Introduction générale

Avec la mondialisation et la globalisation des échanges marchands, l’unité de production doit aujourd’hui faire face à plusieurs contraintes, comme les délais, la qualité et la complexification des produits, etc. Ces différentes contraintes rendent les systèmes de productions très complexes. Face à ces contraintes, les études d’évaluation des performances des systèmes de production est devenue obligatoire pour les industriels et ceci pour des raisons aussi bien de coût que de productivité. Ces études doivent être réalisées dès la phase de conception. Aussi, l'évaluation des performances est une étape clé dans n'importe quelle étude de conception. En se basant sur les résultats de l'évaluation des performances, il est possible de prendre des décisions et de réaliser des compromis pour l'optimisation du système de production global (**BENFARHI A**)

Maintenant que l'offre est supérieure à la demande dans plusieurs secteurs de production, les entreprises doivent être capables de s'adapter le plus rapidement possible aux changements de la production afin de satisfaire les exigences des consommateurs. L'évolution des technologies, comme l'informatique ou la robotique, a permis la naissance de nouveaux types de systèmes de production qui répondent bien à ces exigences : les systèmes flexibles de production (SFP). L'objectif principal de ces derniers est de concilier à la fois efficacité et flexibilité qui sont généralement considérées comme des critères contradictoires.

Le cycle de développement d'un atelier flexible est un processus extrêmement complexe constitué d'un ensemble d'étapes parmi lesquelles on trouve la spécification des produits à fabriquer, la conception et la mise en marche du système. Le principal problème, lorsque l'on se place dans le contexte des systèmes de production, est d'être capable de proposer une méthodologie complète de spécification, de modélisation, d'analyse et de contrôle de ces systèmes. L'utilisation d'une telle méthodologie permet de réduire, voire éviter, les coûts supplémentaires induits par la remise en cause ou la correction de la conception lors de la phase de mise en œuvre.

L’influence de cette métamorphose dans l’industrie se traduit par la nécessité d’avoir des systèmes capables de s’adapter aux changements de production. Ces systèmes doivent se révéler flexibles et robustes afin de répondre aux exigences en diversité, productivité et notamment en assurant la qualité exigée. Les coûts d’exploitation doivent aussi être optimisés tout en diminuant les risques des défaillances.

Aussi pour réaliser une étude d’évaluation des performances, Il est nécessaire de posséder une représentation du système à étudier, c'est-à-dire un modèle. L’extraction des paramètres de performances se fait alors soit par simulation, soit par analyse. Dans notre travail, nous ne nous intéressons qu’aux méthodes analytiques, qui sont des outils d’évaluation des performances permettant une simplification du modèle et proposant des évaluation du système Même si l’industrie dispose d’un système de production flexible et robuste qui peut répondre aux besoins du marché, la négligence du facteur temps aura des conséquences sur les performances et par conséquent sur la rentabilité.

Dans la littérature, plusieurs outils correspondant à ces exigences. Parmi eux on trouve, les Réseaux de Pétri Stochastique (RdPS) ces derniers sont largement utilisés dans la Système de production (SdP). En effet, ils offrent à la fois une puissance de description des systèmes et une richesse de traitement et d’interprétation des modèles résultants. De plus, comme ils intègrent un aspect stochastique, ils permettent de prendre en compte des notions aussi aléatoires tel que les taux de panne d’une machine.

Le travail que nous présentons dans ce mémoire a comme objectif de présenter les aspects aléatoires sur les systèmes de production et de développer un modèle stochastique considérant ces aspects aléatoires fixés afin d’exploiter le modèle généré pour l'analyse de performances des systèmes de production étudiés.

En effet ce mémoire est divisé en quatre chapitres:

En premier lieu, nous réalisons un état de l’art cernant le fonctionnement des systèmes de production et particulièrement l'impact de l'aspect aléatoire sur ces derniers

En deuxième lieu, nous allons introduire un nouveau modèle formel permettant la spécification d’une large classe des systèmes flexibles de production en tenant en compte de l'aspect aléatoire.

En troisième lieu, nous allons développer un modèle stochastique considérant les aspects aléatoires fixés.

En dernier lieu, nous allons exploiter le modèle généré pour l'analyse de performances des systèmes de production étudiés

**Les aspects aléatoires dans les systèmes de production flexibles**

1. **Introduction**

Ce premier chapitre est consacré, dans une première partie, à la présentation d'un état de l'art concernant le fonctionnement des systèmes de production et particulièrement l'impact des aspects aléatoires sur ces derniers. Dans une deuxième partie, ce chapitre présente différentes approches de modélisations proposées dans la littérature pour représenter les différents aspects de fonctionnement des systèmes de production flexibles. La synthèse des travaux existants permettra de justifier le choix de notre approche de modélisation.

1. **Système de production**

Un système de production (SdP) est un système artificiel composé d’un ensemble d’éléments matériels et immatériels qui interagissent et interfèrent à la production de biens ou de services par une entreprise. Pour évaluer la performance d’un SdP, il est important de connaître sa structure, sa composition, ses différentes fonctions, sa nature, ses conditions d’exploitation et son environnement. Dans ce qui suit nous allons présenter les domaines d’application des SdP ainsi que les influences des aspects aléatoires sur ces derniers.

* 1. **Domaine d’application des systèmes de production**

Les systèmes de production couvrent plusieurs domaines d’application tels que : les systèmes informatiques, les systèmes manufacturiers, les systèmes hospitaliers et les services.

Le tableau 1 donne quelques exemples de domaine d’application, de leurs ressources et de leur opération (Caumond, 2006).

|  |  |  |
| --- | --- | --- |
| Domaine d’application | Type de ressource | Type d’opération |
| Système informatique | Microprocesseur, bus, interface homme/machine, console,… | Traitement, sauvegarde sur disque, lecture d’un fichier |
| Système manufacturier | Machine, opérateur, outils, moyen de transport,… | Transformation, assemblage, transport, maintenance,… |
| Système hospitalier | Docteurs, infirmières, lits, bloc opératoire, équipements,… | Opération chirurgicale, accueil du patient, diagnostic … |
| Service | Guichetier, hôtesse d’accueil,… | Traitement de la demande, redirection vers un autre service … |

**Tableau 1:Exemples de domaines d’applications, de leurs ressources et de leurs opérations**

Le domaine le plus utilisée en système de production est celle des systèmes manufacturiers. Nous nous intéresserons dans la suite de ce mémoire à ce type de système et en particulier aux systèmes de production flexible.

* 1. **Présentation des systèmes de production**

D’après (Danø, 1966), la production est définit par une succession d’opérations en passant de l’état de matières premières (composants entrant dans le système) à l’état de produits finis. Les opérations peuvent être de différents types : opérations de transformation, d’assemblage, de désassemblage, de transfert etc. Selon (Askin & Standridge, 1993), « la production est le passage d’une conception à un produit fini». Plus précisément un système de production est un espace constitué de :  
•Moyens de production (ressources) : les ressources du système de production sont l’ensemble des moyens mis en œuvre capables de réaliser diverses opérations pour transformer les entités (il peut s’agir de matières premières, de composants, de produits semi-finis ou de produits finis) pour passer de la matière première au produit fini. Ces ressources peuvent être des machines (stations), des zones de stockage, des moyens de transport, des palettes, des outils etc.,  
•Opérateurs (OP) : les opérateurs sont les types d’opérations que les ressources peuvent réalisées. Les opérations peuvent être de différents types : opérations de transformation, d’assemblage, de stockage, de transport etc.

De plus, un système de production est soumis à une charge qui définit l’ensemble des biens ou des services que le système doit réaliser. Pour écouler cette charge, le système utilise une ou plusieurs ressources et doit respecter certaines contraintes temporelles (Caumond, 2006). La charge d’un système de production est divisée en « job ». Un travail (Job) suit la réalisation d’un produit fini dans toutes les étapes de production (de la matière première jusqu’au produit fini). Chaque travail a une gamme qui décrit la suite des opérations qu’il doit réaliser. La gamme décrit les opérations à réaliser en donnant:  
•Les produits consommés (les produits entrants) et les produits sortants.  
•L’ordre entre les opérations à réaliser.

Mais la gamme ne donne pas:  
•La ou les ressources qui doivent réaliser les opérations (on ne connaît pas les ressources à l’avance).  
•Leur durée (on ne connaît pas la durée de la gamme et des opérations à l'avance).

* 1. **Classifications des systèmes de production**

On peut classer les systèmes de production selon les aspects de productivité et de flexibilité alors on trouve 4 types:

•Systèmes Manufacturiers Dédiées (SMD)  
•Systèmes Manufacturiers Cellulaires (SMC)  
•Systèmes Manufacturiers Flexibles (SMF)  
•Système Manufacturiers Reconfigurables (SMR)

* + 1. **Systèmes Manufacturiers Dédiées (SMD)**

Les lignes de fabrication sont conçues pour la fabrication en grande série d’un type de produit unique selon une gamme linéaire .Dans ce type de système, les machines sont organisées suivant la gamme de fabrication. Ce type de système a un avantage et un inconvénient. Son avantage se résume au fait que les lignes de fabrication ont un taux de production très élevé. L’inconvénient est qu’un seul type de produit ou une famille de produits, qui se diversifient légèrement les uns des autres, peut être fabriquée dans ces systèmes.

* + 1. **Les systèmes Manufacturiers Cellulaires (SMC)**

Les systèmes de production cellulaires fabriquent les produits dans des zones dédiées appelées cellules. Toute cellule regroupe les différents équipements nécessaires pour la fabrication d’un type de produit.

* + 1. **Systèmes Manufacturiers Flexibles (SMF)**

Les SMF ont été introduits dans les années 1980, pour répondre non pas aux volumes et aux variétés de production, mais pour répondre également aux nouvelles technologies de production et aux besoins des marchés. Ils sont donc des systèmes intégrés de moyens de production performants et de haute précision qui interagissent et interfèrent dans le but de réaliser un ensemble de types de produits, qui peut changer avec le temps (types, quantités), tout en assurant le compromis coût/performance et en respectant la qualité de la production (Berruet & Kanso, 2010). L’obstination (robustesse) de ce type de système a été remarquable grâce à la flexibilité introduite, néanmoins leur coût est trop élevé. Les SMF visent à rendre flexible l’ensemble de l’outil de production. C'est-à-dire, le préparer à s’adapter aux divers changements de son environnement, sans qu’il y ait besoin d’engager de nouveaux investissements en biens d'équipement, ou d’engendrer de longues pertes de temps. Sa finalité est de réaliser une multitude d’opérations hétérogènes à partir d’un nombre très limité de ressources. Cette vision desdits systèmes (SMF) est très fréquente dans l’industrie manufacturière où une installation est constituée d’un ensemble de stations (machines de production) reliées par des zones de stockage et des moyens de transport. Le produit est acheminé d’une zone (machine ou zone de stockage) à une autre en subissant des transformations réalisées dans un temps prédéfini. Un SMF est un système de production avec une flexibilité générale.

* + 1. **Systèmes Manufacturiers Reconfigurables (SMR)**

Ce nouveau concept de système de fabrication apparaît dans les années 2000 dans le but de fournir la productivité et la flexibilité, au bon moment et au meilleur coût (coût réduits) en intégrant de nouveaux principes (Aladad, 2009) comme :

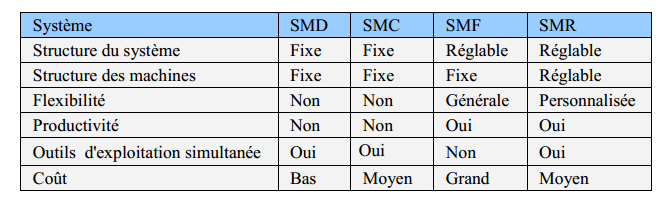
•La modularité tant sur le plan de la conception mécanique, logiciels et matériels par l’intégration des nouvelles technologies.

•La convertibilité, par l’aptitude de mise en course rapide.  
•La diagnosticabilité, permet de mesurer l'état d'un système à fin de diagnostiquer toute faute du système dans un délai fini et de rendre le système de fabrication disponible à la production.  
•La personnalisation en fonction des produits.

Ce nouveau paradigme concernant les systèmes manufacturiers « reconfigurables » sont conçus pour répondre efficacement aux changements rapides du marché, en tenant compte des contraintes à priori contradictoires telles que la haute flexibilité, la haute productivité, le coût et la qualité. Un SMR est un système de production avec une flexibilité mesurée.

* + 1. **Caractéristiques des systèmes de production**

Le tableau suivant présente une comparaison des différents types de systèmes de production en fonction d'un certain nombre de contraintes.



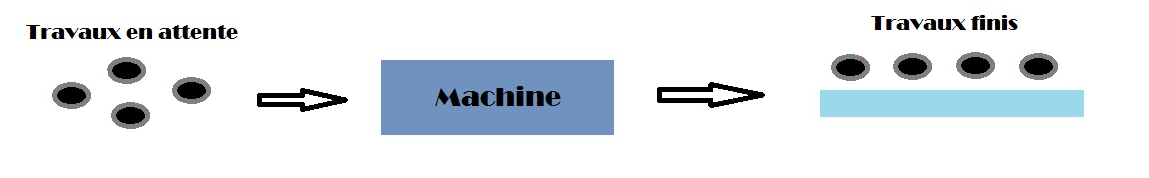
**Tableau 2: une comparaison entre les différents types des systèmes manufacturiers**

1. **Organisation des Systèmes de production flexible**

Les systèmes de production flexible sont classés en fonction du volume de production et du nombre de pièces produites. Une grande variété de produits traités provoque une diminution du volume de production.

