étiquetées maximales est que ce dernier est un modèle implémentable.

Nous commençons par introduire la notion de la maximalité avec des exemples des réseaux de pétri, ce qui permettra de générer, pour tout système étiqueté, un système de transitions étiquetées maximales.

Soit le réseau de pétri marqué de la Figure 2.26. (a). Après le tir de la transition t_1 , il est évident que les tirs des transitions t_2 et t_3 sont conditionnés par la fin de l'action liée à la transition t_1 . Pour capturer cette dépendance causale entre les tirs des transitions, nous considérons que les jetons produits par le tir de la transition t_1 sont liés à cette transition, entre autres le jeton de la place s_2 et le jeton de la place s_3 . On peut remarquer que, dans l'état initial, le jeton dans s_1 n'est lié à aucune transition, ce jeton est dit libre dans cet état là. Dans le cas où la transition t_2 serait tirée, on pourrait déduire que l'action associée au tir de t_1 s'est terminée. De ce fait, le jeton dans s_3 deviendra libre. Le marquage résultant du tir de la transition t_2 est donné par la Figure 2.26. (c).

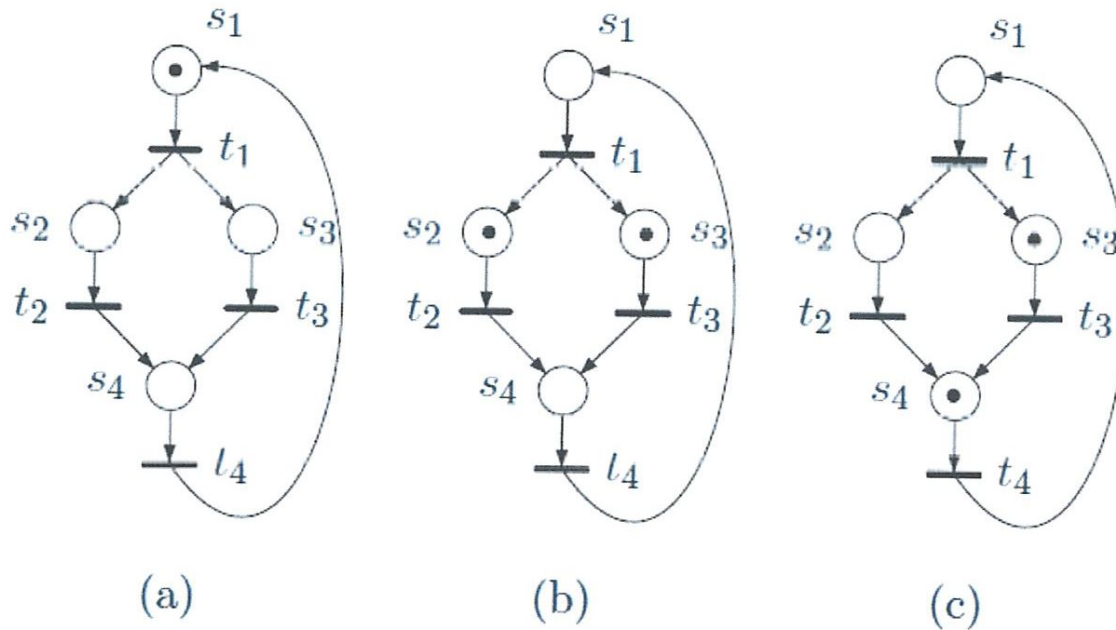


Figure 2.26 : Exemple d'un réseau de pétri marqué.

Pour distinguer entre jetons libres et jetons liés dans une place, on peut imaginer qu'une place est composée de deux parties disjointes. La partie de gauche contiendra les jetons libres tandis que celle de droite contiendra les jetons liés.

Dans une place, le nombre des jetons libres sera désigné par FT (pour free tokens) alors que l'ensemble des jetons liés sera désigné par BT (pour bound tokens). On obtiendra ainsi la succession des marquages de la Figure 2.27.

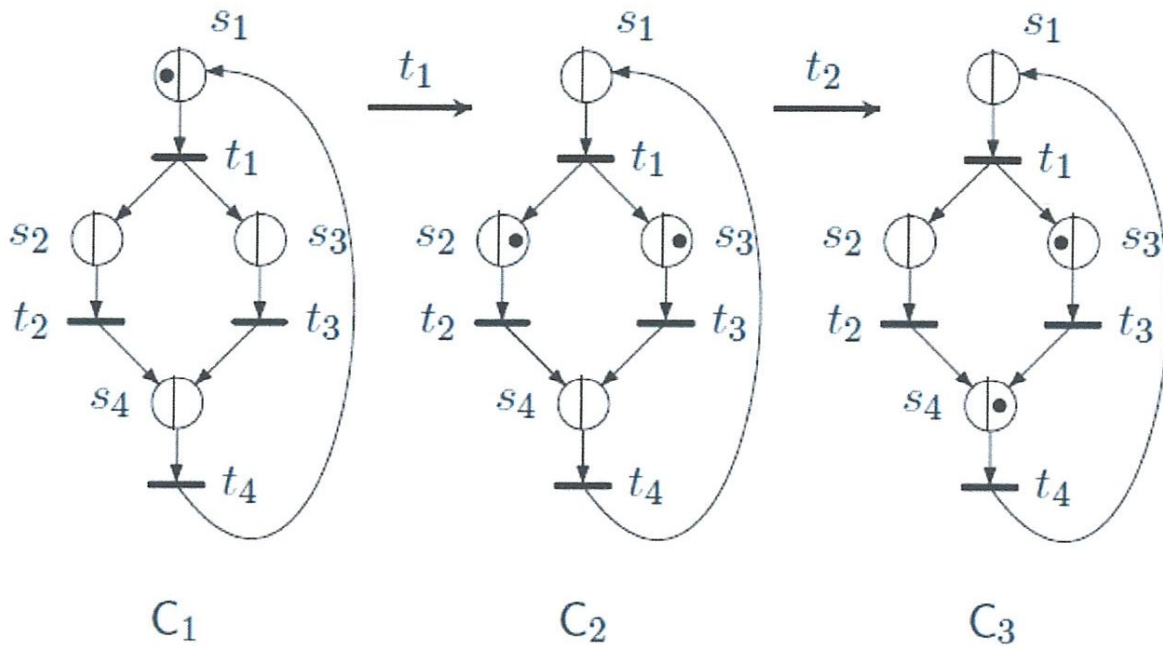


Figure 2.27 : Jetons libres et jetons liés dans un marquage.

A partir de la configuration C_2 on aurait pu tirer la transition t_3 . La configuration résultante serait C_4 . (Voir Figure 2.28). Le jeton dans s_4 est maintenant lié à la transition t_3 et le jeton dans s_2 est désormais libre puisque la fin de l'action associée à la transition t_1 implique la libération du jeton de s_2 .

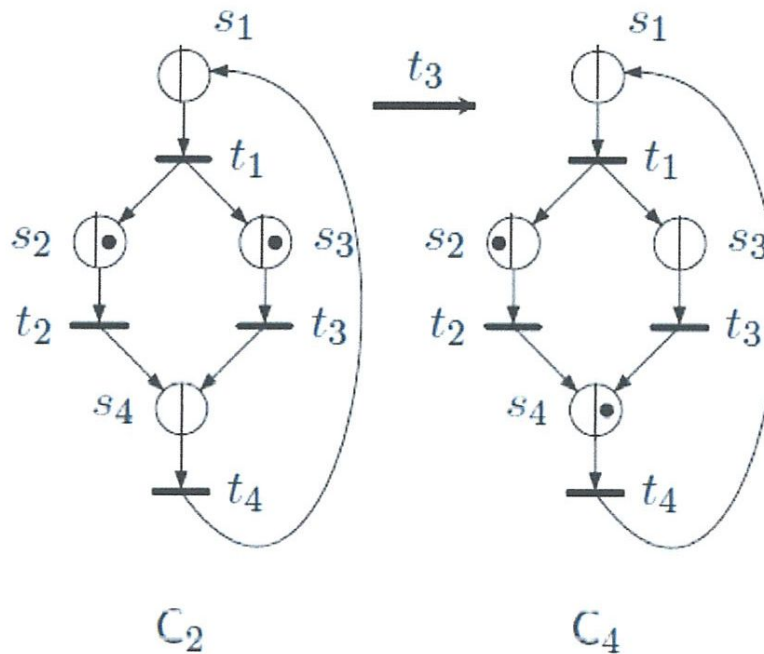


Figure 2.28 : Franchissement de la transition t_3 .

IV.1 - Liaison des jetons :

Par un tir de la transition t_1 , on obtiendra le réseau marqué de la Figure 2.29. (b). A partir de ce marquage, on remarque que la transition t_1 est franchissable. Le franchissement de cette transition conduira à la configuration de la Figure 2.29. (c).

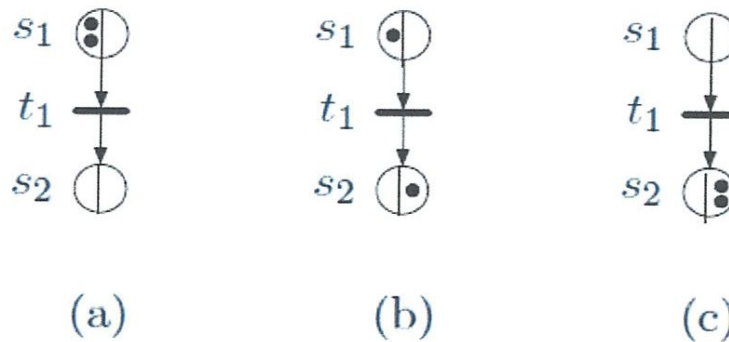


Figure 2.29 : Liaison des jetons.

Les deux jetons de la place s_2 sont liés. En effet, l'un est lié au premier tir de la transition t_1 alors que l'autre est lié au second tir de la même transition (deux actions associées à t_1 pouvant être en exécution parallèle). Pour lever cette ambiguïté, chaque tir de transition sera identifié par un nom d'événement. De ce fait, le lien d'un jeton sera caractérisé aussi bien par la transition qui la produite que par le nom d'événement identifiant le franchissement de cette transition.

IV.2 - Successions de franchissement :

Les successions des franchissements de l'exemple précédent sont illustrées par la Figure 2.30.

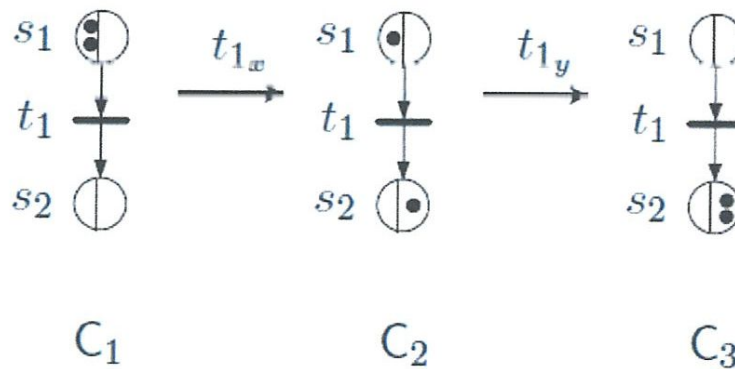


Figure 2.30 : Successions de franchissements de t_1 .

Dans la configuration C2, l'ensemble des jetons liés dans s2 est $BT = \{t1x\}$ alors que l'ensemble des jetons liés dans s2 de la configuration C3 est $BT = \{t1x, t1y\}$. Le nom de l'événement x désigne le premier franchissement de la transition t1 alors qu'y désigne le deuxième franchissement de cette même transition.

Un autre problème se pose quant aux jetons liés à un même tir de transition. Pour le voir, considérons le réseau de pétri de la Figure 2.31. (a).

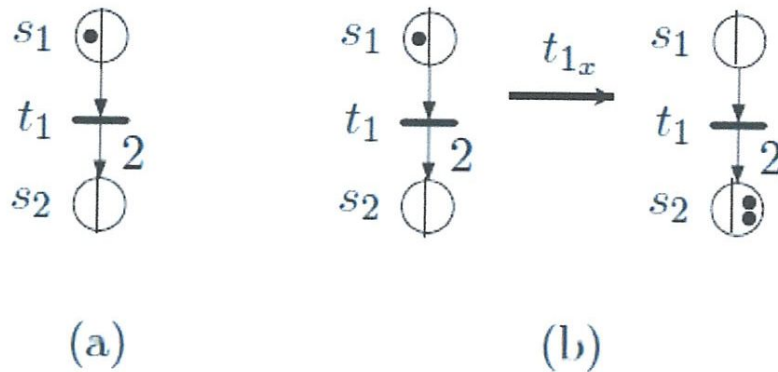


Figure 2.31 : Réseau de pétri avec arc sortant de poids supérieur a un.