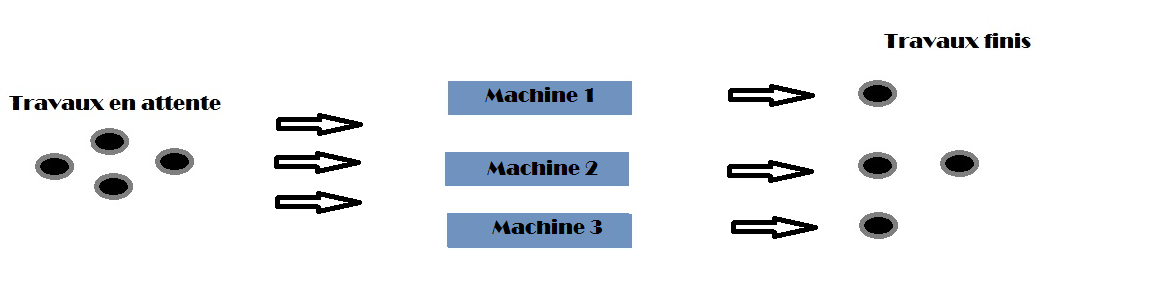
* 1. **Le système de production à Machine unique**

Dans ce système, il n’existe qu’une seule ressource (machine) et les tâches (jobs) sont mono-opération la ressource ne peut exécuter qu’une seul tâche à la fois. La figure 1 montre un exemple d’un atelier à machine unique avec quatre travaux.

  
**Figure 1 : Machine unique**

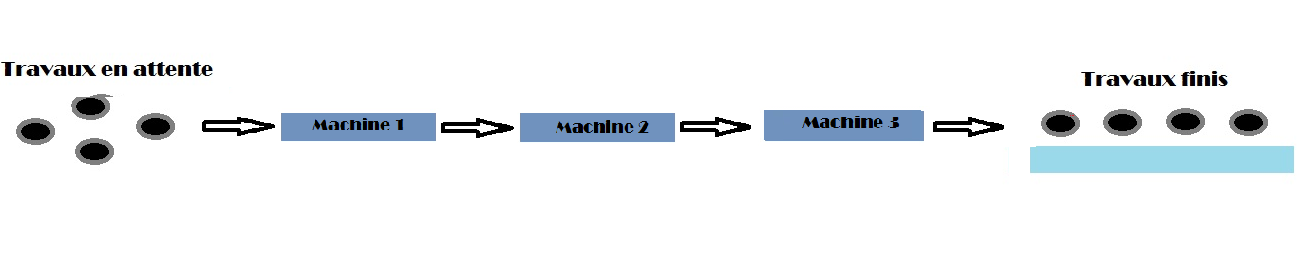
* 1. **Le système de production à Machine parallèles**

Les tâches sont mono-opération mais il existe plusieurs ressources identiques susceptibles de les exécuter. La figure 2 montre un exemple d’un atelier à cheminements unique avec quatre travaux et trois machines.

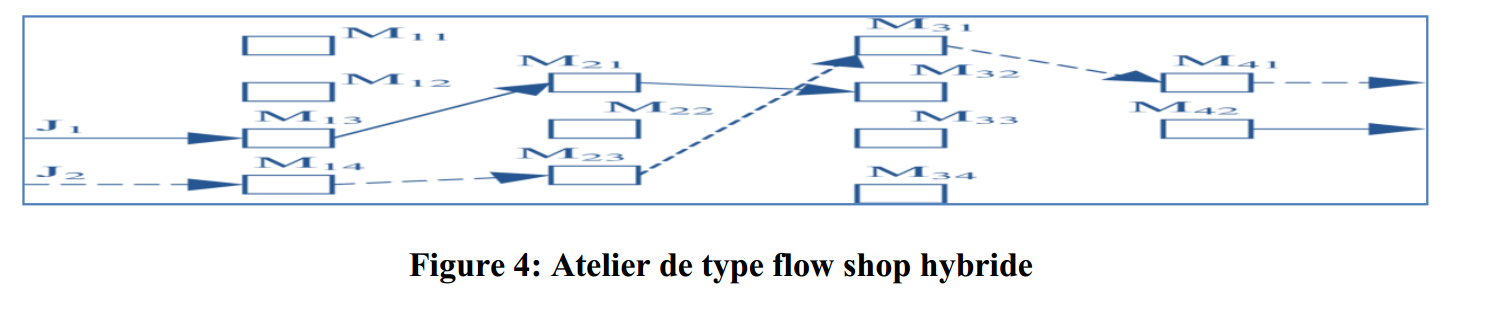
****

**Figure 2 : Machine parallèles**

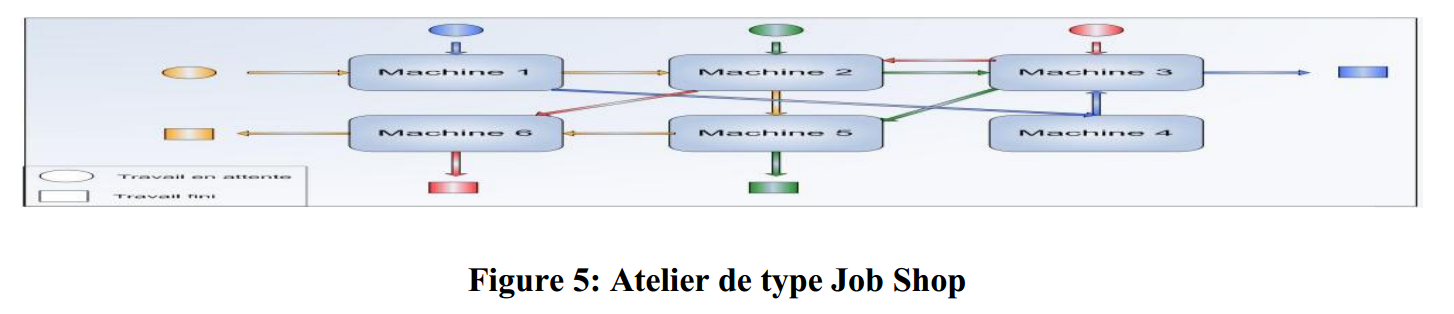
* 1. **Le système de production à cheminement unique (flowshop)**

Les tâches sont multi-opération et leur exécution nécessite l’utilisation de plusieurs ressources dans le même ordre. Chaque ressource n’existant qu’en un seul exemplaire. La figure 3 montre un exemple d’un atelier à cheminement unique avec quatre travaux et 3 machines.    
**Figure 3 : Atelier de type flow shop**

**Flow Shop hybride** : Se dit lorsque les ateliers, composés de ressources parallèles, sont organisés en flow shop. Les produits sont multi-opération, chaque opération étant réalisée sur une des machines de l’atelier. La figure 4 montre un exemple d’un atelier flow shop hybride avec deux travaux et quatre machines, chaque machine existe en plusieurs exemplaires.



* 1. **Le système de production à cheminements multiples (Job Shop)**  
     Les tâches sont multi-opération et l’exécution d’une tâche nécessite l’utilisation de plusieurs ressources dans un ordre qui lui est propre. Chaque ressource n’existant qu’en un seul exemplaire. La figure 5 montre un exemple d’un atelier à cheminements multiples avec quatre travaux et six machines.



**Job Shop hybride** : Se dit lorsque les ateliers, composés de ressources parallèles, sont organisés en job shop. Les produits sont multi-opération, chaque opération étant réalisée sur une des machines de l’atelier.

* 1. **Le système de production à cheminement libre (open shop)**

Les tâches sont multi-opération et leur exécution nécessite l’utilisation de plusieurs ressources dans n’importe quel ordre(les opérations des tâches sont indépendantes). Chaque ressource existe en un seul exemplaire. Plus précisément, chaque produit à traiter doit subir un ensemble d’opérations sur un ensemble de machines, mais dans un ordre totalement libre.

Dans les ateliers de type **open-shop hybride**, une machine peut exister en plusieurs exemplaires identiques fonctionnant en parallèle pour une grande demande, le système flow-shop est le plus adopté, mais avec une variété réduite de produits. Pour une grande gamme, le système de production job-shop est plus efficace pour un volume de production réduit. Dans ce qui suit, nous travaillons sur les ateliers les plus complexes qui sont les ateliers job shop hybride. Nous allons supposer que les ressources ne sont pas uniquement les machines comme s’est présenté précédemment dans la littérature ,mais peuvent être aussi des moyens de transport et des zones de stockage.

1. **Aspect aléatoires dans les systèmes de production**

Tout au long de leur vie opérationnelle, les systèmes industriels simples ou complexes sont soumis aux fluctuations issues d’origines multiples et variés : indisponibilité des systèmes due aux arrêts propres et induits, perte de performance due aux ralentissements et aux écarts de cadence, perte de qualité. Ces fluctuations entrainent une perte de leur efficience, engendrant une baisse de rendement et une augmentation des coûts indirects (maintenance, reconditionnement, reconfiguration…). (Timothée K ,2011)

La bonne tenue de ces systèmes dépend étroitement de leur efficience caractérisée par un ensemble d’indicateurs de performance permettant d’avoir une vision plus globalisante du rendement et des moyens engagés. L’évaluation de performance de tels systèmes industriels par une étude comportementale nominale et dysfonctionnelle de leurs ressources élémentaires (humaines, organisationnelles, techniques) permet de disposer d’indicateurs décisionnels qui assureront une optimisation des politiques de maintenance et de production.

Évaluer la performance globale d'un système de production de biens est un problème ardu qui nécessite la prise en compte de la bonne interaction de ses différents constituants (humains, organisationnels, techniques)  
  
Avec la complexité croissante des systèmes industriels et l'importance que l'on attache à leur capacité à fonctionner correctement et d'une manière continue, le besoin de modéliser fidèlement leurs comportements fonctionnel et dysfonctionnel pour ensuite évaluer leur performance globale, se fait de plus en plus pressant.  
  
Nous présentons dans ce mémoire une méthode de modélisation et dévaluation d’un système de production flexible se basant sur les réseaux de Petri Stochastique cette méthode considère l’aspect panne qu’elle modélise comme aspect aléatoire. En se basant sur cette méthode, les performances de notre système de biens es evalué selon une la modélisation comportementale (temporelle et stochastique) complétée par la simulation des dysfonctionnements.

1. **Modélisation**

D’après (Boimond & Jean-Louis) l’étape de modélisation est une phase essentielle à la simulation (expérimentation), à la validation, et à l’évaluation du système de production. La modélisation consiste à créer une représentation simplifiée d'un système pour résoudre un problème le concernant: le modèle. Grâce au modèle il est possible de représenter simplement un système, un concept et de le simuler.

* 1. **Outils de modélisation**

Dans le but de justifier le choix de l’outil que nous avons adopté pour représenter les systèmes de production, nous allons présentons une synthèse des principales caractéristiques des outils les plus utilisés dans ce domaine, à savoir les machines à états, les automates, les statecharts, les réseaux de file d’attente,Réseaux d’automates stochastiques et les réseaux de Petri. Nous préciserons les points forts et faibles de chacun et nous les comparerons à nos besoins. Cette procédure nous permet de mettre en évidence les raisons de nos choix et les contraintes que nous devrons imposer pour respecter les limites de l’outil choisi (Esteban, Nketsa, & Combacau, 2005) :

**Les machines à états** : les machines à états sont fréquemment utilisées pour décrire les systèmes à événements discrets car il est possible de découper ce type de système en plusieurs processus séquentiels. Ils permettent de capturer sans difficulté le comportement séquentiel d’une application. Parmi les désavantages des machines à états nous pouvons citer le problème de l’explosion combinatoire quand le système devient plus complexe et la difficulté qu’il existe à représenter des évolutions simultanées (Esteban, Nketsa, & Combacau, 2005).

**Statechart** : d’après (Harel, 1987) et (Esteban, Nketsa, & Combacau, 2005), les statecharts sont présentés comme une extension du concept de machine à états adapté à la spécification des systèmes à événements discrets complexes. Ils incluent les concepts de parallélisme, d’abstraction, de hiérarchie, d’orthogonalité et de synchronisation des états. Les statecharts permettent de modéliser simplement les mécanismes de préemption sur un groupe d’états et comportent un mécanisme d’historique permettant de replacer le modèle dans le dernier état qui avait été atteint avant une préemption. Les statecharts sont à considérer pour des applications comportant ces mécanismes, mais il faut savoir que la mise en œuvre d’un statechart est un problème non résolu à l’heure actuelle, aussi il y a une difficulté de la validation des synchronisations.

**Réseaux de Petri** : d’après (Chemla), les réseaux de Petri (RdP) permettent de modéliser des systèmes séquentiels. Ils sont reconnus comme étant un outil puissant de modélisation des parallélismes, des conflits et des partages de ressources en présence de synchronisation. Parmi les désavantages des RdP, nous pouvons citer le problème de l’explosion combinatoire quand le système devient plus complexe, par conséquent, sa représentation graphique peut devenir peu lisible.

**Réseaux de files d'attente** : la motivation première de ces modèles, était la modélisation des systèmes informatiques. Ensuite, ils ont été adaptés pour les systèmes de production. Les réseaux de Files d’attente permettent de modéliser et d’analyser les systèmes de type client/serveur, c’est un réseau de stations interconnectées. D’après (Nana, Legrand, Singhoff, & Marce, 2003), il peut toutefois être difficile de représenter certains systèmes à l’aide de files d’attente. Le développement de modèles hiérarchiques avec différents niveaux de détail n’est pas simple avec cette approche.

**Réseaux d’automates stochastiques :** introduit au début des années 80 par Brigitte Plateau. Ces réseaux sont basés sur l'idée que le système modélisé peut se décomposer sous la forme de  
plusieurs sous-systèmes. Chacun de ces sous-systèmes peut évoluer indépendamment et peut  
interagir avec les autres en se synchronisant ou en faisant varier la valeur de ses transitions en  
fonction des états des autres sous-systèmes, ce comportement étant typiquement celui d'un  
système parallèle. Ils sont appliqués principalement dans les systèmes informatiques et de télécommunication parallèles

**Les approches existant :**

De nombreux travaux (Paolo B, 2011) (BENFARHI A, 2012) (Ajmone M, 1993) ([Christoph L](http://dl.acm.org/author_page.cfm?id=81320492528&coll=DL&dl=ACM&trk=0&cfid=559846708&cftoken=65522806),1998) présentent aujourd’hui pour évaluer les performances des systèmes de production mais presque on ne trouve pas qui utilise les réseaux de Petri stochastique coloré pour les systèmes de production flexible.   
En effet BENFARHI A(2012) présente une modelisation du SpP cellulaire avec le formalisme des Réseaux d’automates stochastiques pour l’évaluation de ses performances.  
M. Ajmone Marsan(1993) représente une modélisation générale avec les réseaux de Petri stochastique normale dans plusieurs domaines parmi eux le système de production cellulaire et il a évalué les performances de ce système.   
Paolo B(2011) présente une modélisation du système de production avec le réseau de Petri stochastique simple et il a présenté les propriétés quantitatives et qualitatives du SdP et il a réussi à évaluer les performances du système à l’aide du HSLA (Hybrid Automata Stochastic Logic), Cependant cette modélisation souffre d’une limitation. En effet, la modélisation des systèmes réels mène parfois à des RdP de taille ingérable, ce qui diminue leur lisibilité et rend leur manipulation et même leur analyse difficile.