Par le franchissement de la transition t1 on obtient la dérivation de la Figure 2.31. (b). La partie droite BT de la place s2 contient deux jetons liés au tir t1x, c'est-à-dire $BT = \{t1x, t1x\}$. Etant donné que BT est un ensemble, nous considérons qu'un jeton lié est un triplet (n, t, x) de $N \times T \times M$, (noté aussi ntx), où n est le nombre d'instances, t est la transition productrice de ce jeton et x est le nom d'événement associé au franchissement de la transition t. On note $BT = \{n1t1x1, n2t2x2, \dots\}$ l'ensemble (éventuellement vide) des jetons liés. Dans l'exemple précédent $BT = \{2t1x\}$. Par conséquent, le marquage d'une place s est un couple (FT, BT) où FT est le nombre de jetons libres dans s.

IV.3 - Identification des jetons consommés :

Nous admettons qu'un jeton de s_3 est lié au franchissement t_1 (t_1x) et que l'autre jeton est lié au franchissement de t_2 (t_2y).

Parmi les jetons liés dans s_3 , on veut connaître le jeton consommé dans le premier franchissement de t_3 et celui consommé dans le second franchissement de cette même transition. Cette information est indispensable pour connaître, dans chaque configuration, les actions (associées aux transitions) qui ont terminé leur exécution. Pour ce faire, nous associons au niveau d'un franchissement les noms d'événements identifiant les jetons liés consommés par ce franchissement. Ceci nous donne la succession des franchissements de la Figure 2.32.

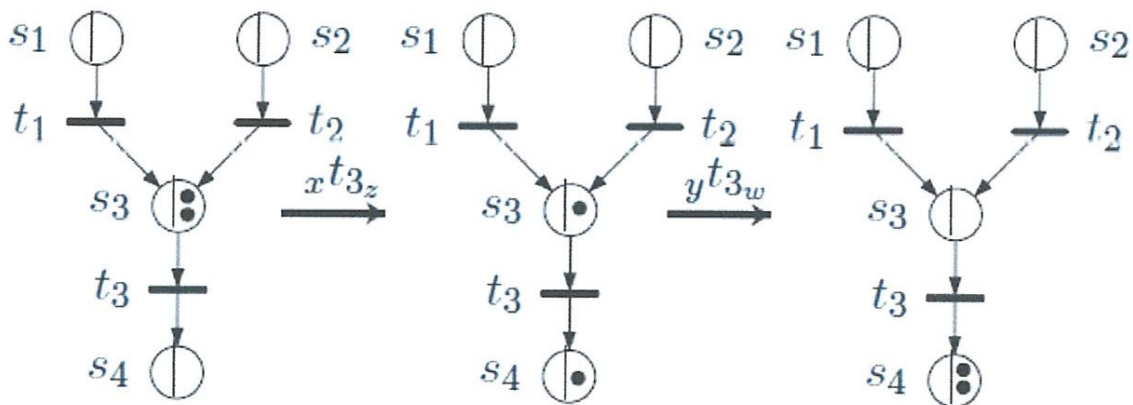


Figure 2.32 : Noms d'événements identifiant les jetons consommés.

Dans l'exemple de la Figure 2.27, nous avons noté qu'après le franchissement de la transition t_2 , le jeton qui était lié à la transition t_1 dans la place s_3 devient libre dans le marquage résultant. Ceci est dû au fait que le franchissement de la transition t_3 est conditionné par la fin de l'action relative à la transition t_1 .

V - Conclusion :

Dans ce chapitre, on a parlé des réseaux de pétri ou on a montré les différentes propriétés et les concepts de bases et aussi nous avons présenté l'intuition derrière la sémantique de maximalité cette sémantique qui nous conduit de parler des systèmes de transitions étiquetées maximales qui sont des modèles de vrai parallélisme.

En échappant à l'hypothèse d'atomicité spatiale et temporelle des actions, les transitions de ce modèle sont considérées comme des événements représentant que le début de l'exécution des actions, et l'ensemble d'événements maximaux associé à chaque état correspond aux actions qui sont potentiellement en cours d'exécution. Cela rend possible d'une part, l'introduction de la sémantique de maximalité aux méthodologies de conception basées sur le raffinement d'actions, et dans une autre part, la prise en compte du temps.



Chapitre 03

Conception

I - Introduction :

Parmi les langages de description formelle du système les réseaux de pétri, qui est un formalisme du comportement de systèmes de haut niveau, il possède à la fois une syntaxe et une sémantique bien définie (dit formel). Dans ce projet on va baser sur la sémantique des modèles de vrai parallélisme.

II - Conception :

Notre application qui nous avons développé, est un simulateur de RDP-maximalité et elle se compose on deux grandes parties, se traduisant en deux fenêtres :

- ✚ la première concerne la représentation graphique (création des RDPs).
- ✚ la deuxième concernant la simulation des RDPs (Franchissement).

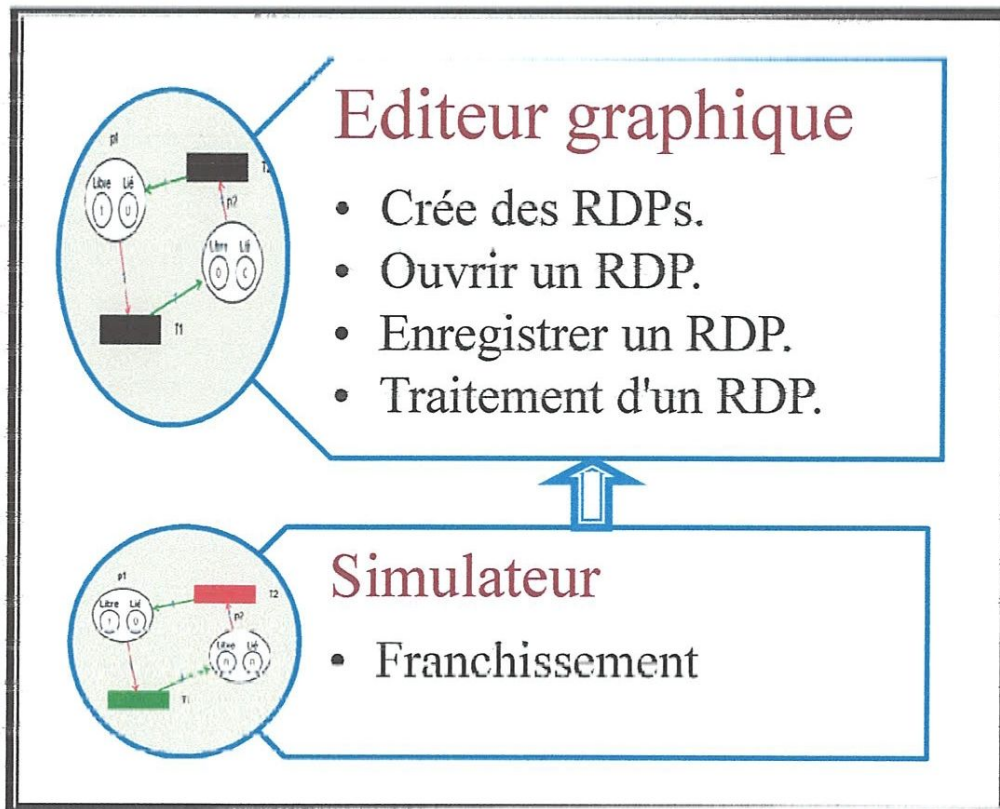


Figure 3.01 : Conception de notre système.

II.1 - Conception de l'éditeur graphique :

cette partie presenter notre editeur graphique donc on va vois voir les different fonction de cette editeur graphiques.

II.1.1 - Creation de RDP :

Elle se traduit on quatre éléments : les places, les transitions, les flèches entrantes et les flèches sortantes.

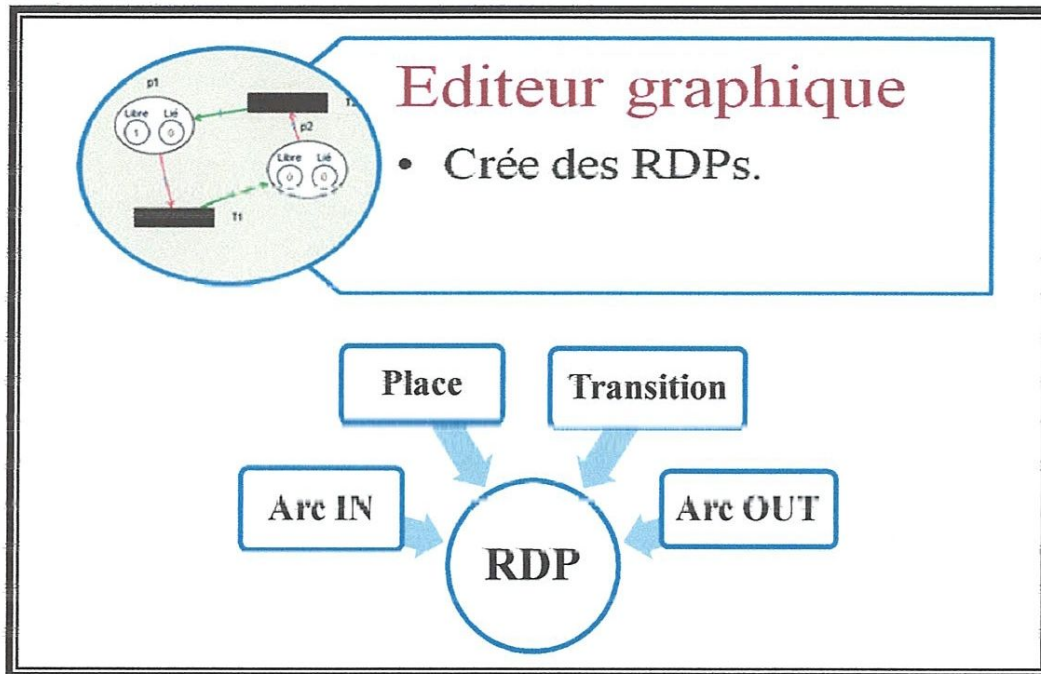


Figure 3.02 : Création de RDP.

Les place et les transitions se sont des vertex crée a l'aide de l'outil JgraphX.

Et voici comment implanter un Vertex «Transition» :

```
Object Place1 = Graphe1.insertVertex (Parent1, null, "", X, Y, 80, 20,
"strokeColor=black;fillColor=black");
```

Et les Arc IN et les Arc OUT se sont des Edge crée à l'aide de l'outil JgraphX.

Et voici comment implanter un Edge : «Arc IN».

```
Object Fleche1 = Graphe1.insertEdge (Parent1, null, Poids, Objet1, Objet2,
"startArrow=none;endArrow=open;strokeColor=red;");
```

II.1.2 - L'ouverture et l'enregistrement d'un RDP :

On a utilisé l'outil JDOM pour faire cette opération.

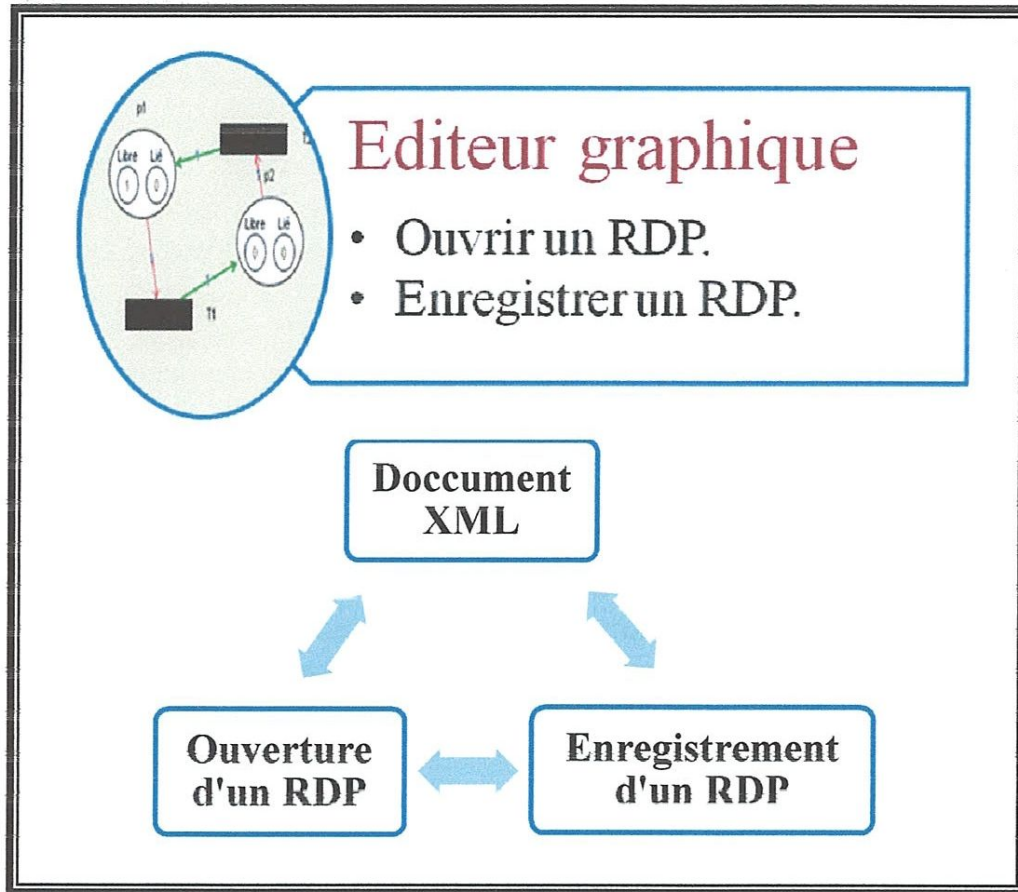


Figure 3.03 : L'ouverture et l'enregistrement de RDP.

Pour ouvrir un RDP :

```
Documents = Instance1.build (new File ("Reseaux/" + Chaine1));
```

Pour enregistrer un RDP :

```
new XMLOutputter (Format.getPrettyFormat ()).output (Documents, new  
FileOutputStream ("Reseaux/" + Chaine1 + ".xml"));
```

II.1.3 - Traitement d'un RDP :

Dans cette partie on va présenter les différentes opérations sur les éléments graphiques de RDP.

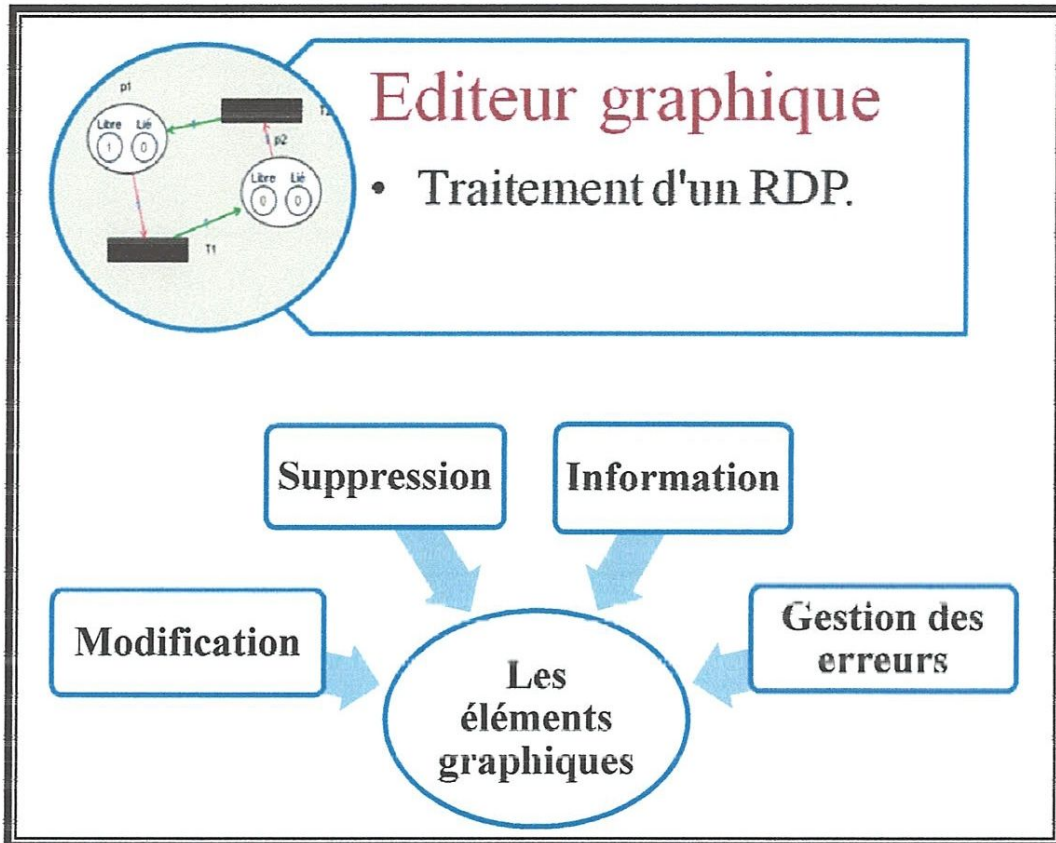


Figure 3.04 : Traitement de RDP.

II.2 - Conception du simulateur :

La simulation s'affiche dans une autre fenêtre mais elle est en relation avec l'éditeur par ce que elle utilise les graphes qui ont été créés ou enregistré dans un répertoire.

La simulation sera passée avec plusieurs étapes :

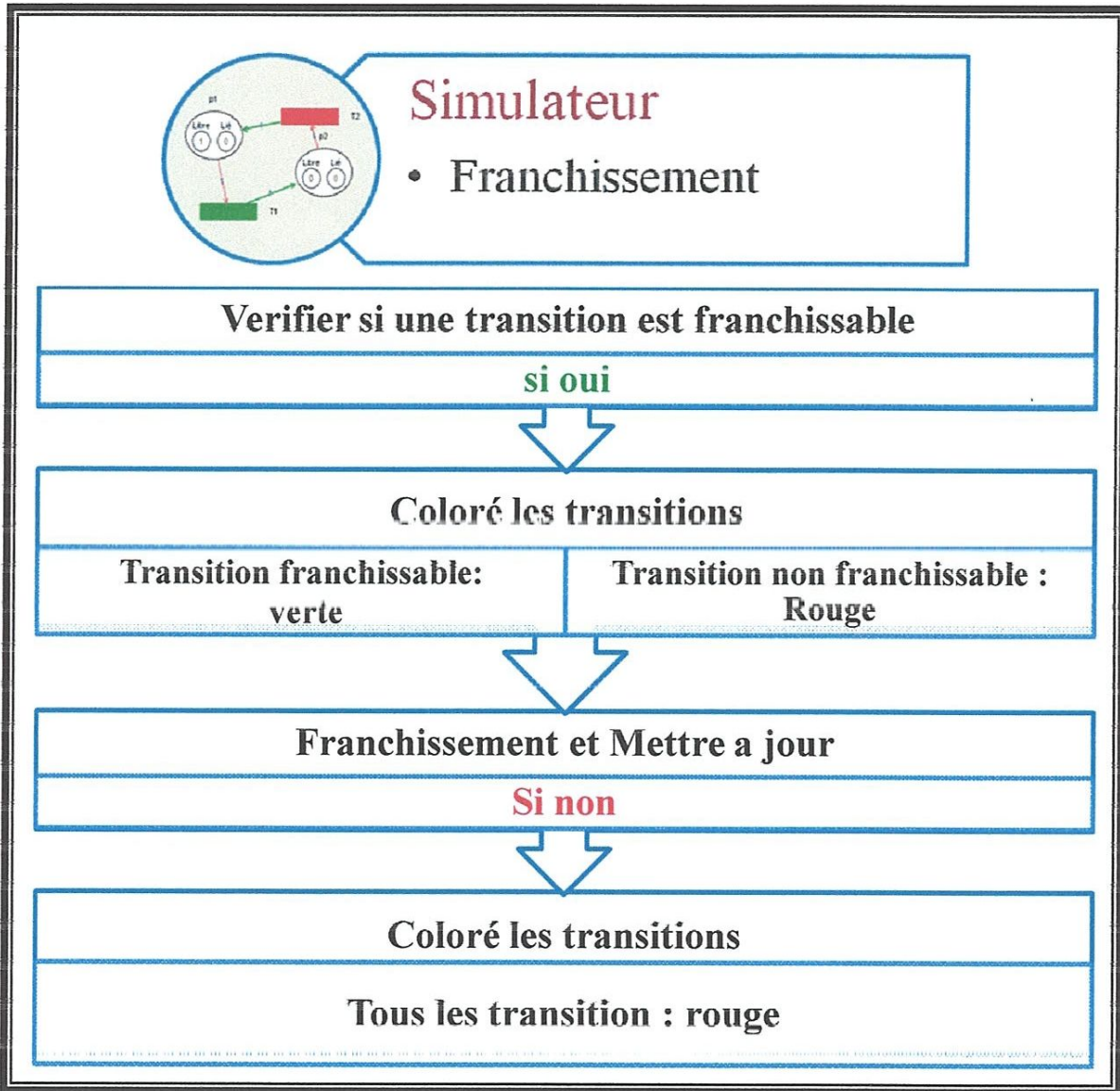


Figure 3.05 : La Simulation.

II.2 .1 - Vérification pour franchissement :

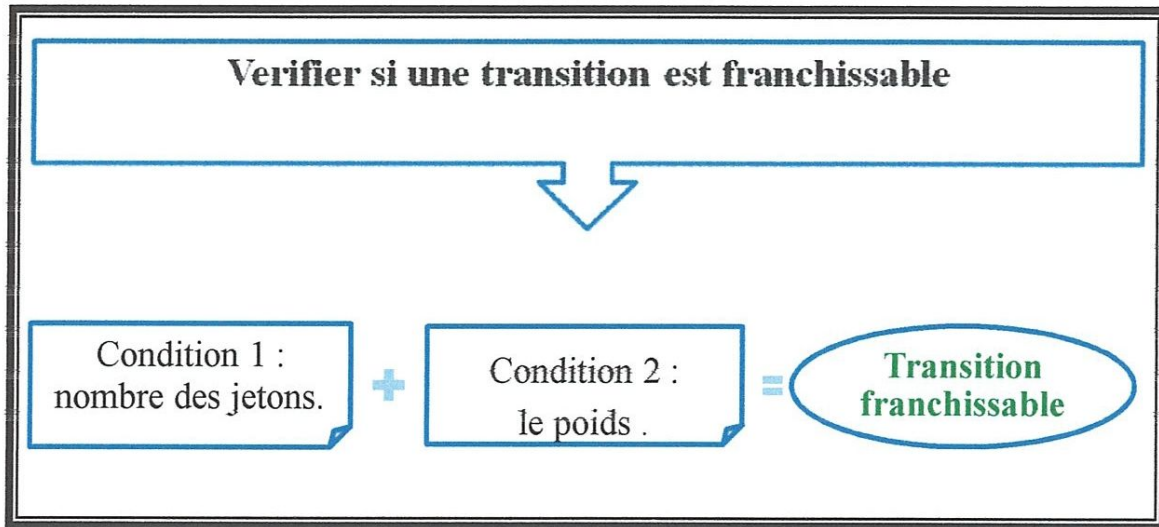


Figure 3.06 : La vérification.

Après l'étape de vérification on va :

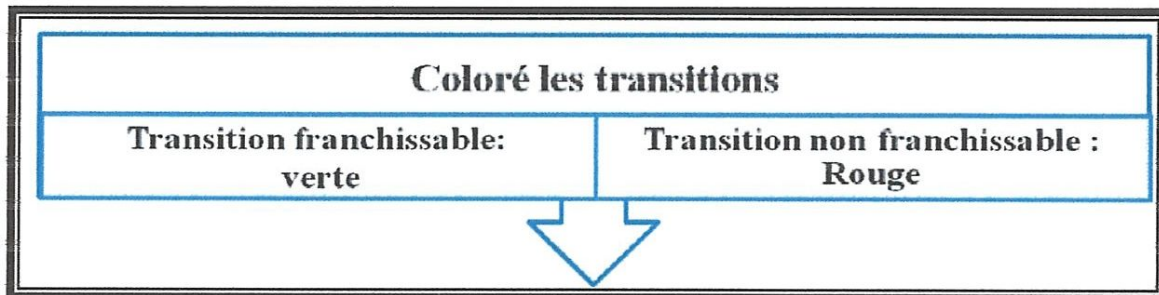


Figure 3.07 : Colorisation des transitions.