* 1. ***Choix de l’approche de modélisation***

Le travail demandé dans ce stage est de développer un modèle stochastique considérant les aspects aléatoires fixés pour un système de production flexible, à fin d’exploiter ce modèle généré pour l'analyse de performances des systèmes étudiés.

Pour cela, notre choix s’est porté sur les réseaux de Petri stochastique colorés (RDPSC) pour différentes raisons :

* Leur nature graphique qui offre la convivialité souhaitée, puisqu’ils disposent d’une représentation graphique attrayante, qui accroit la lisibilité et facilite la compréhension des modèles,
* Ils permettent de modéliser les systèmes complexes exemple : chaines de montage, réseaux de communication, architectures clients–serveurs etc. Pour analyser non pas seulement les disfonctionnements éventuels mais aussi les performances telque les nombres des pièces fabriquées, temps d’attente, les taux de bon du fonctionnement etc.
* Ils permettent de représenter les aspects aléatoires ainsi les performances de système
* Ils sont munis d’un grand pouvoir d’expression, qui les rend très adaptés à décrire des comportements complexes, réactifs ou concurrents,
* Ils permettent de représenter relativement simplement les différents concepts de l’algorithmique et de la programmation (séquence, itération, conditionnelle, parallélisme, mécanismes de synchronisation, etc.). Les RdPSC permettent de modéliser l’affectation générale, les structures de données ou les conditions booléennes pouvant apparaître dans les synchronisations,
* La disponibilité d’un grand nombre d’outils pour la simulation et la vérification des modéles étudiés.
* Leur potentiel pour l’analyse mathématique qui permet de vérifier certaines propriétés du système (atteignabilité d’un état, blocages, etc.),
* Le modèle de RdPSC est à objectifs multiples. Il peut servir à la fois pour l’évaluation de propriétés comportementales et pour d’autres besoins tels que la génération systématique de code ou encore l’évaluation de performances**.**

1. **Réseaux de Petri Stochastique (RdPS)**

Il existe plusieurs extensions du RdPS:

* **Réseaux de Petri Stochastiques temporisé** (Stochastic Petri Nets (SPN))  
  dans lequel les temps d’exécution des transitions sont représentés par des variables aléatoires de distributions exponentielles.
* **Réseaux de Petri Stochastiques Généralisés** (Generalized Stochastic Petri Net (GSPN))  
  RdP dans lequel le franchissement de certaines transitions est immédiat, et les temps d’exécution des autres transitions sont représentés par des variables aléatoires de distributions exponentielles.
* **Réseaux de Petri Stochastiques et déterministes** (Deterministic and Stochastic Petri Net (SPN))  
  C’est une extension des GSPN dans laquelle les temps d’exécution des transitions temporisées sont déterministes ou représentés par des variables aléatoires de distributions exponentielles.

***6.2 Réseaux de Petri Stochastique***

Depuis la sémantique d'un réseau de Petri stochastique (SPN) est un processus stochastique à événements discrets (DESP), nous rappelons brièvement ce qui est un DESP. Un DESP est donnée par un espace S de l'Etat et deux familles de variables aléatoires {Sn} et {Tn} indexés par IN.  
S0 est l'état initial et T0 = 0 est le temps initial. Pour chaque n> 0, Sn ∈ S est l'état après l'apparition du n-ième événement et Tn ∈ IR≥0 est le moment de survenue de cet événement

Un RDPS se compose d’un ensemble de places représentés par des cercles, d’un ensemble de transitions représentés par des traits et d’un ensemble d’arcs orientés, chaque arc connectant une place à une transition ou une transition à une place.

Les places servent à représenter les états du système : un état est représenté par une distribution de marque (on dit aussi jetons) entre les places. À tout instant une place peut contenir zéro, une ou plusieurs marques représentées par des points Les transitions et les arcs décrivent les possibilités d’évolution entre états.

Ces évolutions se traduisent par le franchissement de transition : une transition est dite validée que si chacune des places qui lui sont connectées en entrée (places d’entrée), contient au moins une marque. Une transition ne peut être franchie que si elle est validée. Il existe 3types de transitions: transition immédiate (le temps de franchissement est zéro) cette transition est plus priorité par rapport aux autre transitions et elle peut avoir une valeur de priorité pour les transitions de même types et un poids pour calculer ses probabilités de franchissement, transition avec temps fixe le temps de franchissement de cette transition est fixé, et transition exponentielle cette transition a un temps aléatoire qui dépend de la variable λ qui est le taux associé à la transition (paramètre de la loi exponentielle).

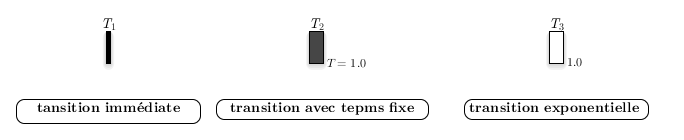


Figure6 : types des transitions dans RdPS

Le franchissement de la transition consiste à retirer une marque de chaque place d’entrée et à déposer une marque dans chacune des places qui lui sont connectées en sortie (places de sortie). En effet de point de vue de la modélisation, les RdPS ont la possibilité de modéliser aisément des comportements très divers ; synchronisation, parallélisme, partage de ressources, et d’autres mécanismes similaires. Cependant cette modélisation souffre d’une limitation. En effet, la modélisation des systèmes réels mène parfois à des RdPS de taille ingérable, ce qui diminue leur lisibilité et rend leur manipulation et même leur analyse difficile (problème d’explosion combinatoire) (Haiouni, 2010).

Pour apporter des solutions à cette limitation et augmenter la puissance de modélisation, des extensions du modèle RdPS dites des RdPS de haut niveau ont été proposées dans lesquelles d’autres aspects ont été pris en compte, tels que ; la modularité, la colorisation, la temporisation, et autres. Dans ce qui suit nous examinons les principales extensions des RdPSC

***6.3 Reseaux de Petri Stochastique coloré***

Comme nous l’avons mentionné plus haut, les réseaux de Petri stochastique colorés (RDSPC) constituent une abréviation des RDP ordinaires. Grâce à l’introduction de la notion « couleur» pour les marques (on dit aussi jetons), ils permettent la représentation des systèmes plus complexes. Les RdPSC sont plus adaptés à modéliser les systèmes paramétrés dont les comportements dépendent de la structure de base du modèle plutôt que des cardinalités des domaines de couleurs. Les idées de base sont les suivantes :

On attache à chaque place ainsi qu’à chaque transition un domaine de couleur. Chaque jeton contenu dans une place est coloré par l’une des couleurs du domaine de la place. Chaque couleur

d’une transition représente une manière différente de franchir la transition.

L’évaluation d‘un arc n’est plus un entier (nombre de jetons à retrancher ou à ajouter) mais une fonction dont l’ensemble de départ est le domaine de la transition adjacente et l’ensemble d’arrivée

L’évaluation d‘un arc n’est plus un entier (nombre de jetons à retrancher ou à ajouter) mais une fonction dont l’ensemble de départ est le domaine de la transition adjacente et l’ensemble d’arrivée est le multi-ensemble (car dans une place peut avoir plusieurs jetons ayant la même valeur) associé au domaine de la place adjacente.

Lors du franchissement d’une transition, on instancie la transition par l’une des couleurs de son domaine, puis on évalue les fonctions qui étiquettent les arcs d’entrée de la transition. Cette évaluation fournit le nombre de jetons de chaque couleur à retrancher à chaque place d’entrée. Si le marquage courant dispose d’assez de jetons on les retire et l’on évalue alors les arcs de sortie de la transition pour déterminer le nombre et la couleur des jetons à déposer dans chaque place de sortie.

Les principales raisons justifient l’appel aux RdPSC :

La possibilité de transporter une information structurée car les places ne contiennent pas que des jetons uniformes.

Ce type de réseaux s’utilise pour éviter le problème de taille importante des RdPSC ordinaires dans le cas où des entités différentes présentent des comportements similaires ce qui condense le modèle.

Evaluer et optimiser les performances du système

**Définition :** Soit RdPSC= (P, T, Pre, Post, C, M0, Λ) un réseau de Petri stochastique coloré où :   
P est un ensemble fini de places   
T est un ensemble fini de transitions  
Pre et Post sont, respectivement, les matrices d’incidence avant et après  
C = (C1,C2,…) l’ensemble des classes de couleurs  
M0 est le marquage initial  
Λ = (λ1, .., λ|T|) où λi est le taux associé à la transition ti (paramètre de la loi exponentielle)

1. ***Conclusion***

Dans une première partie de ce chapitre, nous avons présenté les systèmes de production et particulièrement l’aspect aléatoire dans ces derniers.  
Dans une deuxième partie et suite à une études des différentes approches de modélisation existantes des systèmes de production, notre choix de modélisation s’est porté sur les RdPSC pour différentes raisons citées plus haut. Le modèle généré sera utilisé comme support pour la détermination, de manière formelle, les aspects aléatoires (panne, retard,…) pour générer les performances du système. Comme l’étape de modélisation à base de RdPSC n’est pas évidente pour les utilisateurs, nous avons développé un modèle formel permettant à l’utilisateur d'un système de production de décrire et de spécifier les informations relatives à son atelier. Le deuxième chapitre décrira le modèle formel proposé.

**APPROCHE ET SPECIFICATION FORMELLE DES SYSTEMES DE PRODUCTION FLEXIBLE**

1. ***Introduction***

Dans ce chapitre, nous proposons un modèle formel qui permet la spécification d’une large classe des systèmes flexibles de production. Ce modèle permet une description simple et complète de la partie architecturale ainsi que de la partie procédurale du système étudié. La partie architecturale décrit les composants physiques de l’atelier (machines, moyens de transport, etc.) ainsi que leurs capacités fonctionnelles. La partie procédurale prend en compte les différentes gammes de fabrication.

1. **Présentation de l’approche**

Nous proposons une approche de modélisation qui permet d’évaluer les performances de notre système de production flexibles L’approche proposée inclut trois étapes : l’étape de spécification,  
l’étape de modélisation et transformation et en fin l’étape d’évaluation des performances :

Système

Modélisation

Analyse des résultats

Modèle

Analyse

Mesures de performances perperformance

Dans l'étape de spécification du système de production flexible on décrit en premier lieu l’aspect architectural de l’atelier de production flexible qui décrit les caractéristiques fonctionnelles de différents composants physiques et en deuxième lieu l’aspect procédural qui représente les différentes gammes de fabrication permettant la réalisation des produits finis ou semi fini .

Puis en passe à l’étape de modélisation et transformation qui présente notre modélisation  
avec le réseau de Petri stochastique coloré d’un système de production flexible en particulier pour une machine puis la transformation en un modèle globale en respectant le modèle formel générer et on termine cette étapes par une simulation du RdPSC.

Et à la fin l’étape d’évaluation de notre modèle ..

1. ***Fonction de production***

Les entreprises industrielles sont des systèmes complexes, décomposables en plusieurs fonctions essentielles, toutes interdépendantes. Parmi toutes ces fonctions, la fonction production occupe, du moins dans les entreprises de transformation, une place essentielle. La fonction de production, que l'on pourrait également désigner sous le nom de gestion des opérations, concerne l'agencement et la conduite des flux physiques entre poste de transformation d'inputs en outputs de façon à atteindre des objectifs mesurables exprimés en terme de quantité, qualité, délais et coûts.

Cette définition englobe la fabrication proprement dite des produits et les activités associées d'approvisionnement en matières premières et composantes, de gestion des stocks, de contrôle de la qualité des produits.

Comme c’est présenté dans le chapitre précédent, nous nous sommes intéressées à la fonction de production au sein des ateliers de type « job shop hybride », en particulier, dans le but d’évaluer les performances du système. Pour ça notre choix de résolution est basé sur une modélisation utilisant les RdPSC. Vu que l’étape de modélisation à base de RdPSC n’est pas compréhensible et connue par l’utilisateur (qui est supposé ne pas être spécialiste en RdPSC), nous avons défini dans ce chapitre un modèle formel permettant de spécifier l’atelier flexible à base d'éléments maîtrises par l'utilisateur des ateliers de production. En effet, à travers ce modèle l’utilisateur à la possibilité de décrire les différents composants physiques formant l’atelier (machines, moyens de transport et zones de stockage) et de spécifier les différentes gammes de fabrication permettant la satisfaction des commandes. L'outil d'aide que nous voulons offrir aux utilisateurs des systèmes de production possède deux entrées et une sortie :

* L’entrée :  
  ⮚Une commande de production que l’atelier reçoit à chaque fois,  
  ⮚La spécification de l’atelier : les informations introduites par l’utilisateur à travers le modèle formel (spécification du niveau procédural et architectural de l’atelier) sont introduites dans le système une seule fois lors du chargement (mise en place) de l’atelier, ces informations seront modifiés que si on ajoutera des nouvelles modifications ou si on chargera un autre atelier, sinon ces informations restent comme des entrées statiques dans le système.
* Sortie :  
   l’évaluation de la performance du système

Dans ce qui suit on va présenter le modèle de spécification formel que nous avons développé.

1. ***Niveaux de spécification des systèmes flexibles de production***

La spécification du problème est un préliminaire indispensable et d'une grande importance. Puisque c'est dans cette étape que l'on doit définir précisément ce que l'on veut mettre en évidence avec la simulation et la modélisation et à quelle précision on attend. On détermine aussi les indicateurs de performance qui vont permettre de vérifier si l'on a atteint les objectifs fixés. Enfin, il faut pouvoir fournir des données au modèle. Celles-ci sont relatives à tous les éléments utilisés dans la simulation, comme par exemple :

* Données sur les produits à fabriquer : gammes de fabrication, loi d'arrivée dans le modèle,
* Données sur les moyens de production : nombre et types de machine, nombre et types de ressource complémentaire de production (outils, palettes, etc.),
* Données sur les systèmes de manutention : nombre et types de convoyeur ou de chariot, capacité en nombre d'articles, vitesse/temps d’exécution de déplacement, etc.,
* Données sur les stocks et les magasins : types et capacités, règles d'entrée et de sortie, temps minimal et maximal du séjour des pièces dans le stock (s’il doit être soumis à des contraintes temporelles), etc.,

Aussi bien d’autres données qui sont en relation avec le temps, qu’on va les détailler par la suite. Pour la spécification des systèmes flexibles de production, nous distinguons deux niveaux :

* Un niveau architectural : décrit les caractéristiques fonctionnelles des différents composants physiques (machines, moyens de transport, zones de stockage) et leurs interaction au sein du système.
* Un niveau procédural : décrit les aspects de production. En effet, il représente les différentes gammes de fabrication permettant la réalisation des produits finis. Chaque gamme est décrite par l’ensemble des opérations qui la forment.  
  1. ***Le niveau architectural***

Le niveau architectural de l’atelier flexible de production décrit les caractéristiques fonctionnelles des différents composants physiques, il est constitué du :

* **l'unité de charge :**

L'unité de charge (LU) représente le processus par où les matières premières sont chargés, les paramètres de Lu sont:

* ensemble de matériaux chargés: le type de matières premières du système de production
* la taille des tampons
* les temps de chargement: la distribution du temps de charge pour chaque type de matériau.
* **Machine**:  
    
  Machine est ensemble fini, noté M, de m machines différentes qui permet d’exécuter un ensemble d’opérations.

Les informations associées aux machines sont :

* Toute machine est capable d’exécuter un ou plusieurs opérations,
* Les machines sont indépendantes les unes des autres,
* Les tailles des buffers d’entrées et de sorties,
* Le temps d’exécution
* **Moyens de transport**

Le moyens de transport est un ensemble fini, noté T qui permet de transporter les pièces vers les machines.

Les informations associées aux moyens de transport sont :

* Un moyen de transport est caractérisé par sa mobilité et par sa capacité.
* le niveau de la liberté de l'unité, en précisant si la trajectoire est fixe ou susceptibles de changer en fonction des besoins
* le délai de livraison en fonction des matériaux à être livré et l'emplacement (initiale ou de destination).  
  1. ***Le niveau procédural***

Le niveau procédural représente les différentes gammes de fabrication permettant la réalisation des produits finis.

* Toute gamme de fabrication est une séquence ordonnée d’opérations,
* Chaque produit est fabriqué selon une ou plusieurs gammes de fabrication,
* Chaque opération est exécutée par une ou plusieurs machines,
* Chaque opération a un ensemble de produits entrants et un ensemble de produits sortants,
* Le temps d’exécution d’une opération diffère d’une machine à une autre.

Et, les contraintes intrinsèques au système de production flexible sont les suivantes :

* La préemption n'est pas permise (l’exécution d'une opération en cours sur une machine ne peut être interrompue),
* Une machine ne peut effectuer qu'une seule opération à la fois,
* Deux opérations d'une même tâche ne peuvent être exécutées simultanément (contraintes de précédence),
* Chaque tâche possède un ordre spécifique,
* Les tâches sont indépendantes les unes des autres.

1. ***Un m****o****dèle formel pour la spécification des systèmes flexibles de***  
   ***production***

Dans cette section, nous introduisons un modèle formel permettant la spécification d’une classe de systèmes flexibles de production. Dans ce modèle, la spécification formelle d’un tel système est composée de deux niveaux : le niveau architectural et le niveau procédural.

* 1. ***Modèle formel du niveau architectural***

Formellement, le niveau architectural de la spécification est un triplé <LU, M, T> tel que :

LU : l’unité de charge représente le processus où les matières premières sont chargées

M : un ensemble fini de machines  
T : est un ensemble fini de moyens de transport.

Nous associons, à ces composants, des informations relatives à leurs capacités Opérationnelles. Ces derniers représentent les propriétés caractéristiques du niveau architectural de l’atelier flexible. Ces informations décrivent les opérations exécutées dans l’atelier, les opérations exécutées par chaque machine, le temps nécessaire pour l’exécution de ses opérations, le temps des pannes des machines, le temps de réparation des machines, les capacités des zones de stockage, les pièces (opérations) stockés par chaque zone de stockage, la mobilité de chaque moyen de transport (quel composant est atteignable à partir de quel autre), et le temps nécessaire pour transporter les pièces d’une zone à une autre zone.

* + 1. **Modèle formel d’une machine**

On commence par la représentation d’un modèle formel d’une machine, ces informations sont décrites à travers un ensemble d’applications, à savoir : « M », «  MP\_produit », «  O », «  T-M », « T-P », « T-R », Capacité définies comme suit :  
  
Les unités de ressources présentée dans l’atelier est définit par l’application « M » avec :  
M : machines 🡪 IN

L’application « Mp\_produit » représente la (les)matière(s) première(s) nécessaire pour la production de chaque produit:  
Mp\_produit : matière\_premier\_produit 🡪 IN

L’ensemble des opérations exécutées dans l’atelier est défini par l’application « O » avec:  
O : Atelier →2op

Les applications « T-M » « T-P » et « T-R » fournit, pour la machine, l’ensemble des opérations qu’elle exécute et le temps d’exécution nécessaire pour chaque opération ainsi le temps de panne et de réparation de cette machine. Notons que le temps d’exécution, de panne et de réparation sont des temps aléatoire n’est pas un temps statique, il varie d’une Machine à une autre, de plus il diffère de type d’opération. En effet, si une machines (M1) exécute deux opérations (O1) et (O2), le temps d’exécution d’O1 n’est pas forcement égal au temps nécessaire à O2. Par conséquent, le temps d’exécution d’une opération est en fonction de la machine et de l’opération et de types des matières premiers.  
T-M : temps\_d’exécution 🡪T-M: M → tel que T-M(m) ⊆ (O \* T), (∀m∈ M)  
T-P : temps\_de\_ panne 🡪T-P: M → tel que T-P (m) ⊆ (O \* T), (∀m∈ M)  
T-R : temps\_de\_réparation 🡪T-R: M → tel que T-R (m) ⊆ (O \* T), (∀m∈ M)

Avec T c’est le temps aléatoire d’exécution T=λ(m)

L’application « Capacité » représente la capacité du stock (d’entrée ou/et de sortie) dédié à la machine:  
Capacité: M🡪2(SM\*IN\*) tel que SM=In-Stock ∪ut-Stock

* + 1. **modèle formel global des machines**

Notre but est de minimiser le modèle pour qu’il soit lisible est facile à lire pour cela nous allons fusionner les machines dans un seul modèle globales les opérations « M », « O », « T-M », « T-p », « T-R » et « C » restent inchangeables

L’application « Mp-produit » sera modifier par l’application « Mp-produit-machine » qui associe à chaque matière premier la ou les machines qui peut (peuvent) l’exécuter  
Mp-produit-machine : M🡪2(matiére\_premier)

* + 1. **modèle formel des unités de transport**

L’ensemble des unités de transport présentées dans l’atelier est définie par les applications suivantes :

L’application « T » définit l’ensemble des unités de transport dans le SdP est définit comme suit :  
T : unité-transport 🡪IN\*

La propriété caractéristique d’un stock est sa capacité de stockage. Cette propriété est définie par l’application:  
Capacité-transport: C-T → IN\*

Chaque moyen de transport est caractérisé par deux propriétés : Sa mobilité décrivant sa capacité à transporter les produits d’un composant à un autre, le temps nécessaire pour les transporter et sa capacité. Ces propriétés sont introduites par les applications suivantes :  
Mobilité : T →2(pos\*pos\*Q+) où Pos = M∪T Et T\_Capacité : T →IN\*

* 1. ***Modèle formel du niveau procédural***

Formellement, le niveau procédural de la spécification est un couple <G, P > tel que :  
•G est un ensemble fini de processus de fabrication  
•P est un ensemble fini de produits finis et semi-finis.  
  
Le processus de fabrication est une liste qui contient toutes les opérations à exécuter dans le  
but d’usiner une pièce ou bien de décrire la réalisation d’un produit.   
Chaque opération à exécuter peut être une opération de transformation ou de stockage. Pour  
chaque produit sortant d'une opération donnée, le processus de fabrication définit la prochaine  
opération à réaliser sur ce produit.Formellement, chaque procédé de fabrication gi est défini par un système de transition <TNi ,  
F, I, σi > où:  
\* TNi est un ensemble fini d'états (nœuds), avec <NTR ,NA ,ND , S ⊆est une partition  
de TNi.  
<NTR ,NA ,ND , S > représente respectivement les nœuds de transformation, de test,  
d’assemblage, de désassemblage et de stockage.

\*F représente le buffer de sortie qui stock les produits finis

\*I représente le buffer d’entrée qui stocke les matières premières permettant la  
fabrication des produits.

\*σ est une fonction de transition :

σi: TNi\*bag(Pi)🡪 TNi avec Pi ⊆P

Pour chaque processus de fabrication l’application permet d’associer pour chaque nœud  
de traitement l’opération correspondante:  
 NOi :TNi\S🡪O

L’application « S-T-Op » fournit, pour chaque zone de stockage le temps de stockage qui un temps aléatoire qui est associe à chaque opération de stockage ; Se définit comme suit :  
 S-T-Op : B→2 g\*λ(m)) avec λ fonction de temps aléatoire

Un produit fini est réalisé suivant une ou plusieurs gammes de fabrication. Cette relation est définie par l’application:  
Prod : G → P

L’application « Mp\_produit » représente la (les)matière(s) première(s) nécessaire pour la production de chaque produit:  
Mp\_produit : G → 2MP

Nous associons à chaque opération un ensemble de produits entrants et un ensemble de produits sortants. L’application « IOProd » fournit ces informations :

IOProd: O →2 (2P\*2P)

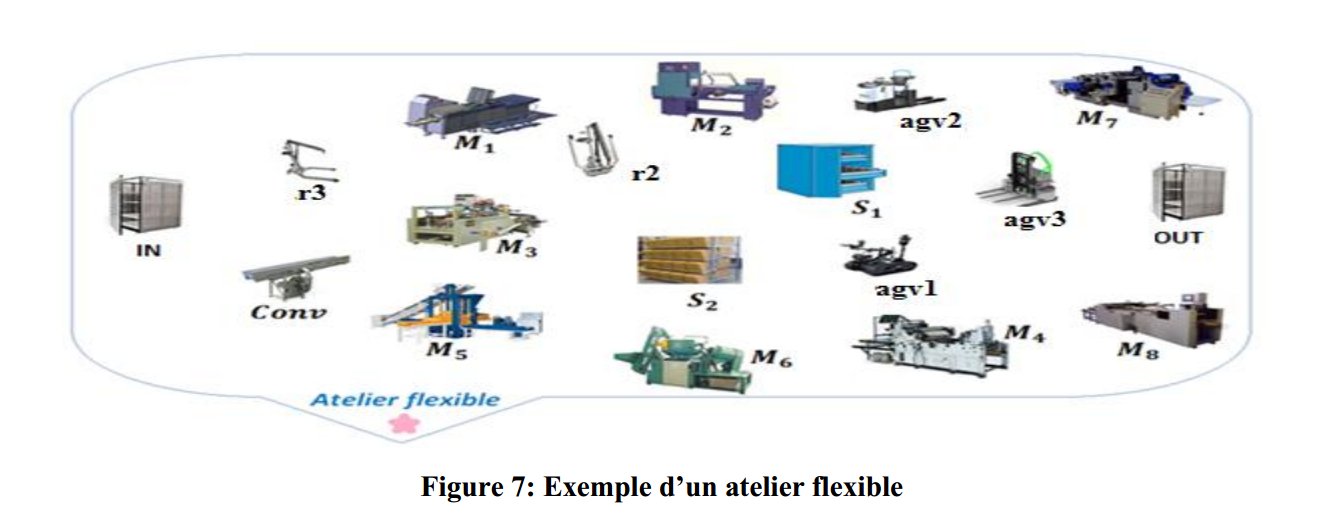
* 1. ***Modèle formel d’initialisation***

Dans notre atelier, nous avons supposé qu’à l’instant zéro, début de production, les moyens de production et les matières premières ne sont pas toutes disponibles, pour cela un modèle formelle permettant de décrire la disponibilité de ces derniers en fonction du temps qui pourra nous aider par la suite lors de l'ordonnancement des opérations.  
L’application « Dispo-M » présente à quel instant les différentes machines sont disponibles:  
Dispo-M : M → Q+   
L’application « Dispo-M\_T » présente à quel instant les matières premiers sont disponibles est quelle machine peux les exécuter.  
Dispo-M\_T : M\_T 🡪 Q+  
L’application « Dispo-T » présente à quel instant les moyens de transport sont disponibles:  
Dispo-T : T → Q+   
L’application « prêt-G » présente les différentes gammes dans le système de production qui sont prêtes à être exécutées :  
Prêt-G : G → Q+   
Cette dernière application est en relation avec l’application « Dispo-MP » qui présente la disponibilité des matières premières sont disponibles :  
Dispo-MP : MP → Q+

1. **Exemple de spécification formel pour le système de la production**

Exemple : Atelier de production

Dans la section précédente nous avons spécifié en détails les deux niveaux de notre atelier. En effet, l’étape de spécification permet de représenter l’atelier formellement dans le but de s’approcher de l’utilisateur. À travers la spécification formelle générée l’utilisateur peut décrire les différentes informations nécessaires au niveau architectural et au niveau procédural au moyen d'éléments qui les maîtrise. Pour la spécification du niveau architecturale du système, l’utilisateur décrit les différentes machines, zones de stockage et moyens de transport, en précisant pour chaque composant ses propres caractéristiques. Et pour la spécification du niveau procédural du système, l’utilisateur décrit les différentes gammes de fabrication permettant la réalisation des produits finis. En particulier, il précise les différentes opérations formant chaque gamme, ainsi que les ensembles des produits entrant et sortant associés à chaque opération, et dans le cas des opérations de stockages il indique le temps de stockage minimal qui doit être respecté pour assurer la qualité du produit. Pour illustrer la spécification d’un atelier flexible, considérons l’atelier illustré par la figure 7. Cet atelier est formé de huit stations de travail, M1, M2, M3, M4, M5, M6, M7, M8 et de quatre zones de stockage IN, S1, OUT et S2 de capacité maximale égale respectivement à cent, dix, deux cent et vingt. M1 est muni d’un stock de sortie de capacité égale à quatre, alors que M4 possède un stock d’entrée capable de contenir au maximum sept pièces. Le système de transport de cet atelier est formé de deux robots r1 et r2, trois AGV agv1, agv2 et agv3 et d’un convoyeur conv. r1 transporte les produits depuis le point d’entrée IN vers M1 et M3. conv transporte les produits depuis le point d’entrée IN vers M3 et M5. r2 alimente les machines M1 et M3 depuis M2, transporte les pièces vers M2 depuis S1 et vers S2 depuis M1. agv1 charge la zone de stockage S1 depuis la machine M5, transporte les pièces vers S2 depuis M3 ou depuis M5, charge la machine M6 depuis S2, transporte les pièces depuis M6 vers M4 et depuis M5 vers M6. agv2 alimente la machine M7 depuis la zone de stockage S2 ou depuis M2 et transporte les pièces depuis S1 vers M2. agv3 charge les machines M8 et M4 depuis S2. agv3 alimente le buffer de sortie OUT depuis M4 ou M7. agv1 alimente le buffer de sortie OUT depuis M8 ou M6. agv2 transporte les pièces depuis M2 vers OUT. Cet atelier permet la fabrication de six produits différents p1, p2, p3, p4, p5 et p6. P1 est fabriqué suivant les gammes G1 et G2. G3 permet la réalisation de p2. G4 permet la réalisation de p3. G5 permet la réalisation de p4. G6 permet la réalisation de p5. G7 permet la réalisation de p6.