Insérer un Vertex verte et un Vertex rouge.

```
Object Place1 = Graphe1.insertVertex (Parent1, null, "", X, Y, 80, 20,
"strokeColor=green;fillColor=green");
```

```
Object Place1 = Graphe1.insertVertex (Parent1, null, "", X, Y, 80, 20,
"strokeColor=red;fillColor=red");
```

II.2.2 – Franchissement et mettre a jour :

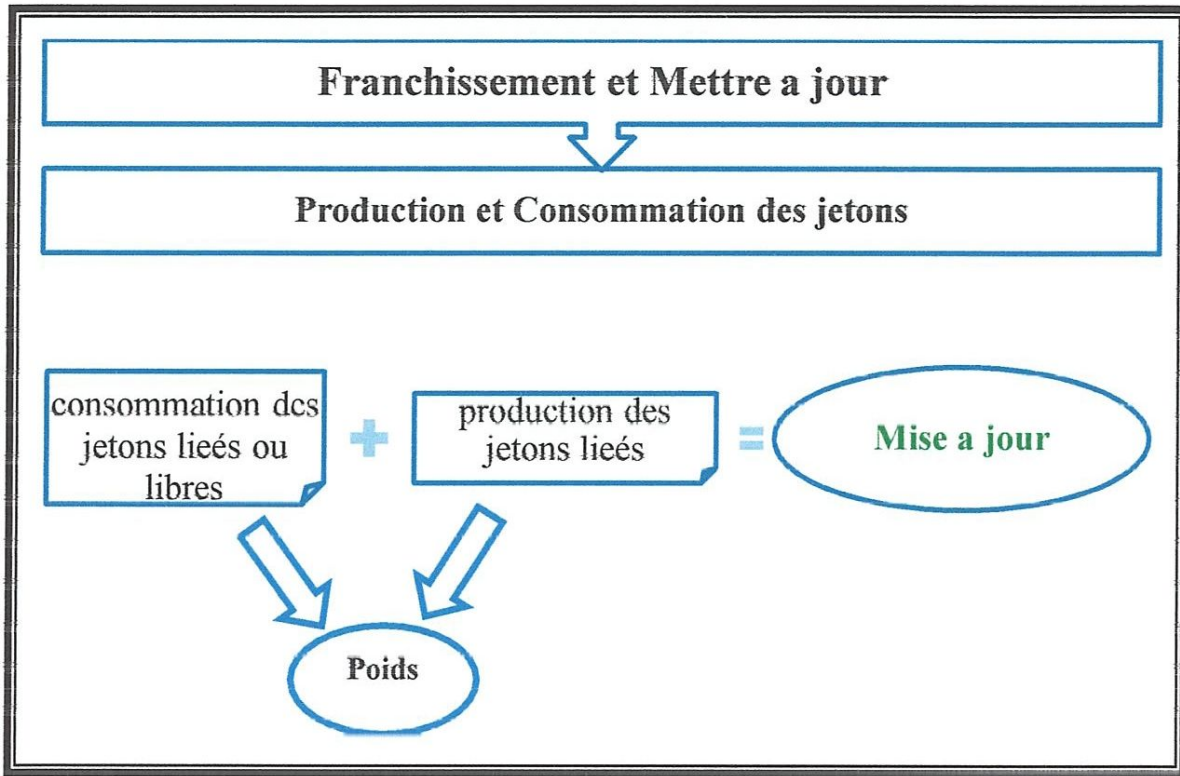


Figure 3.08 : Franchissement des transitions.

III - Conclusion :

Dans ce chapitre on parlé de la phase de la conception de notre système ou on a présentés les différentes fonctionnalités sous forme boites noires et développer jusqu'à on arrive a un état détaillé dans le chapitre suivant on va voir les différentes utilisations de notre application plus une partie des études des cas plus précis et plus détaillé.



Chapitre 04

Etude des cas

I - Introduction :

Dans ce chapitre on va présenter l'application, qui est un simulateur de RDP-maximalité elle est composée de deux grandes parties, se traduisant en deux fenêtres qui sont la fenêtre qui concerne la représentation graphique (création des RDPs) et la deuxième qui concerne la simulation des RDPs ou on va présenter les détails de chaque fenêtre et leurs fonctionnalités de ce chapitre et divise en deux : la partie qui concerne l'application (affichage, utilisation,...) et la deuxième partie est une étude de cas.

II - Réalisation :

II.1 - L'interface de l'éditeur :

Cette partie est peut-être la plus sensible du projet, étant donné que c'est elle qui centralise toutes les autres et les coordonne afin d'avoir un fonctionnement correct et respectant les attentes de l'utilisateur. Heureusement, Java fournit un grand nombre de classes qui permettent de faire ce type d'interface relativement facilement.

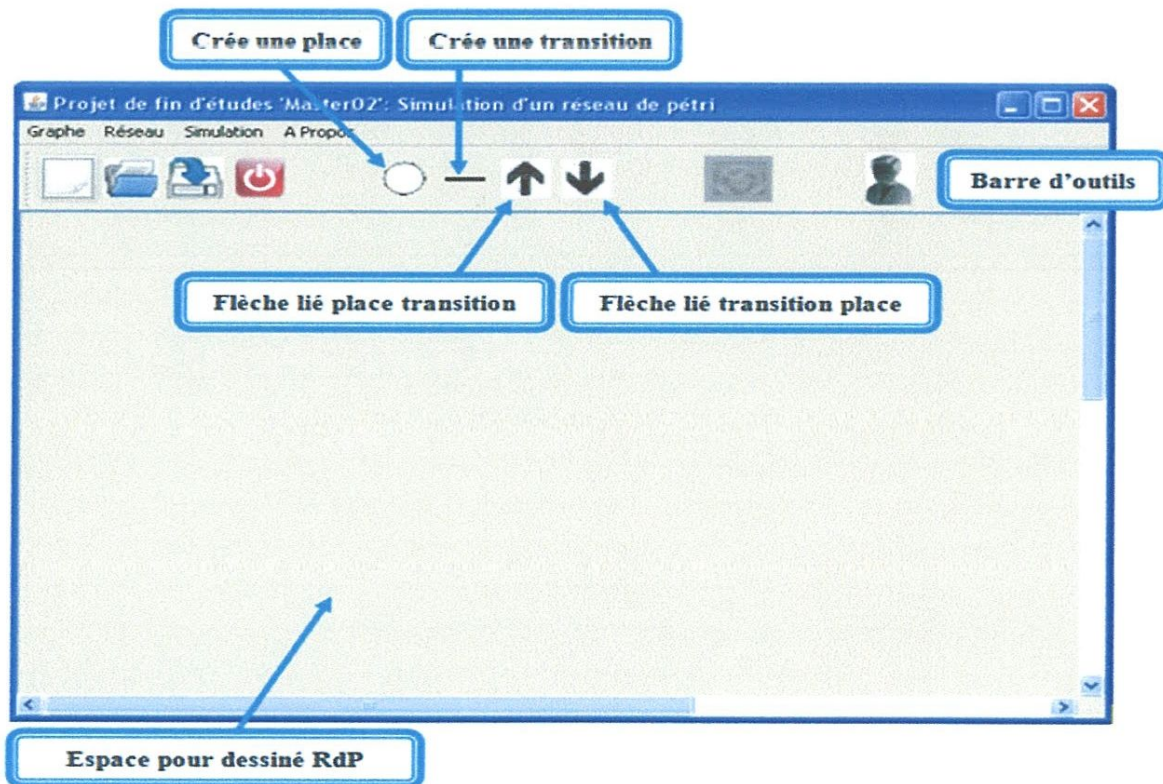


Figure 4.01 : L'interface de l'éditeur.

II.2 - Editeur:

Cette fenêtre qui est une spécialisation de JFrame gère aussi un grand nombre d'évènements, comme ceux relatifs au clavier (raccourcis claviers), aux actions (les boutons), aux fenêtres ou aux menus. Il gère aussi la liste des Afficheurs, l'ouverture et l'enregistrement d'un réseau, etc. Les menus (JMenu et JMenuItem) se trouvant sur la barre de menus (JMenuBar) permettent à l'utilisateur d'utiliser facilement ces fonctionnalités. La barre d'outils (JToolBar) affiche des boutons (JButton) sur lesquels se trouvent des icônes (ImageIcon) qui correspondent aux fonctionnalités les plus utilisées afin de faciliter leur identification.

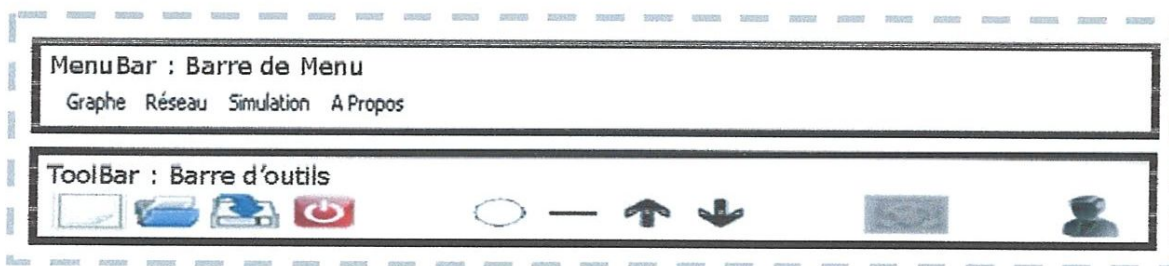


Figure 4.02 : La barre d'outils.

A - MenBar :

Cette barre de menu comporte des menus comme le menu Graphe, Réseau, Simulation, A propos, en voici une description :

Graphe : Ce menu possède les actions habituelles comme Nouveau, Ouvrir, Enregistrer, Quitter.

Réseau : Ce menu possède des éléments de création graphiques du réseau

Simulation : Ce menu donne la possibilité de lancer la simulation.

B - ToolBar :

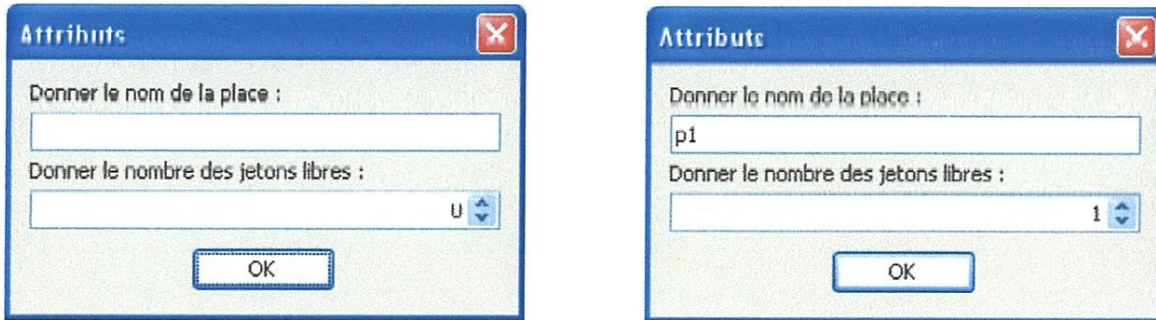
✚ les outils classiques :

- **Nouveau :** ce bouton nous permet de créer (dessiner) un nouveau graphe.
- **Ouvrir :** ce bouton nous permet de ouvrir un graphe qui été dessiner et enregistrer d'avance.
- **Enregistrer :** ce bouton nous permet d'enregistrer le graphe.
- **Quitter :** ce bouton nous permet de quitter l'application.

Les outils graphiques :

Ce sont les plus utiles pour utilisateur. En effet elle lui permet de créer les éléments graphiques du réseau. On peut donc créer ainsi une Place, une Transition, un ArcIN, un ArcOUT.

- **Création des places :** le bouton place nous permet de dessiner une place avec ces propriétés le nom et le nombre des jetons libres et choisi p1 pour le nom et 1 pour nombre des jetons libres.



Et voici le résultat :

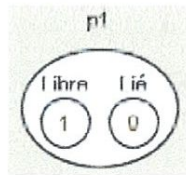
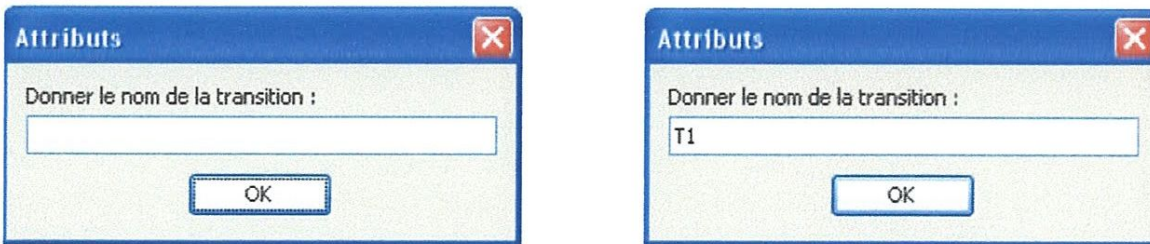


Figure 4.03 : Création des places.

- **Création des transitions :**

Le bouton transition nous permet de dessiner une transition avec cette propriété le nom nous avons choisi T1 comme nom.



Et voici le résultat :



Figure 4.04 : Création des transitions.

- **Création des arcs :** on a deux types de flèche :
 - **Arc IN :**

Le bouton ArcIN nous permet de dessiner un flèche entrent rouge qui relié une place avec une transition et ca propriété est le poids et nous avons choisi 1 comme poids par défaut.



Et voici le résultat :

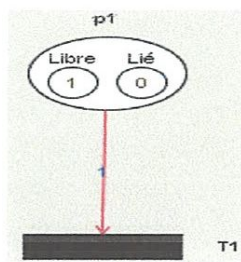


Figure 4.05 : Création des arcIN.

- **Arc OUT :**

Le bouton ArcOUT nous permet de dessiner un flèche sortent verte qui relié une transition avec une place et ca propriété est le poids et nous avons choisi 1 comme poids par défaut.



Et voici le résultat :

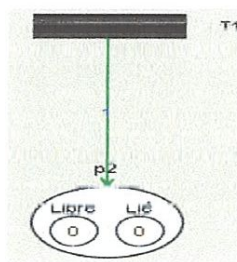


Figure 4.06 : Création des arcOUT.

Les outils de simulation:

- **Lancer la simulation :**

Ce bouton nous permet de lancer la simulation du graphe soit un nouveau ou bien un graphe déjà enregistré dans une autre fenêtre.

II.3 - Quelques opérations sur les éléments graphiques :

- **La suppression :**

Cette opération nous permet de supprimer soit une place ou bien une transition ou des arcs.

Supprimer une place:

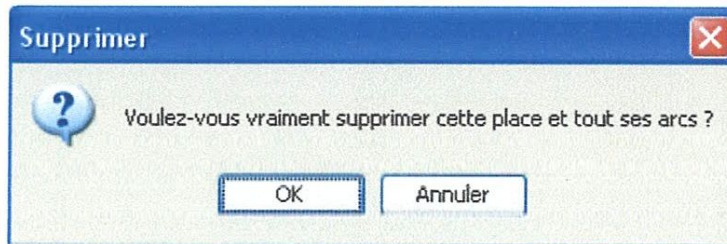


Figure 4.07 : Suppression d'une place.

Supprimer une transition:

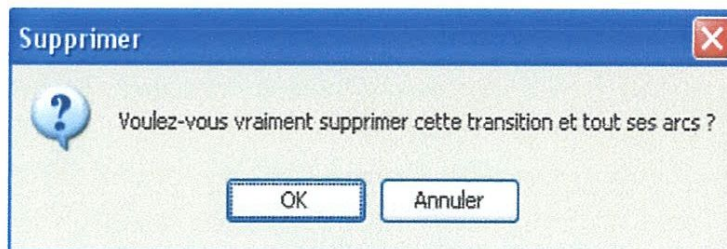


Figure 4.08 : Suppression d'une transition.

Supprimer un arc :

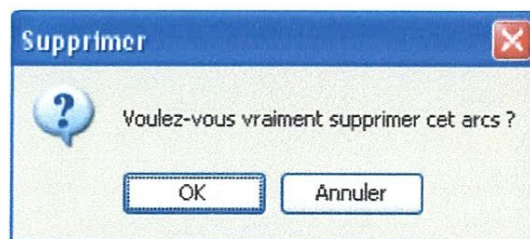


Figure 4.09 : Suppression d'un arc.

- **La modification :**

Cette opération nous permet de modifier une place c'est-à-dire ces propriétés nom et nombre des jetons libres ou une transition ca propriété nom ou bien les arcs ces propriétés le poids.

Modifier une place :

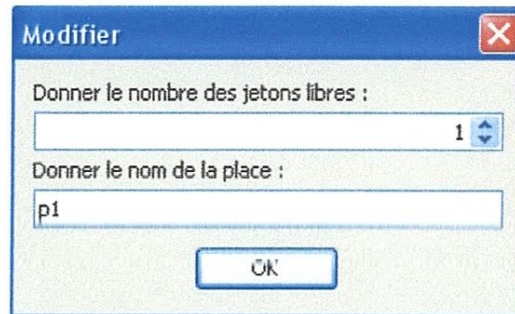


Figure 4.10 : Modification d'une place.

Modifier une transition :

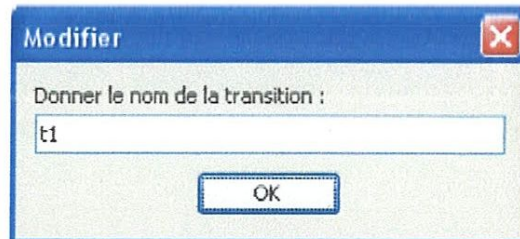


Figure 4.11 : Modification d'une transition.

Modifier un arc :

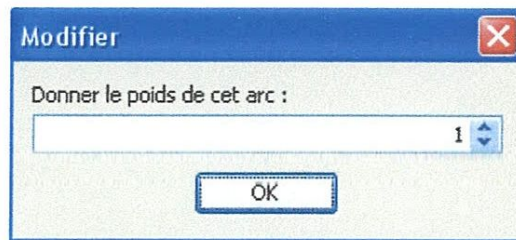


Figure 4.12 : Modification d'un arc.

- **Information :**

Cette opération nous permet d'afficher l'information de chaque élément de graph.

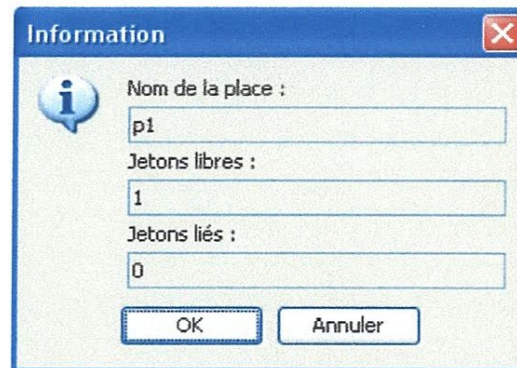


Figure 4.13 : Information d'une place.

- **Gestion des erreurs :**

On a deux messages d'erreur qui peuvent s'afficher :

Le premier indique qu'on ne peut pas donner un même nom pour deux places au même temps.

Le deuxième indique qu'on ne peut pas donner un même nom pour deux transitions au même temps.

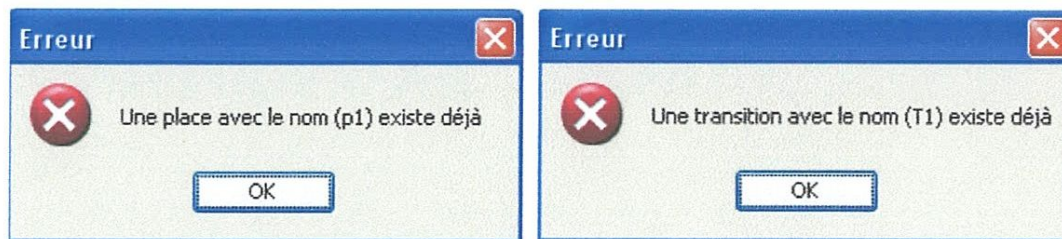


Figure 4.14 : Validation du nom de place et transition.

III - Etude des cas :

Dans cette partie on va traiter quelques exemples avec notre projet pour montrer le côté application des notions qu'on a vu parler de notre mémoire on va prendre quatre exemples.

Le premier est l'exemple de quatre saisons c'est un exemple simple pour comprendre les principes de l'application et le deuxième exemple c'est un exemple un peu compliqué pour montrer la sémantique de maximalité et le troisième c'est l'exemple de la machine à remplir et à boucler et on termine par l'exemple ou bien le problème des philosophes.

III.1 - Le modèle quatre saisons :

On a choisi l'exemple de quatre saisons qui contient quatre places (le nombre des saisons) et quatre transitions :

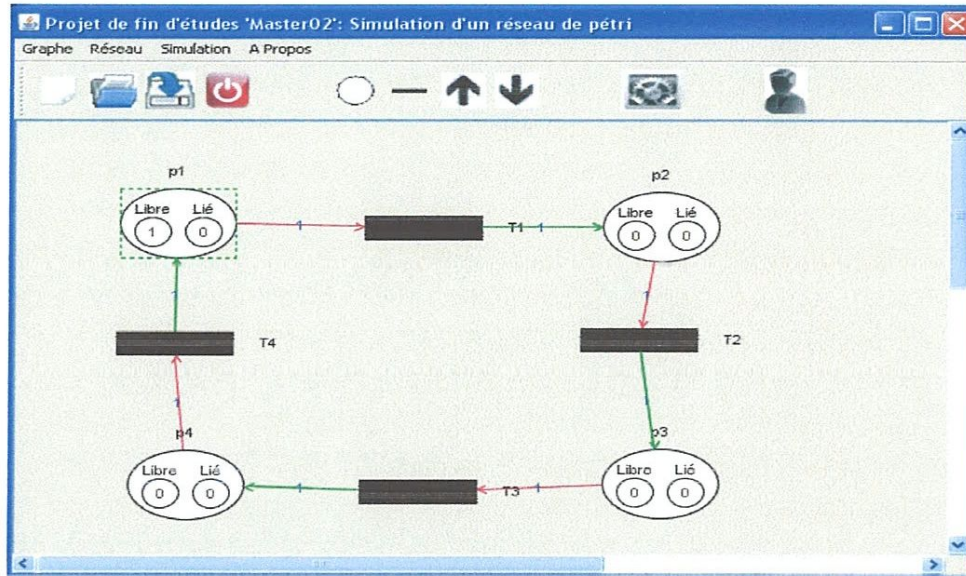


Figure 4.15 : Exemple d'un RDP de quatre saisons.

Et quant on lance la simulation on remarque que la transition T1 est franchissable car elle est coloré avec le vert. Et lorsque on clic sur la transition T1 on remarque le déplacement d'un seul (un seul jeton car le poids est égal a 1 pour l'arc) le jeton se déplace de la place P1 vers la place P2 et devient lié et la transition T2 devient franchissable.

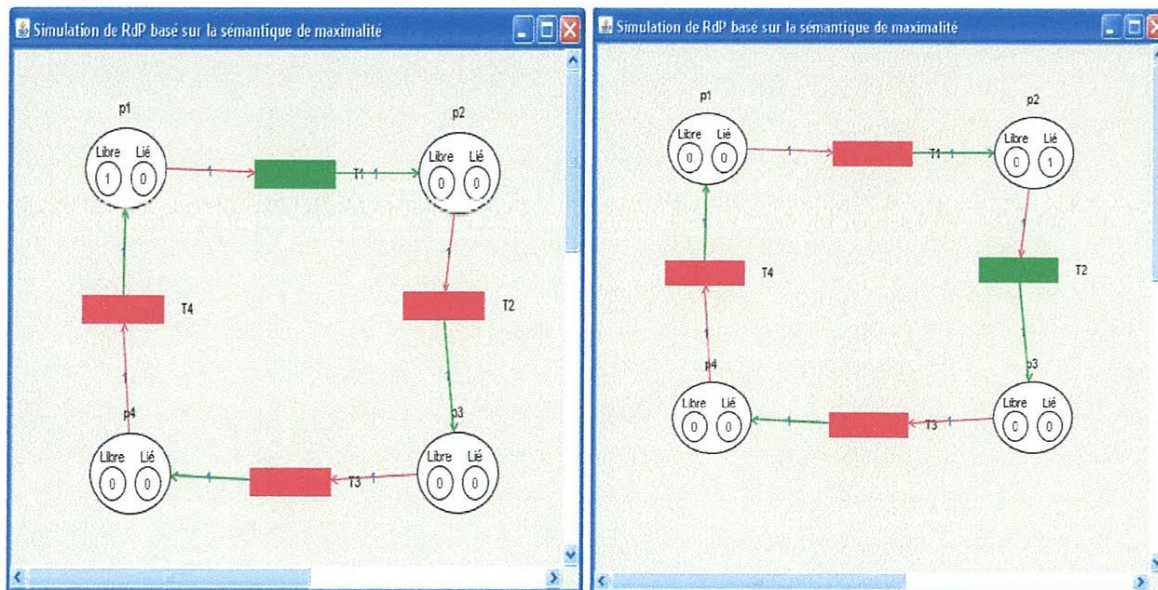


Figure 4.16 : Simulation de l'exemple de quatre saisons.