Les sept gammes sont décrites par le tableau ci-dessous :

g1 = op1.op5,   
g2 = op3.S2 [3 u.t].op6,  
g3 = op2.op4.S1[6 u.t].op7,  
g4 = op3.S1[5u.t].op9,  
g5 = op1.S2[6u.t].op8,  
g6 = op11. Op9.op10  
g7 = op12. Op13.op14. op15

L’ensemble des opérations exécutées dans notre atelier sont spécifiées dans le tableau ci-dessous :

Atelier Operation  
O *(* *atelier)* {op1,op2,op3,op4,op5,op6,op7,op9,S2,S3,op8,op10, op11,op12, op13, op14, op15}

Le modèle formel du niveau architectural de la spécification de cet atelier est un triplet <M, B, H> tel que :

M = {M1, M2, M3, M4, M5, M6, M7, M8}

L’ensemble des opérations exécutées par les différentes machines sont spécifiées dans le tableau ci-dessous :

T-O

|  |  |
| --- | --- |
| M | M(op) |
| M1 | O1,O2,O11 |
| M2 | O7,O9,O5 |
| M3 | O1,O3 |
| M4 | O15,O14,O8 |
| M5 | O4,O12 |
| M6 | O6,O13 |
| M7 | O8,O10 |
| M8 | O6,O15 |

Le tableau suivant indique les stocks d’entrée ou /et stocks de sortie et les capacités maximales des machine:

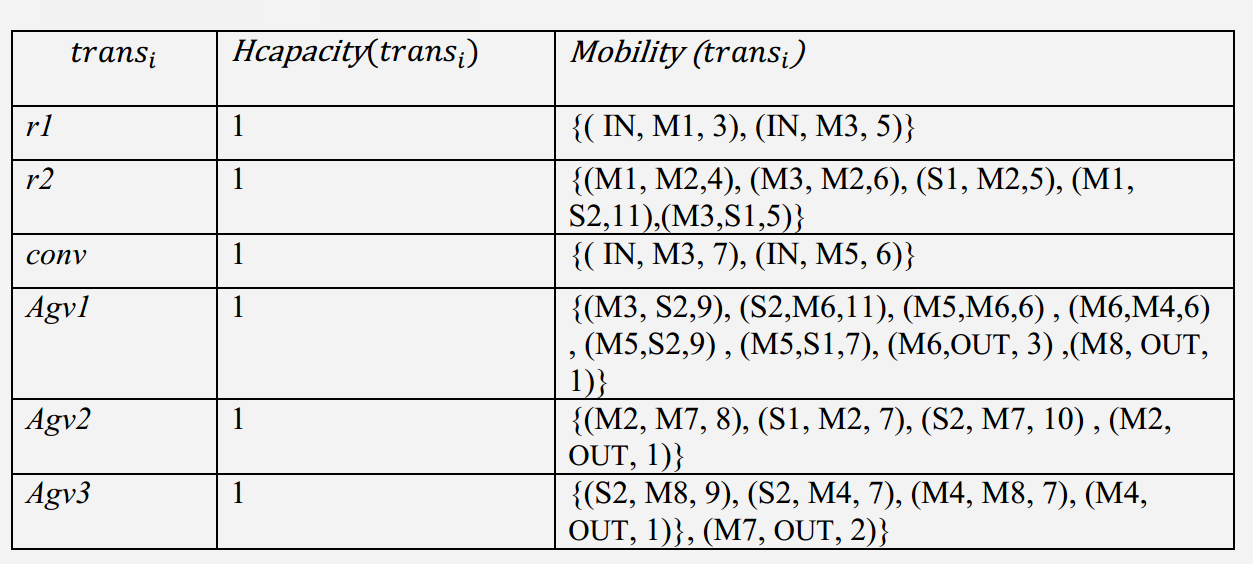
M1 {(stock\_entree, 5)}, {(stock\_sortie, 5)}  
M2 {(stock\_entree, 6)}, {(stock\_ sortie, 6)}  
M3 {(stock\_entree, 7)}, {(stock\_ sortie, 7)}  
M4 {(stock\_entree, 1)}, {(stock\_ sortie, 4)}  
M5 {(stock\_entree, 5)}, {(stock\_ sortie, 8)}  
M6 {(stock\_entree, 6)}, {(stock\_ sortie, 6)}  
M7 {(stock\_entree, 6)}, {(stock\_ sortie, 6)}  
M8 {(stock\_entree, 1)}, {(stock\_ sortie, 2)}

**B = {S1, S2, S3, IN, OUT}**

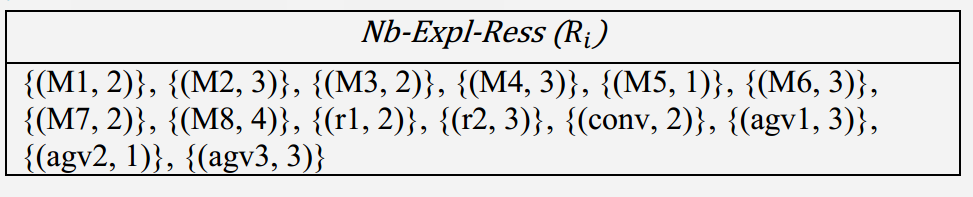
Capacity (Si) IN =100 = 10 =20 OUT =200

**H = {r1, r2, agv1, agv2, agv3, conv}**

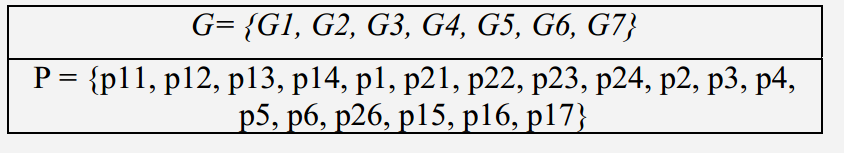
HCapacity et Mobility sont décrites dans le tableau suivant :

****

Le tableau suivant décrit le nombre d'exemplaires qui existe pour chaque ressource :

****

Le modèle formel du niveau procédural de la spécification de cet atelier est un couple <G, P > tel que :



Les gammes de fabrication s’écrivent comme suit :

G1 : **IN** --P11-->**N1** --P11-->**N2**--P1-->**Out**

G2 : **IN** --P13-->**N1** --P14-->**S2**-->**N2**--P1-->**Out**

G3 :

G4 :

G5 :

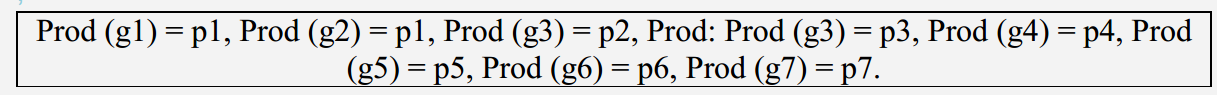
G6 :

G7 :

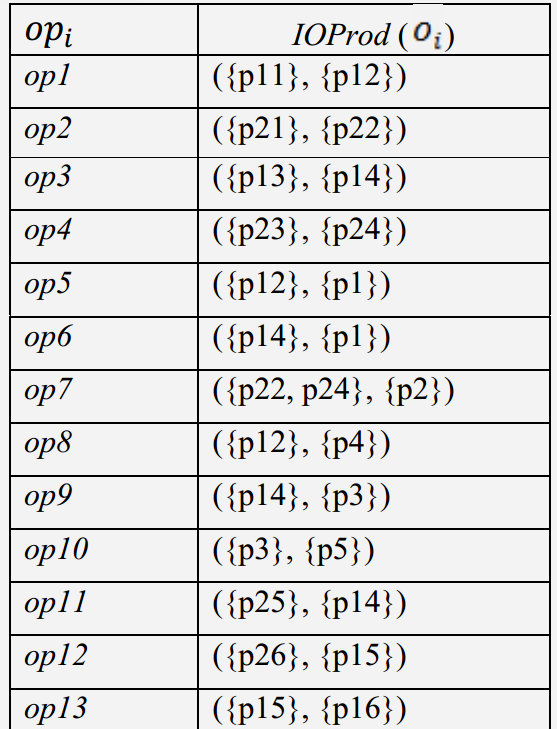
Formellement G1 est un < TN1, F, I, **σi** > où :  
∪ NT ∪ NA ∪ ND ∪ B tels que : NTR= {n1, n2}, NA=ND =NT = ∅ , S=B  
  
−FU  
−  
−La fonction NO1 est définie comme suit : NO1(n1)=op1,NO1(n2) =op5.

PS : idem pour les autres gammes

« Prod » est décrit dans le tableau suivant :



La description des produits entrants et sortants, associée à chaque opération, est donnée dans le tableau suivant :



1. **Conclusion**

Dans ce chapitre, nous avons introduit un nouveau modèle formel permettant la spécification des ateliers flexibles e. Cette spécification est faite en deux niveaux : le niveau architectural et le niveau procédural.  
Le niveau architectural permet de décrire les différents composants physiques de l'atelier étudié ainsi que les différentes caractéristiques qui leur sont associées.   
Le niveau procédural décrit les différentes gammes de fabrication permettant la réalisation des différents produits finis.   
Dans le chapitre suivant, nous introduisons un modèle réseau de Petri stochastique coloré temporisé générique modélisant le comportement des ateliers flexibles à partir d’une spécification conforme à notre modèle formel.

**Modélisation du système de production flexible par le réseau de Petri stochastique**

1. **Introduction**

Ce chapitre introduit le nouveau modèle réseau de Petri stochastique coloré que nous avons développé pour représenter le comportement générique des ateliers flexibles. Les spécifiés du modèle sont déduites à partir d’une spécification conforme à notre modèle formel. La dernière partie de ce chapitre est consacrée à la présentation des premiers résultats de simulations du modèle pour évaluer les performances des systèmes de production.

1. ***Choix de la plateforme***

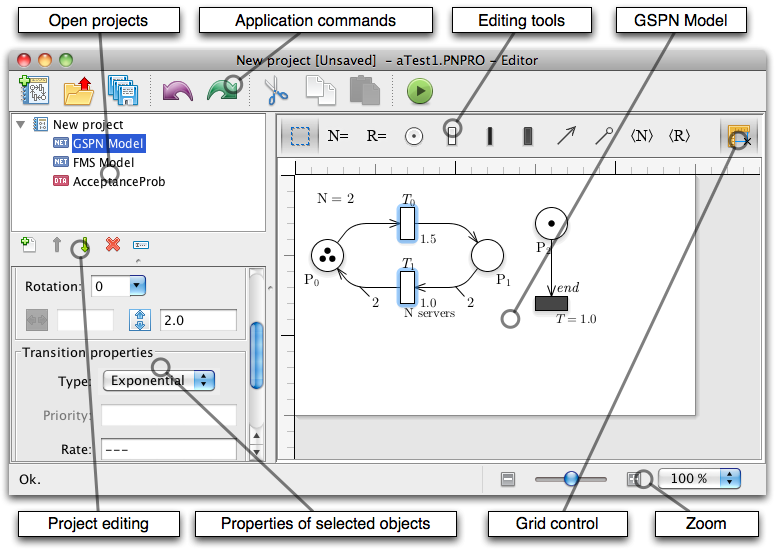
Pour pouvoir modéliser et évaluer les performances des systèmes de production, il est important de voir son comportement. L’outil que nous allons utiliser pour la modélisation est la New GreatSPN 1.5, un logiciel très intéressant pour l’évaluation des performances des réseaux de Petri stochastiques Généralisés.

New GreatSPN est un outil qui gère les RdPSG. Il a été développé à Italie. Sa principale fonction est d’éditer, simuler et analyser les réseaux de Pétri stochastique. Il se compose de plusieurs modules offrant ainsi plusieurs fonctionnalités. Il permet de modéliser graphiquement un RdPS à partir de l’interface utilisateur. Le modèle comprend la structure statique du RdPS (place, transition, arc) et la partie dynamique (les jetons valués). L’intégration du temps d’une manière quantitative (durée associée aux transitions) ou d’une manière aléatoire (transition exponentiel) permettra de bénéficier des différentes méthodes d'analyse de performance offertes par l’outil New GreatSPN.