Et le même scénario pour les autres transitions :

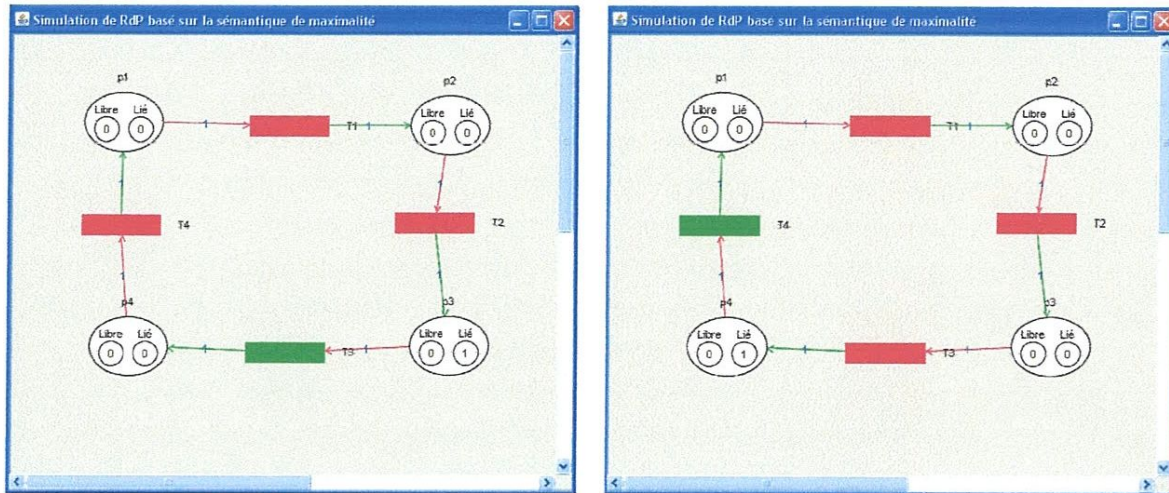


Figure 4.17 : Développement de la simulation de l'exemple de quatre saisons.

III.2 - Le modèle de la machine à remplir et à boucher :

III.2.1 - Description :

La machine à remplir et à boucher des bouteilles est composée de trois postes travaillant en parallèle.

- Le poste 1 :

Sert au transfert et au chargement. Dans on premier temps, on sort le vérin de transfert B pour à décaler le convoyeur d'une position vers la droite. Ensuite, le vérin A sert au chargement d'une nouvelle bouteille vide.

- Le poste 2 :

Sert au remplissage des bouteilles à l'aide de la vanne D.

Le poste 3 :

Est le poste de bouchage.

Les actions de chargement d'une bouteille, remplissage d'une bouteille et bouchage d'une bouteille sont effectuées en parallèle. Le transfert par le vérin B n'est effectué que lorsque ces trois opérations sont terminées.

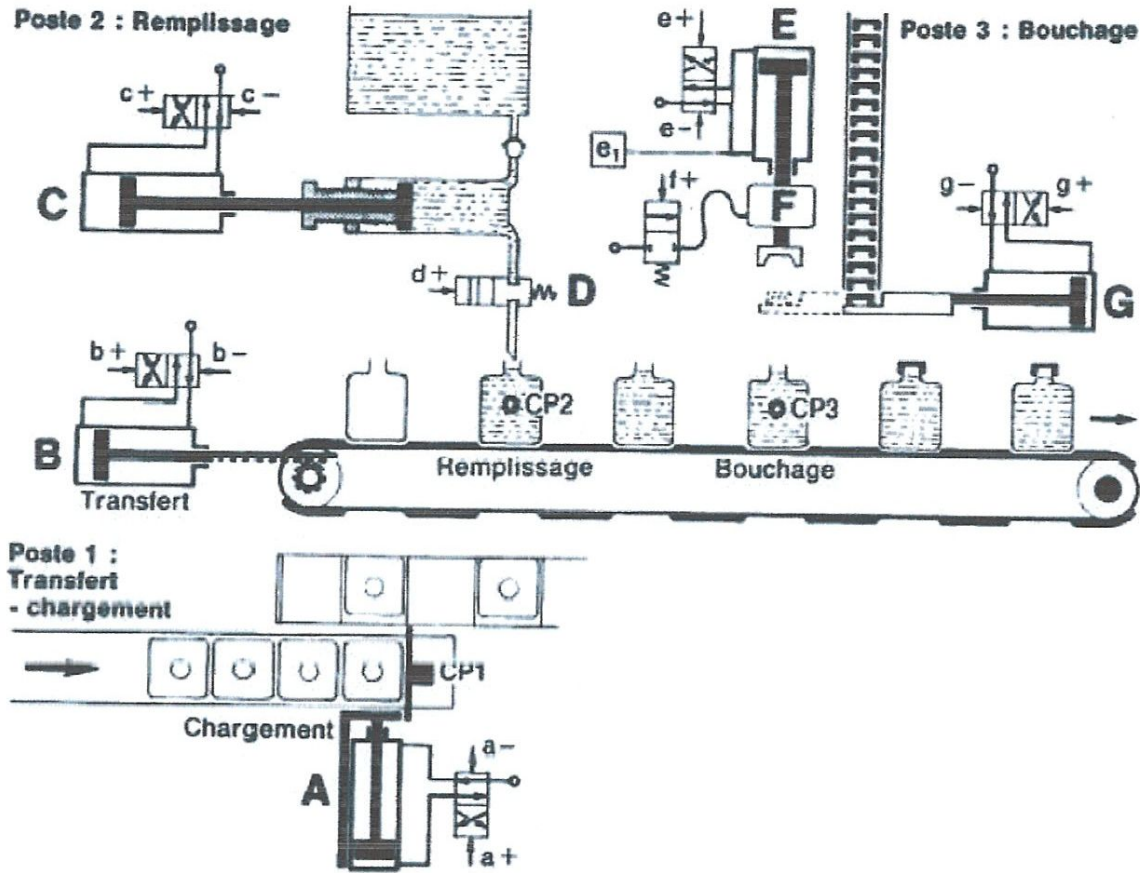


Figure 4.18 : La machine à remplir et à boucher.

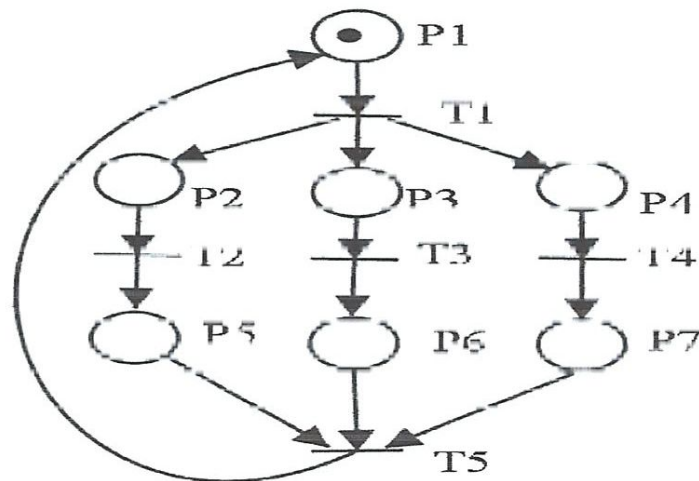


Figure 4.19 : RDP machine à remplir et à boucher.

III.2.2 - Représentation et simulation de modèle:

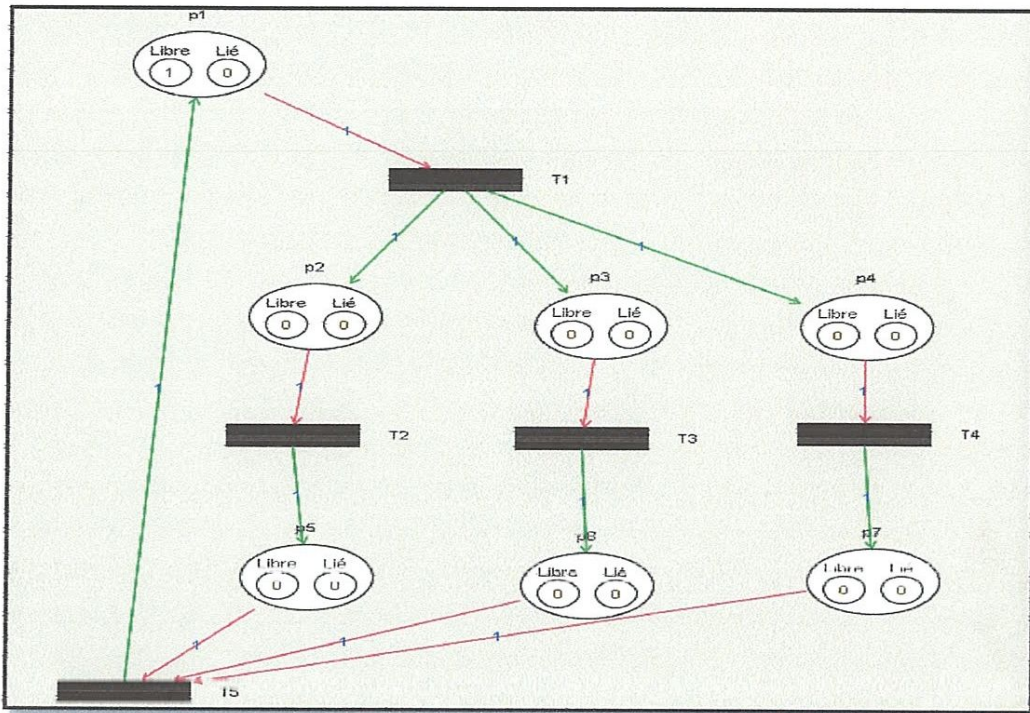


Figure 4.20 : La représentation graphique de modèle.

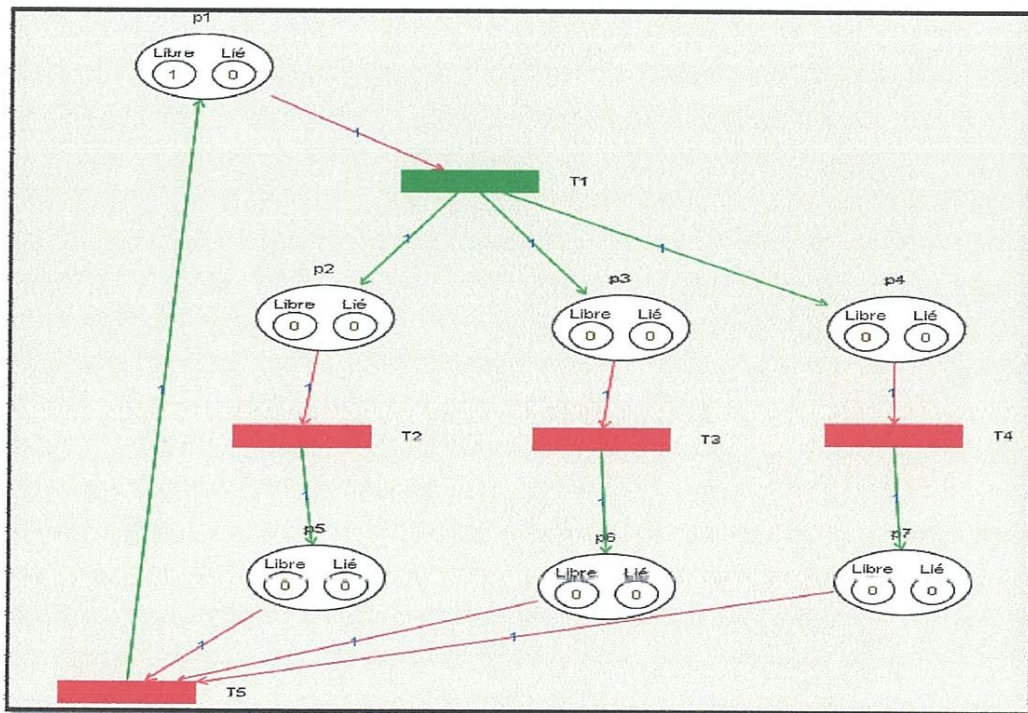


Figure 4.21 : La simulation de modèle.

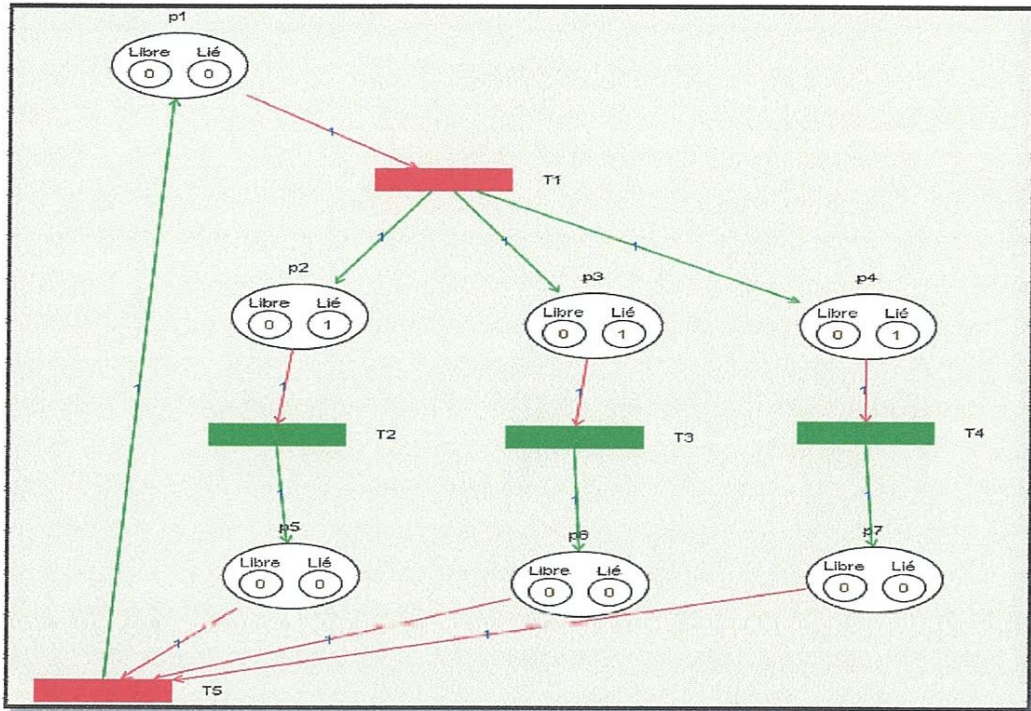


Figure 4.22 : Franchissement de T1.

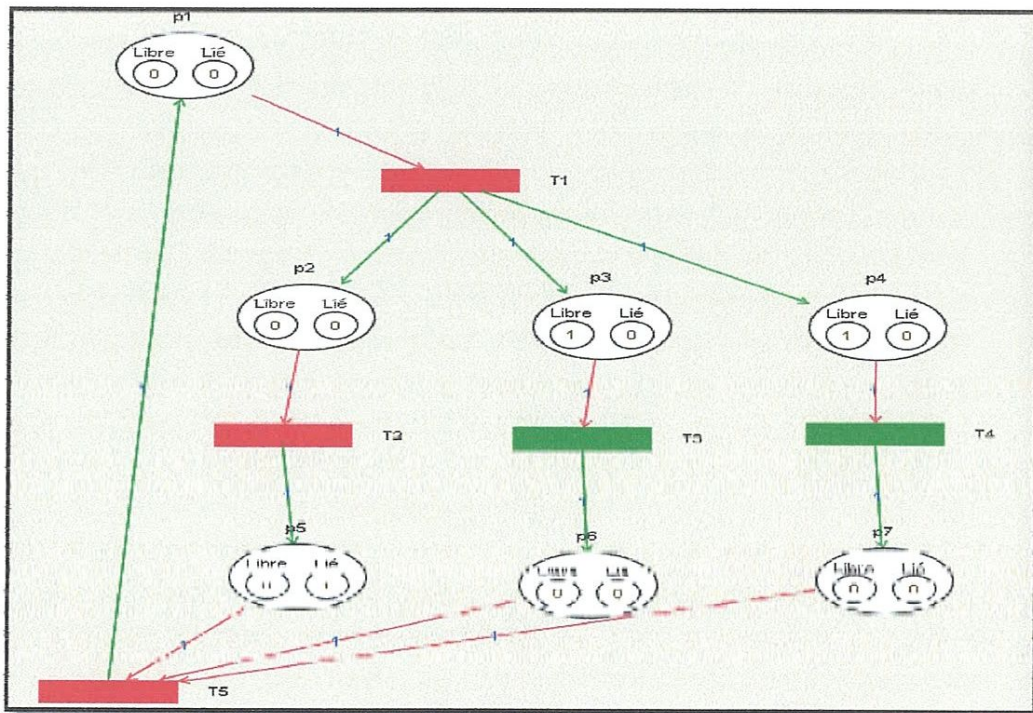


Figure 4.23 : Franchissement de T2

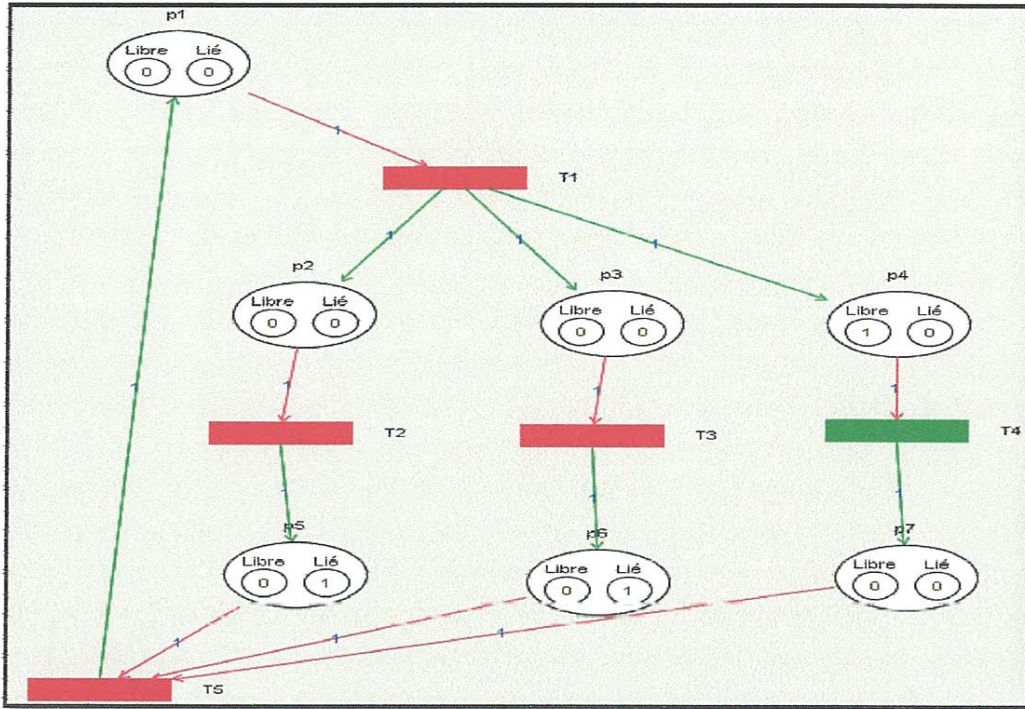


Figure 4.24 : Franchissement de T3.

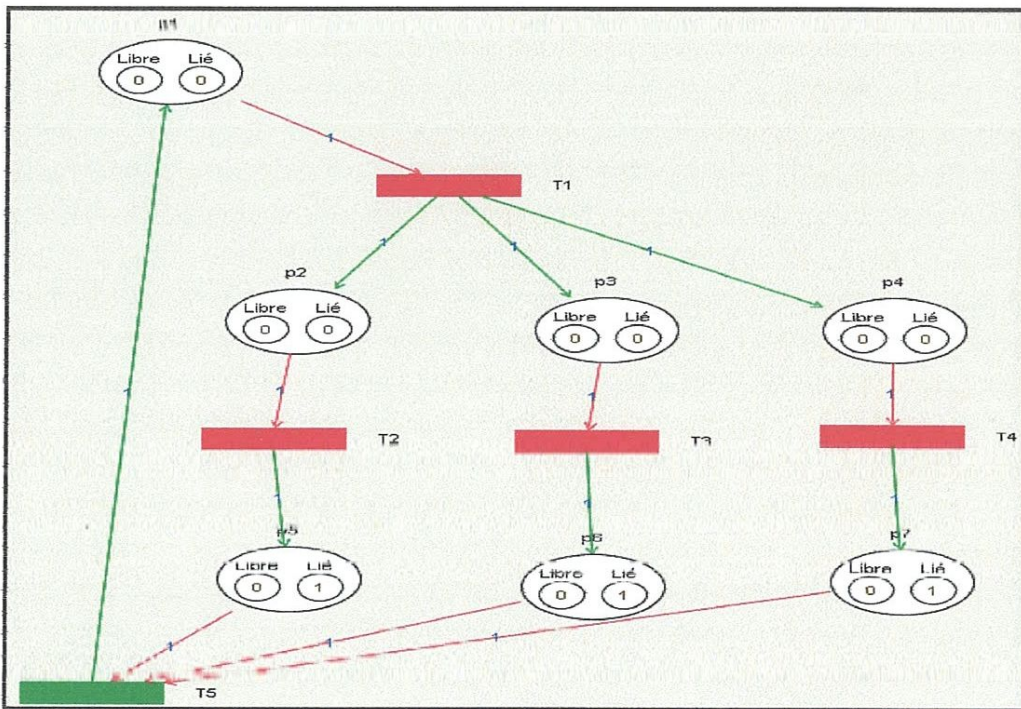


Figure 4.25 : Franchissement de T4.

III.3 - Le Problème des philosophes :

III.3.1 - Description :

Le problème des philosophes et des spaghettis est le cas d'école concernant le problème classique de partage de ressources en informatique système. Il concerne l'ordonnancement des processus et l'allocation des ressources à ces derniers. Ce problème a été énoncé par Edsger Dijkstra.

La situation est la suivante :

- cinq philosophes (initialement mais il peut y en avoir beaucoup plus) se trouvent autour d'une table.
- chacun des philosophes a devant lui un plat de spaghetti.
- à gauche de chaque plat de spaghetti se trouve une fourchette.



Figure 4.26 : Le problème des philosophes.

Pour manger, un philosophe a besoin de deux fourchettes : celle qui se trouve à gauche de sa propre assiette, et celle qui se trouve à droite (c'est-à-dire les deux fourchettes qui entourent sa propre assiette).

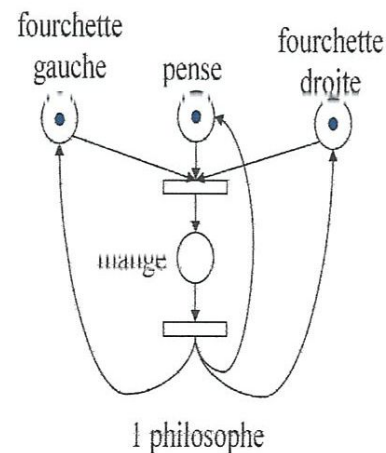
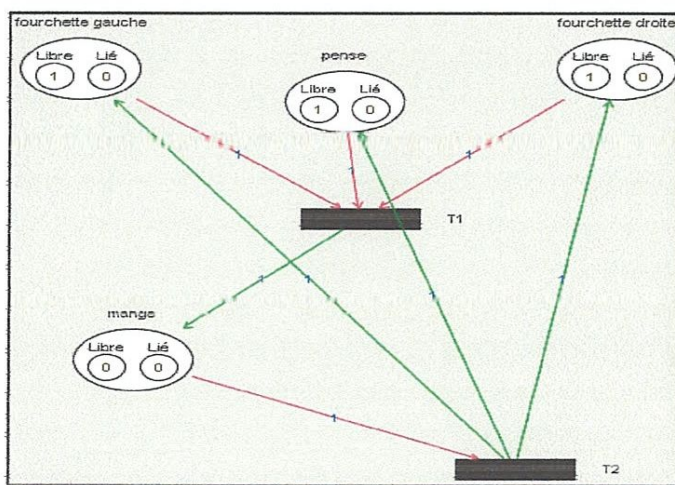


Figure 4.27 : RDP d'un seul philosophe.

III.3.1 - Représentation et simulation de modèle:

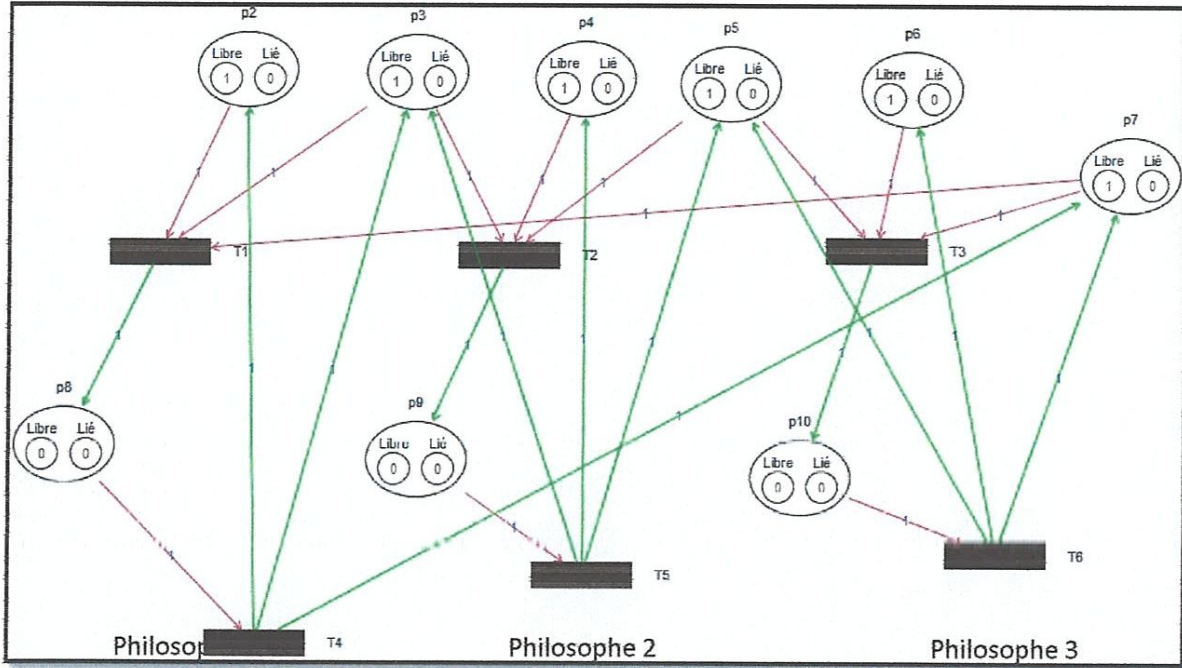


Figure 4.28 : RDP de trois philosophes.