NewGreatSPN est une interface graphique permettant :

* L'édition des RdPSGs,
* L’analyse structurelle des RdPSGs,
* Le calcul du graphe d'accessibilité des RdPSGs,
* Le calcul des distributions des probabilités des jetons dans chaque état du système : au régime permanent et au régime stationnaire.
* L’étude des propriétés comportementales (vivacité, bornitude, atteignabilité, blocage).
* La simulation de la dynamicité du modèle.
* La visualisation des résultats, …

L’environnement GreatSPN est représenté par la figure 10.

****Figure 10 : L’interface de l’outil New GreatSpn

1. ***Modélisation du système de production***

L’objectif de notre travail est d’évaluer les performances de notre système de production flexible avec l’analyse des séquences d’ordonnancement qui dépend de l’aspect aléatoire (panne/réparation), Plusieurs séquences d’ordonnancement peuvent satisfaire la réalisation des produits commandés, mais leur durée de réalisation n’est pas toujours la même, elle dépend de plusieurs aspect (type des produits, les machines, les opérations, etc). Donc, il serait intéressant d’analysé les séquences d’ordonnancement pour voir l’importance des aspects aléatoires dans l’atelier flexible.   
Notre atelier est composé d’un niveau architectural qui inclut les machines, les moyens des transports et la zone de stockage, et un niveau procédural pour les gammes de production   
Dans ce qui suit, nous allons vous présenter la modélisation de notre système de production flexible réalisée avec l’outil de modélisation choisi New GreatSPN.

***3.1 Modélisation du niveau architectural de l’atelier***

Nous rappelons que le niveau architectural décrit les différents composants physique formant l’atelier (machines, moyens de transport, et zone de stockage), nous allons s’intéresse uniquement à la modélisation des machines pour le niveau architectural, la figure 11 nous montre en fait le modèle d’une machine avec la présence de l’aspect aléatoire (panne et réparation)

Ce réseau (présenté dans la figure 11) représente l’aspect aléatoire (panne) qui peut se produire à l’état de repos (état idle) de la machine et dans ce cas les opérations seront suspendu jusqu’à la réparation de la machine. L’aspect panne peut se produire aussi en cour de traitement de l’opération apres avoir eu les informations de la produit a traité mais dans ce cas le produit sera perdu et la machine sera prête à un nouveau pour le prochain traitement après sa réparation. Si l’exécution de l’opération est terminée le produit sera évacué dans le buffer sorti et la machine sera prête pour l’exécution de la prochaine opération

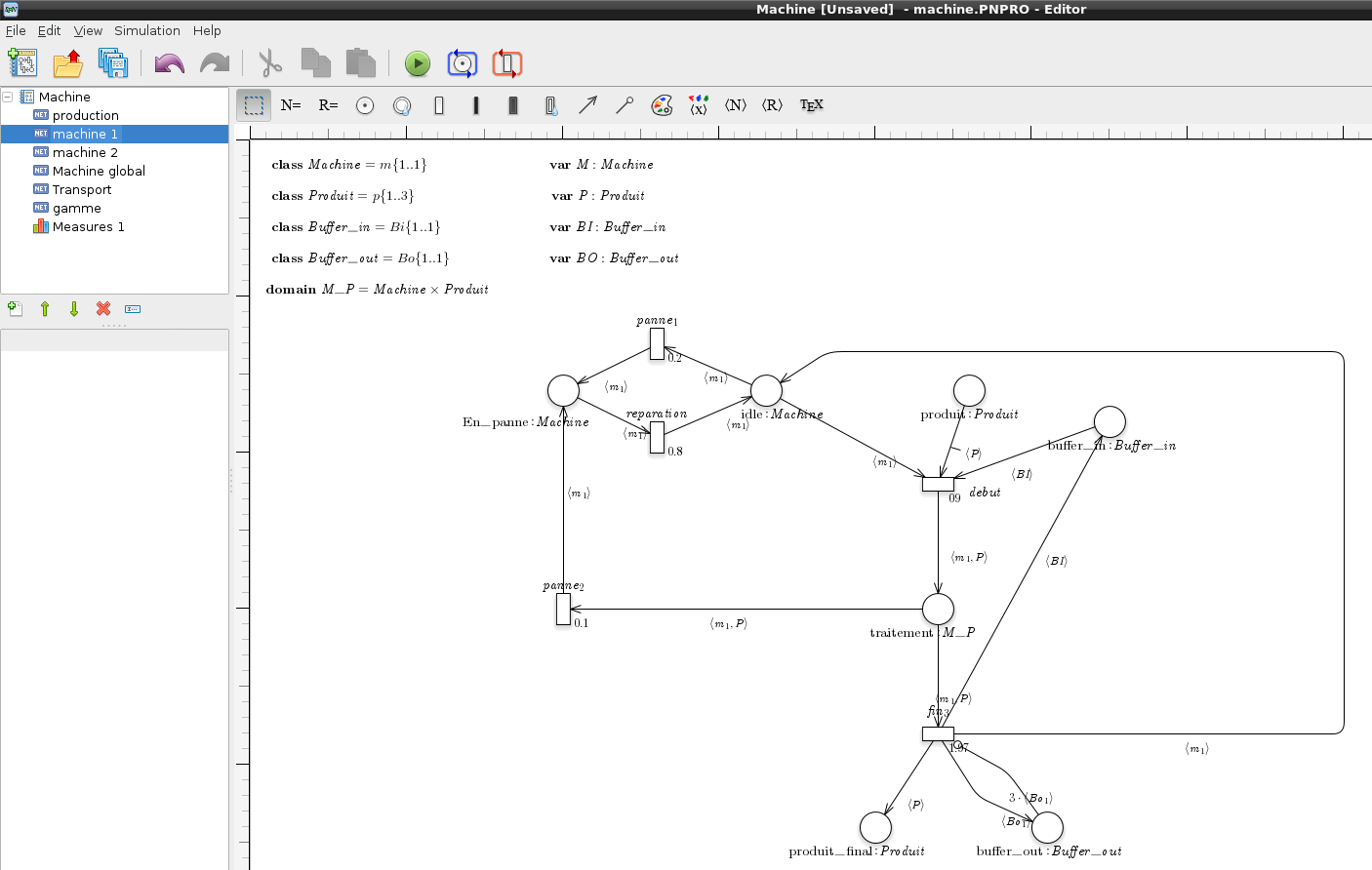


Figure 11 : Modélisation d’une machine

* 1. ***Modélisation du niveau logique de l’atelier***

Le niveau procédural (logique) de l’atelier représente les différentes gammes de fabrication permettant la réalisation des produits finis, Chaque produit est fabriqué selon une ou plusieurs gammes de fabrication.  
La figure 12 montre la modélisation des gammes de fabrications, une gamme est une suite ordonnée d’opérations exécutée par une ou plusieurs machines. Dans ce modèle une gamme est définit par un identificateur (G), un numéro d’opération pour le prochain traitement (Num) ce numéro peut distinguer les opérations pour ne pas confondre entre elles et Opération (O) définit l’opération en cour.  
Exemple : Soit une gamme G1 définit par les opérations suivantes « G1 :O1O3O1O2 »   
La première opération O1 et dans la premier position et son successeur et o3 par contre le deuxième O1 et à la troisième position dans la gamme et son successeur O2   
La place « produit gamme » de ce model définie les gammes associé à chaque produit et le numéro de l’opération de la prochaine exécution.

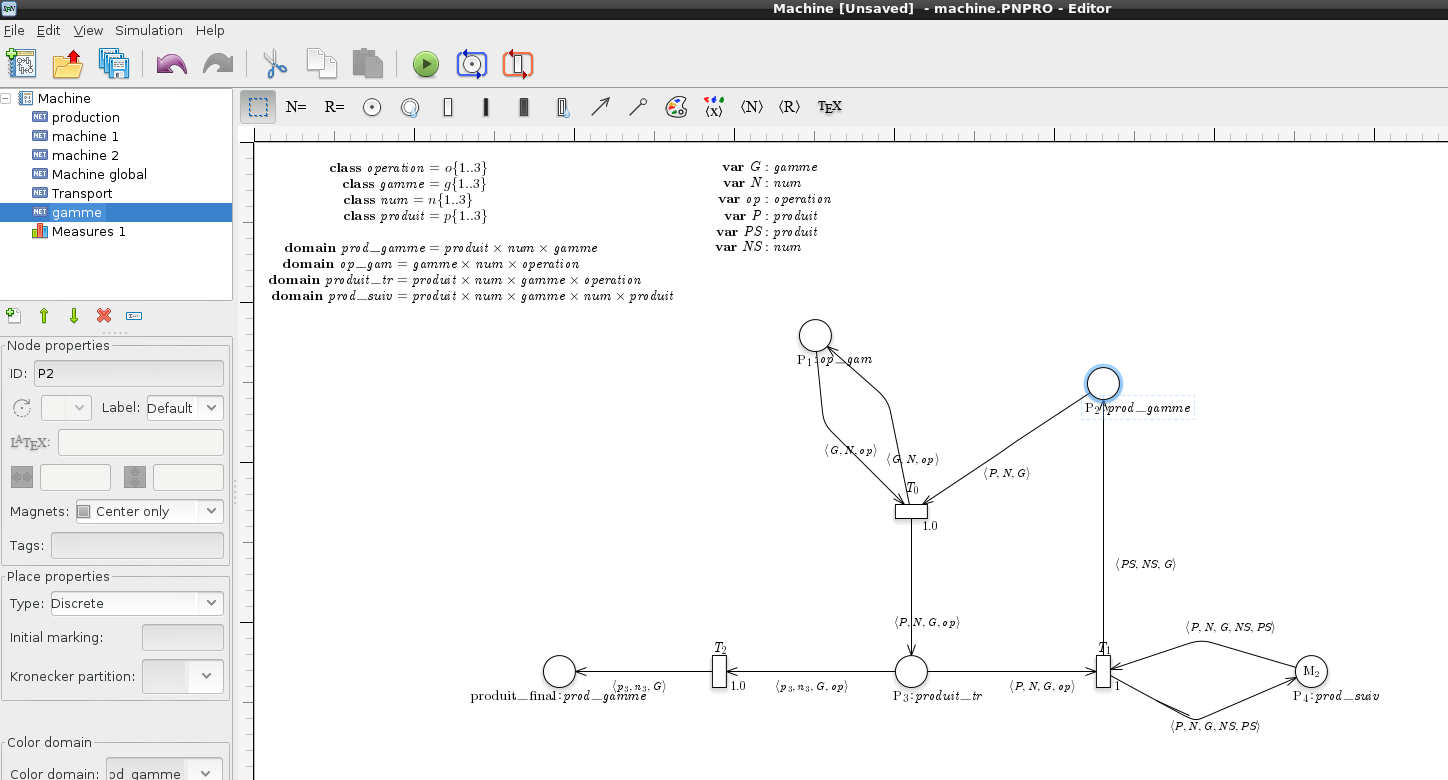


Figure12 : model de gamme

Apres avoir eu les informations du deux places  opérations gamme (type de gamme, opération en cour, sa position) et produit gamme (produit, gamme, le numéro de l’opération pour le prochain traitement) la transition T0 sera franchi si le numéro de la dernier opération est attient alors t2 sera franchi pour évacuer le produit final si non T1 sera franchi pour prépare l’opération suivante à l’aide de la place opérations suivante (produit, numéro de l’opération en cour, type de gamme, numéro de opérations suivantes, produit suivant) qui donne le numéro de la prochaine opération de la gamme.

* 1. **Modèle global du système de production**

Le model global c’est la combinaison du model architectural (model physique) et le model procédurale (model logique), chaque produit dans l’atelier est définie par une gamme et la gamme est représenté par une séquence d’opération.  
  
La figure 13 nous montre en fait les principales étapes d’un système de production (itération générique) satisfaisant les besoins des clients en quantité et en qualité c'est-à-dire les étapes permettant la réalisation des produits finis.

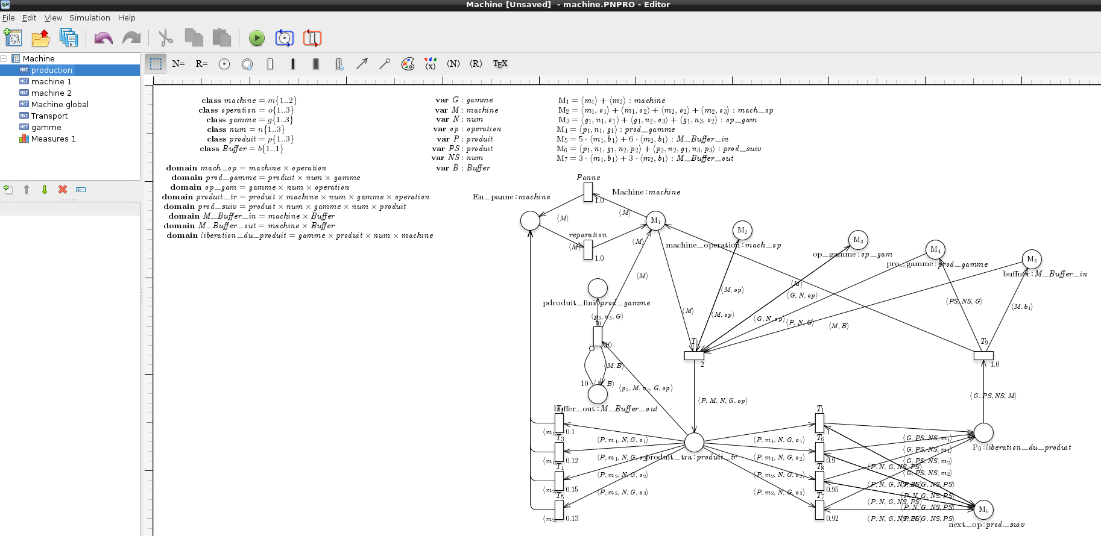


Figure 13: model global du système de production flexible

A partir de l’état idle les machines peuvent tomber en pannes avec un taux μ1 et dans ce cas la machine doit être réparée pour traiter le produit. Pour débuter l’exécution de l’opération il faut avoir les informations du produit, gamme, opération en cour et sa numéro de position dans la gamme et encore il faut avoir une place libre dans le buffer d’entré de la machine. Apres avoir eu ces informations l’opération sera prêt à l’exécution, ou cour d’exécution la machine peut tomber en panne avec un taux μ2, et le produit sera perdu et la machine sera réparé pour un nouveau traitement d’opération le taux de panne μ2 dépend de type de la machine et l’opération en cour. Si l’opération est terminer T1, T6, T7 ou T8 sera franchi pour déterminer l’opération suivante selon le type de l’opération dans ce cas la machine libère le produit du sa buffer de sortie pour l’opération suivante, ce cycle de traitement se répéter jusqu’à atteindre la dernière opération et dans ce cas T0 va se franchi pour évacuer le produit final

1. ***Modèle paramétré***

Pour pouvoir effectuer la génération du modèle RdPS correspondant, nous avons besoin d’un certain nombre d’informations par exemple :  
  
• les machines présentant dans l’atelier  
• Les opérations effectuées par chaque machine.  
• La succession d’opérations d’une gamme et la disponibilité de ces ressources.  
• Les buffers d’entrées et de sorties de chaque machines, etc.