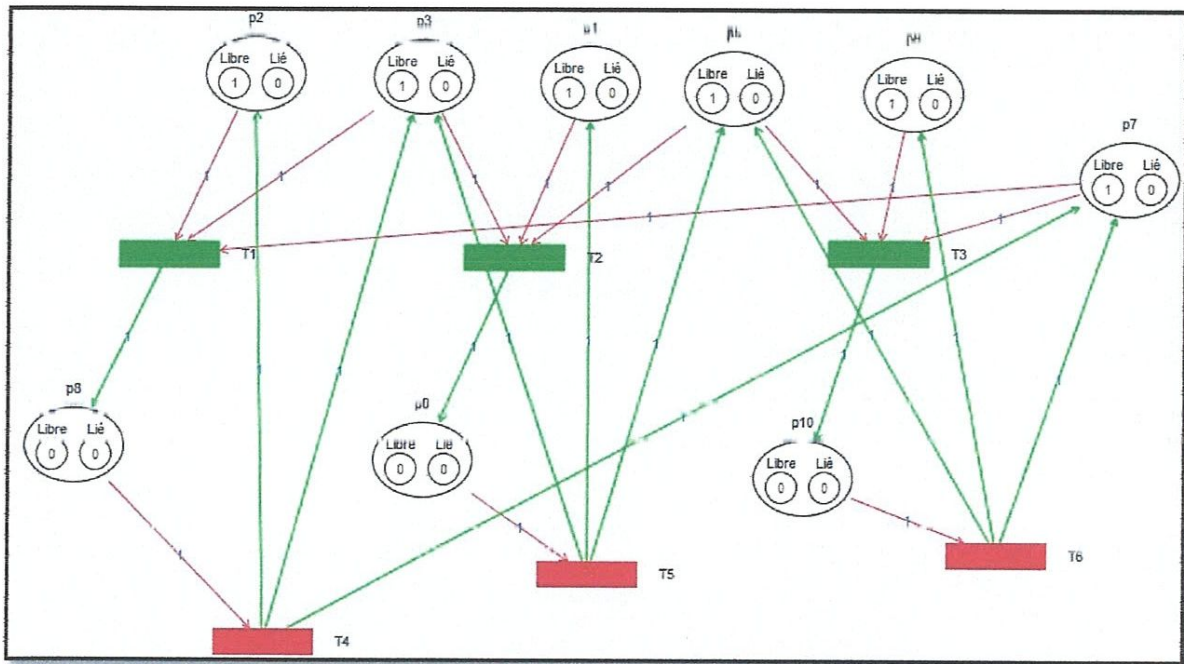


Figure 4.29 : Simulation de RDP de trois philosophes.

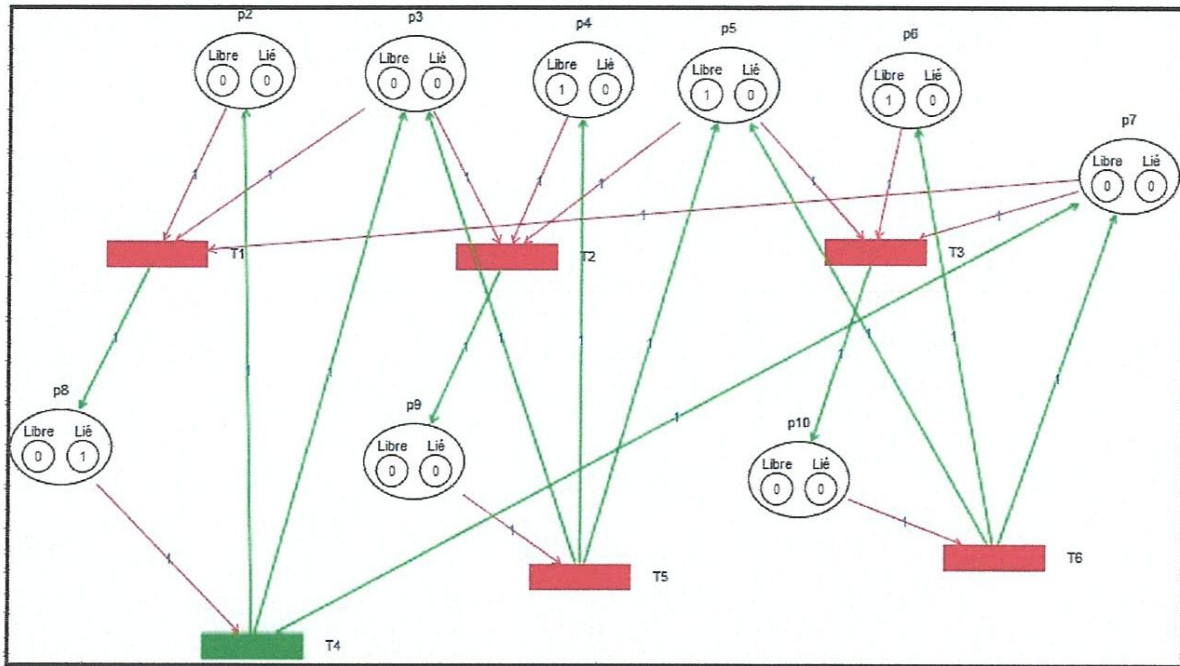


Figure 4.30 : Franchissement de T1 «le premier philosophe mange».

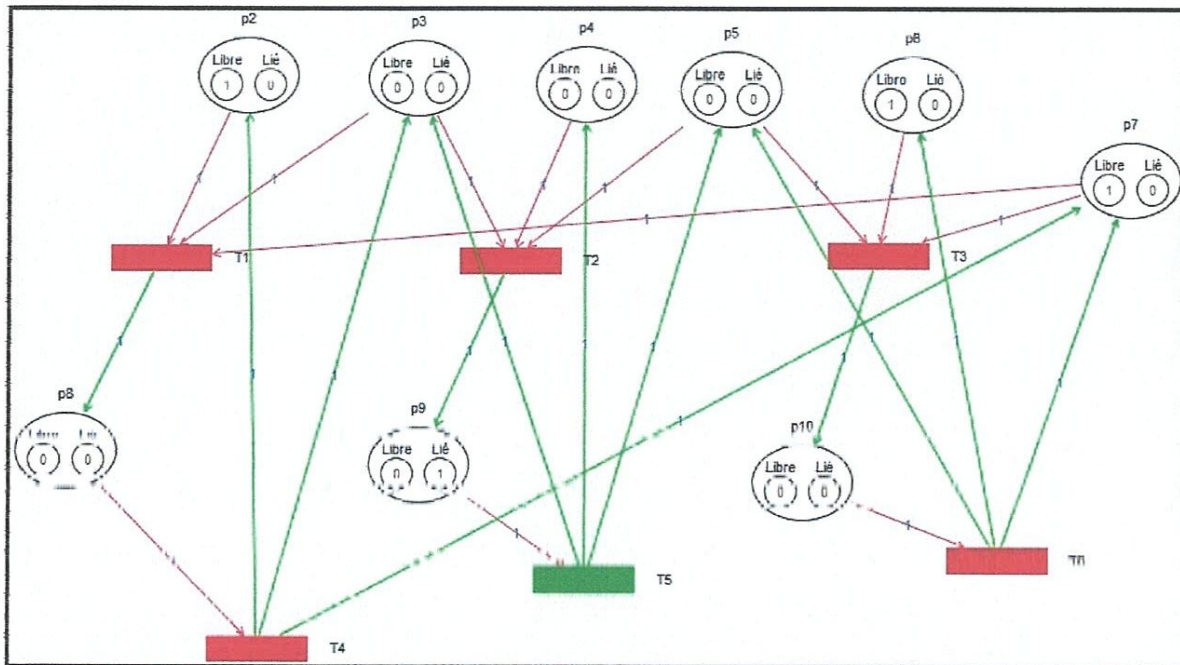


Figure 4.31 : Franchissement de T2 «le deuxième philosophe mange».

IV- Conclusion :

La réalisation de projet qui est un environnement de modélisation visuelle pour réseaux de pétri basé sur la sémantique de maximalité s'inscrit dans une chaîne de développement d'applications critiques.

D'après ces études on a conclu que la modélisation par les réseaux de pétri facilite la représentation des modèles et permet aussi de donner une vérification formelle de ces modèles basé sur une sémantiques bien défini.

Conclusion Générale

Le projet élaboré dans ce mémoire, s'articule autour de le réseau de pétri qui ont été développés pour permettre la modélisation des systèmes à variables d'entrée, de sortie et d'état discrètes.

Notre objectif consiste à développer un environnement de modélisation visuelle pour réseaux de pétri englobant les tâches suivantes

- la représentation graphique de RDP.
- La simulation du fonctionnement de RDP en utilisant la sémantique de maximalité comme un modèle sémantique sous-adjacent.

Le premier but qu'on voulait montrer dans ce projet c'est de bien définir la modélisation par réseau de pétri on ouvre a l'utilisateur le pouvoir de réaliser une représentation graphiques respectant les règles de RDP.

Avant tout, pour atteindre notre objectif, on a choisi l'outil JGraphX qui est un nouveau outil pour ces raisons :

- Squellette d'application directement utilisable.
- Facilité de compréhension.
- Facilité de développement de nouvelles fonctions.

Et comme nous avons travaillé avec les graphes on a besoin d'un autre outil qui est JDOM pour traiter des graphes (ouverture et l'enregistrement).

Le deuxième but qu'on voulait montrer dans ce projet et de réaliser un simulateur basé sur la sémantique du maximalité.

Avant de clore notre mémoire, nous voudrais donner quelques perspectives de continuation de nos travaux. Et comme notre projet et un simulateur on propose pour compléter ce travail de développer un générateur a base de STEMs.

Bibliographie

- [HM80]: M. Hennessy and R. Miner, On observing nondeterminism and concurrency, In “Proc 7th ICALP”, pages 299—309, Springer-Verlag (July 1980).
- [Hoa85]: C. A. R. Hoare. “Communicating Sequential Processes”. Prentice Hall (1985).
- [ISO88]: ISO8807, LOTOS, a formal description technique based on the ordering of observation behavior, ISO (November 1988).
- [Jarras95]: Imed Jarras Vérification et synthèse d’un réseau de pétri relativement à une spécification logique temporelle, Université LAVAL, Juillet 1995.
- [Kurt197]: Jensen Kurt. Coloured petri nets: Basic concepts, analysis methods and practical use, volume 1, Springer, 1997.
- [MER74] : MERLIN P. M., A study of the recoverability of computing systems, PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.
- [Mil80]: R. Milner. “Communication and Concurrency”, volume 92 of LNCS. Springer Verlag (1980).
- [Mil83]: R. Milner. Calculus for synchrony and asynchrony TCS 25, 267—310 (1983).
- [Mil89]: R. Milner. “Communication and Concurrency”. Prentice Hall (1989).
- [RAM74]: RAMCHANDANI C., Analysis of asynchronous concurrent systems by timed Petri nets, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1974, Project MAC Report MAC-TR-120.

Bibliographie

[Rei85]: W. Reisig. Petri nets. In “EATCS Monographs on Theoretical Computer Science”, Berlin, Heidelberg, New York, Tokyo (1985). Springer Verlag.

[Sai96]: D. E. Saidouni. Sémantique de maximalité: Application au raffinement d’actions en LOTOS. PhD thesis, LAAS-CNRS, 7 av. du Colonel Roche, 31077 Toulouse Cedex France, 1996.

[Shi85a]: M. W. Shields. Concurrent machines. The Computer Journal 28(5), 449—465 (1985).

[Tra05]: TRAN Vinh Duc, Rapport final de TIPE, Institut de la Francophonie pour l’Informatique - Promotion 10, 15 juillet 2005.

[Geffroy]: J-C. Geffroy, Introduction aux réseaux de pétri,
www.cyrnac.free.fr/pub/cours/cnamMOCA/CH10.pdf

« Visité le : 16.05.2012 ».