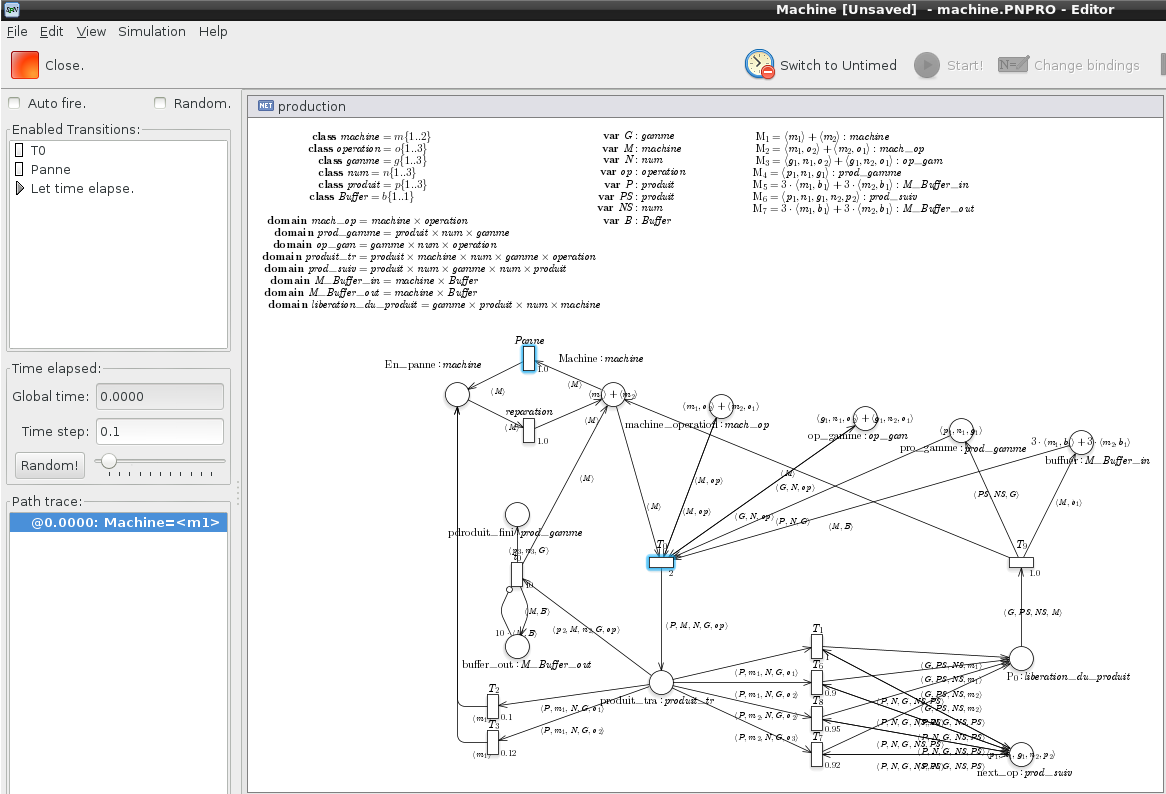
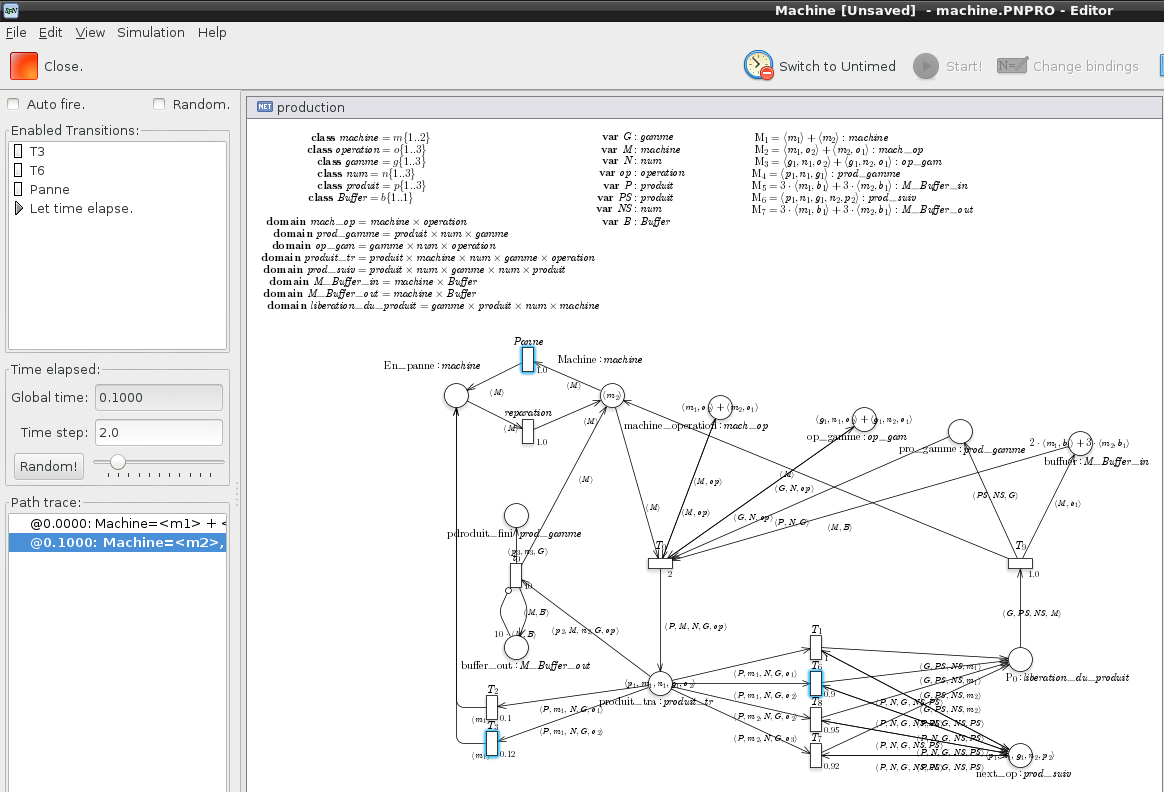
Ces informations sont toutes fournies par l’utilisateur lors de la spécification de l’atelier flexible comme nous les avons vues dans le chapitre précédent. En effet, nous avons donné à l'utilisateur la possibilité de spécifier les deux niveaux procédural et architectural à travers un modèle formel.  
L’utilisateur spécifie les différentes machines, les zones de stockage et les moyens de transport qui forment l’architecture physique du système. En ce qui concerne le niveau procédural du système, il décrit les différentes gammes de fabrication permettant la réalisation des produits finis. Ces informations sont disponibles grâce au marquage initial de certaines places particulières. Le marquage initial de ces places est spécifique à chaque modèle traité, et doit être déterminé à partir des différents éléments du modèle de spécification. Plus précisément, le marquage initial présente l’état initial du système. Le modèle généré est un modèle paramétrique c'est-à-dire il est adapté quelque soit les données fournies en entrées par l’utilisateur (spécification de l’atelier). Pour cela, nous avons paramétré notre modèle généré par l’étude de cas de l’atelier présenté dans le chapitre précédent. Nous présentons, dans ce qui suit, le marquage initial de quelques places pour une atelier avec 2 machines M1 et M2, un produit P1 qui suit la gamme G1 avec G1=O1O3O2 :

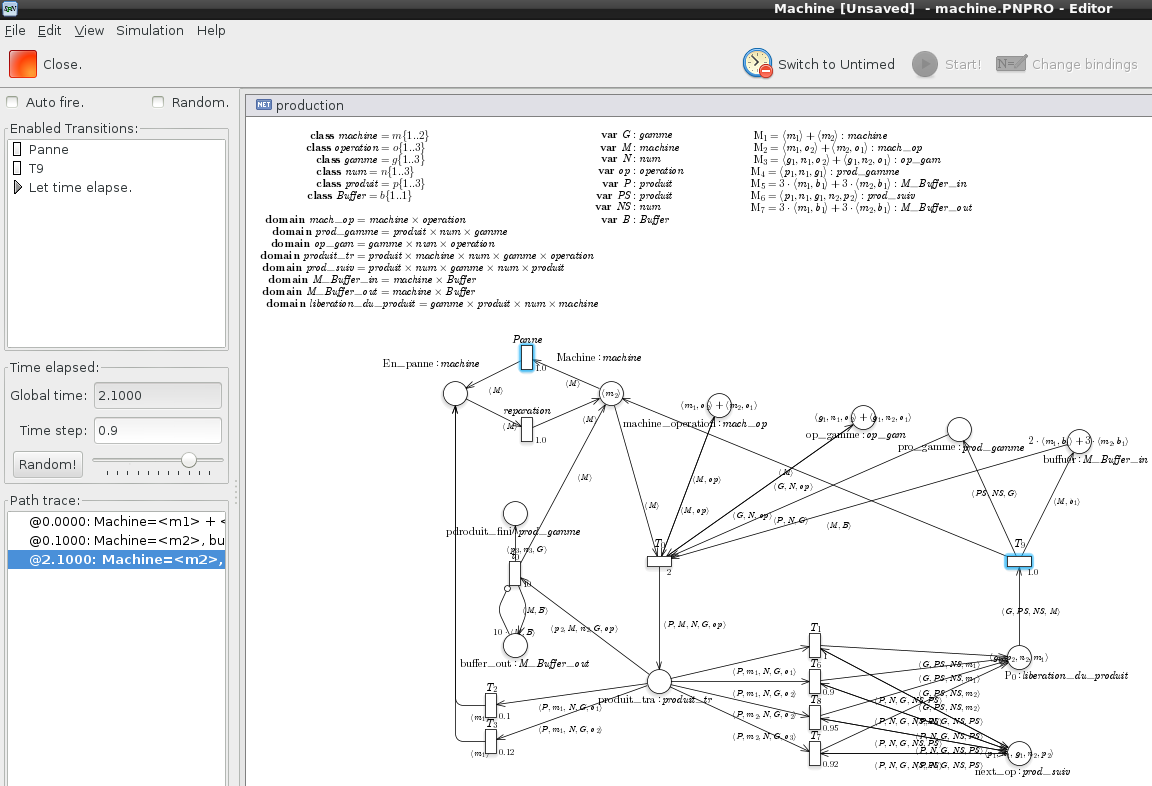
* Le marquage initial de la place machines « Machine » permet de définir les machines de l’atelier   
  M0 (Machine) : (Machine) =<m1>+<m2>
* Le marquage initial de la place machines opérations « mach\_op » permet de définir pour chaque opération la ou les machines qui peut (peuvent) l’exécuter  
  M0 (mach\_op) : (machine\_operation) = <m1,o1>+<m1,o2>+<m2,o2>+<m2,o3>
* Le marquage initial de la place opérations gammes « op\_gamme » permet de définir l’ordonnancement des opérations pour chaque gammes   
  M0 (op\_gamme) : (operation\_gamme) = <g1,n1,o1>+<g1,n2,o3>+<g1,n3,o2>
* Le marquage initial de la place « prod\_gam » qui permet de définir les produits de l’atelier et les gammes qui suivent  
  M0 (prod\_gam) : (produit\_gamme) = <p1,n1,g1>
* Le marquage initial de la place « prod\_suiv » qui permet de définir la prochaine opération de la gamme  
  M0 (prod\_suiv) : (produit\_suivant) = <p1,n1,g1,n2,p2>+<p2,n2,g1,n3,p3>

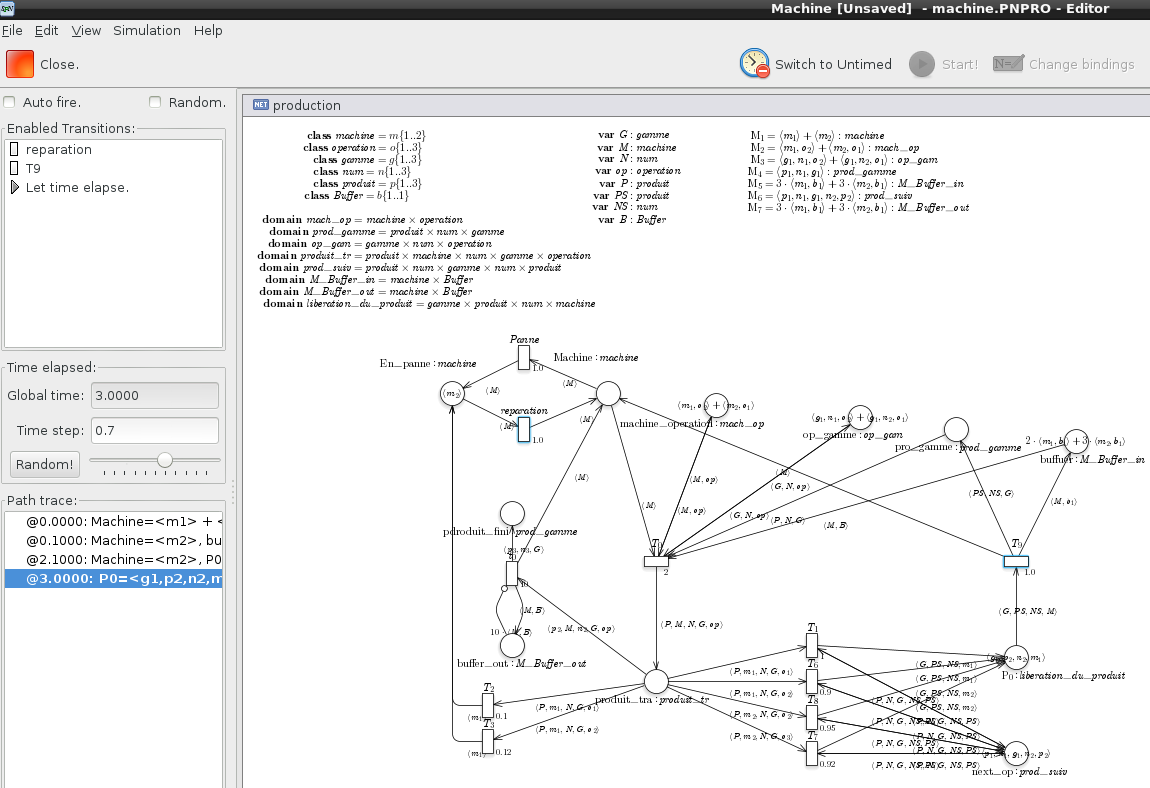
1. ***Simulation du modèle***

Prenons un cas de figure où l’étude de cas de l’atelier présenté dans l'exemple du chapitre précédent reçoit une commande de production pour la fabrication d’un produit P1. Le produit P1 suit la gamme G1 avec G1 :O2O1, il y a deux machine dans l’atelier M1 ne peut exécuter que la première opération et M2 exécute la deuxième opération.

On remarque que la place « produit\_fin », ne contient aucun jeton à l’instant 0 puisque début de production aucun produit n’a été réalisé.

  
Figure14 : Simulation1  
  
Figure15 : Simulation2

  
Figure16 : Simulation3

******Figure17 : Simulation4

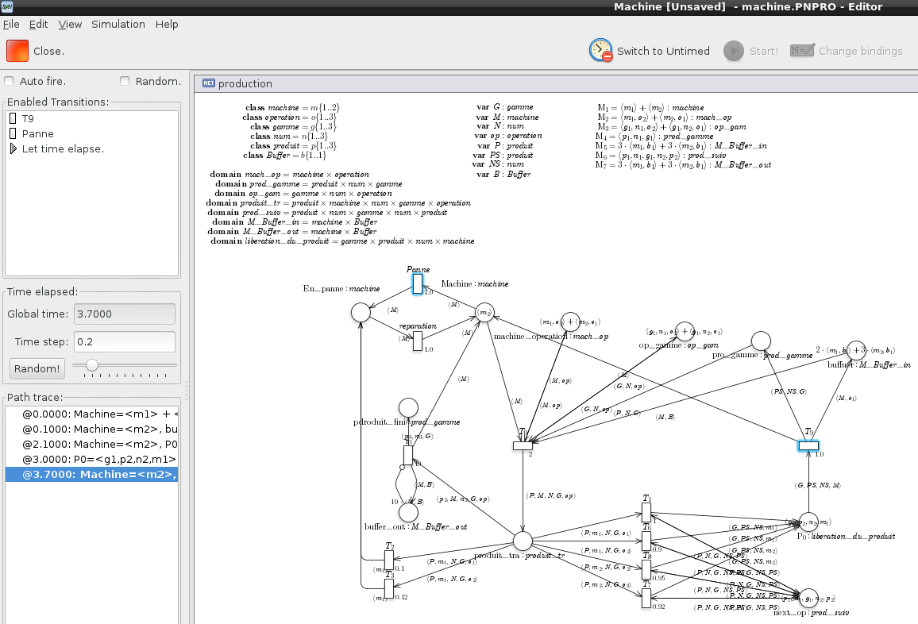
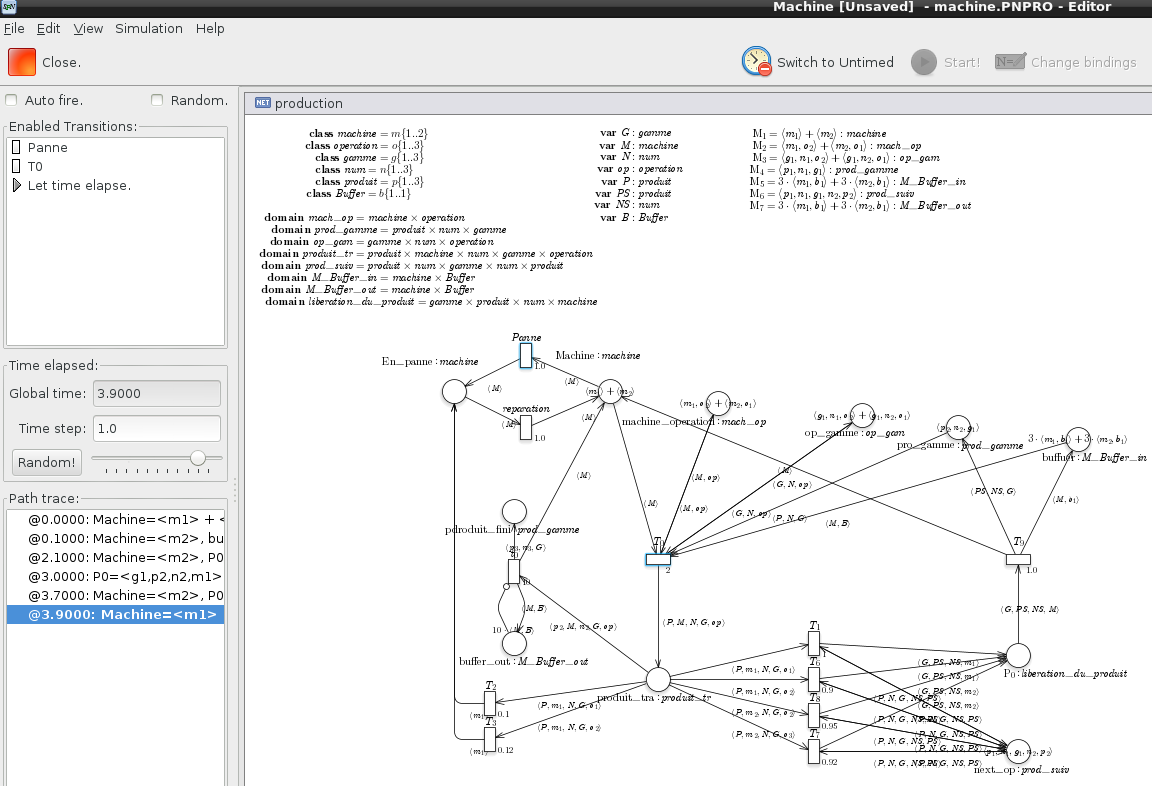
******

Figure18 : Simulation5

******Figure19 : Simulation6

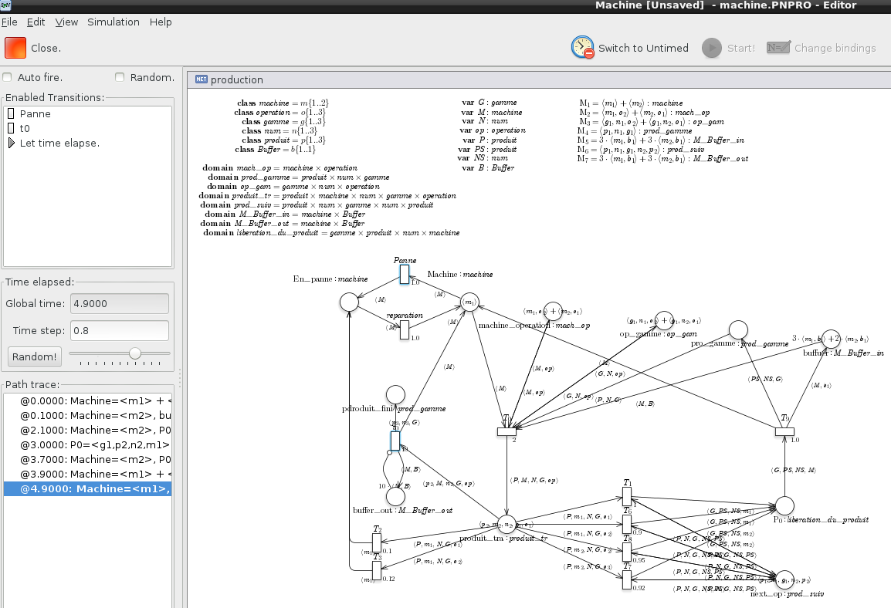
******

Figure20 : Simulation7

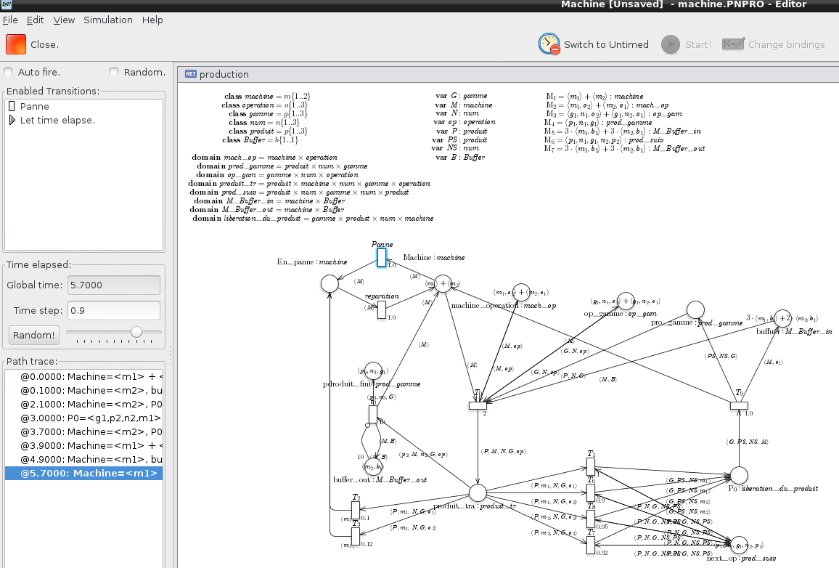
******

Figure12 : Simulation8

Les résultats de simulation prisent lors de ce test nous démontrent que la commande de production sera prête à la disposition du client à l’instant 5.7. Le temps de simulation est un temps aléatoire qui dépend de l’environnement de l’atelier (machine, produit, panne,retard, etc)

1. ***Conclusion***

Ce chapitre présente la génération du modèle Réseaux de Petri stochastique colorés correspondant au système de production flexible à partir de la spécification introduite dans le chapitre précédent. Le modèle proposé représente les différentes étapes permettant la fabrication des produits au sein du système physique. Le modèle généré tient en compte l’aspect aléatoire (panne/réparation) dans le système étudié. Ce qui va faciliter l'exploitation du modèle, par la suite, pour l’évaluation de notre système. Dans le chapitre suivant, nous exploitons notre modèle RdPS pour l’analyse de l’aspect aléatoire et son influence sur le SdP.

***Analyse et Evaluation de performances des systèmes de production flexibles***

***1-Introduction***

Ce chapitre introduit l'approche que nous proposons pour l’analyse de performance des systèmes de productions flexible et l’influence de l’aspect aléatoire sur ces derniers.  
L'approche proposée est basée sur l'exploitation du graphe de marquages associé au modèle réseau de Petri stochastique que nous avons présenté dans le chapitre précédent.  
Comme le modèle généré est un modèle paramétrique, c'est-à-dire il est adapté quel que soit la spécification de l'atelier fournie en entrée par l’utilisateur, l'approche proposée permet de répondre à n'importe quelle commande en fournissant avec une analyse de la performance de l’atelier.

***2-Problématique***

Si l’on considère une évaluation de performance de système de production flexible, alors le réseau de Petri stochastique généralisé est particulièrement bien adapté à la modélisation de l’ordonnancement d’un atelier flexible et l’analyse de ses performances.

En effet, le modèle que nous proposons représente un ordonnancement flexible : disponibilité de plusieurs machines capables de réaliser les différentes opérations avec la présence de l’aspect aléatoire dans le système (Panne/réparation).

Aussi pour réaliser une étude d’évaluation des performances, Il est nécessaire de posséder une représentation du système à étudier, c'est-à-dire un modèle. L’extraction des paramètres de performances se fait alors soit par simulation, soit par analyse. Dans notre travail, nous ne nous intéressons qu’aux méthodes analytiques, qui sont des outils d’évaluation des performances permettant une simplification du modèle et proposant des évaluation du système Même si l’industrie dispose d’un système de production flexible et robuste qui peut répondre aux besoins du marché, la négligence du facteur temps aura des conséquences sur les performances et par conséquent sur la rentabilité.

Nous démontrons alors comment il est possible de déterminer l'ensemble de toutes les séquences de travaux à partir d'une exploration du graphe des marquages accessibles en déterminant le déroulement de tous les enchaînements qui nous conduire vers un produit évacué.

À partir de l’analyse de graphe d’accessibilité, il est facile d’extraire le chemin le moins, le plus et le moyen “coûteux”, c’est- à-dire selon le temps de fabrication avec la représentation de l’influence de l’aspect aléatoire dans le SdP. On se ramène ainsi à traiter un problème du plus court chemin dans le graphe, puisque chaque chemin représente un ordonnancement possible.

Pour cela on doit répondre à ces deux questions :

1. Quel est l'ordre optimal de production pour assurer une grande fluidité de ses opérations ?
2. Quel est l’influence de l’aspect aléatoire dans le système de production flexible ?

***3-Exploration du graphe d’accessibilité***

L'espace des marquages accessibles tient lieu d'espace d'états. L’espace d'états permet de représenter tous les chemins possibles pour la réalisation de la commande de production. Prenons le cas d’exemple pour la réalisation d’un seul type de produit P1 décrit par la gamme G1 avec :

G1 : op1 op3 op2   
 op1 est effectué soit par M1   
op2 est effectué par soit M1 soitM2  
 et op3 est effectué par M2.

Il serait évident que plusieurs chemins possibles permettant tous la réalisation du produit P1 et plus précisément, nous aurons une infinité des chemins possibles puisque à chaque états du système nous pouvons avoir un aspect aléatoire (panne/réparation). Ces chemins sont présentés par le graphe d’accessibilité ci-dessous.

PS : Dans ce qui suit on ne considère pas les chemins qui ont deux pannes successives

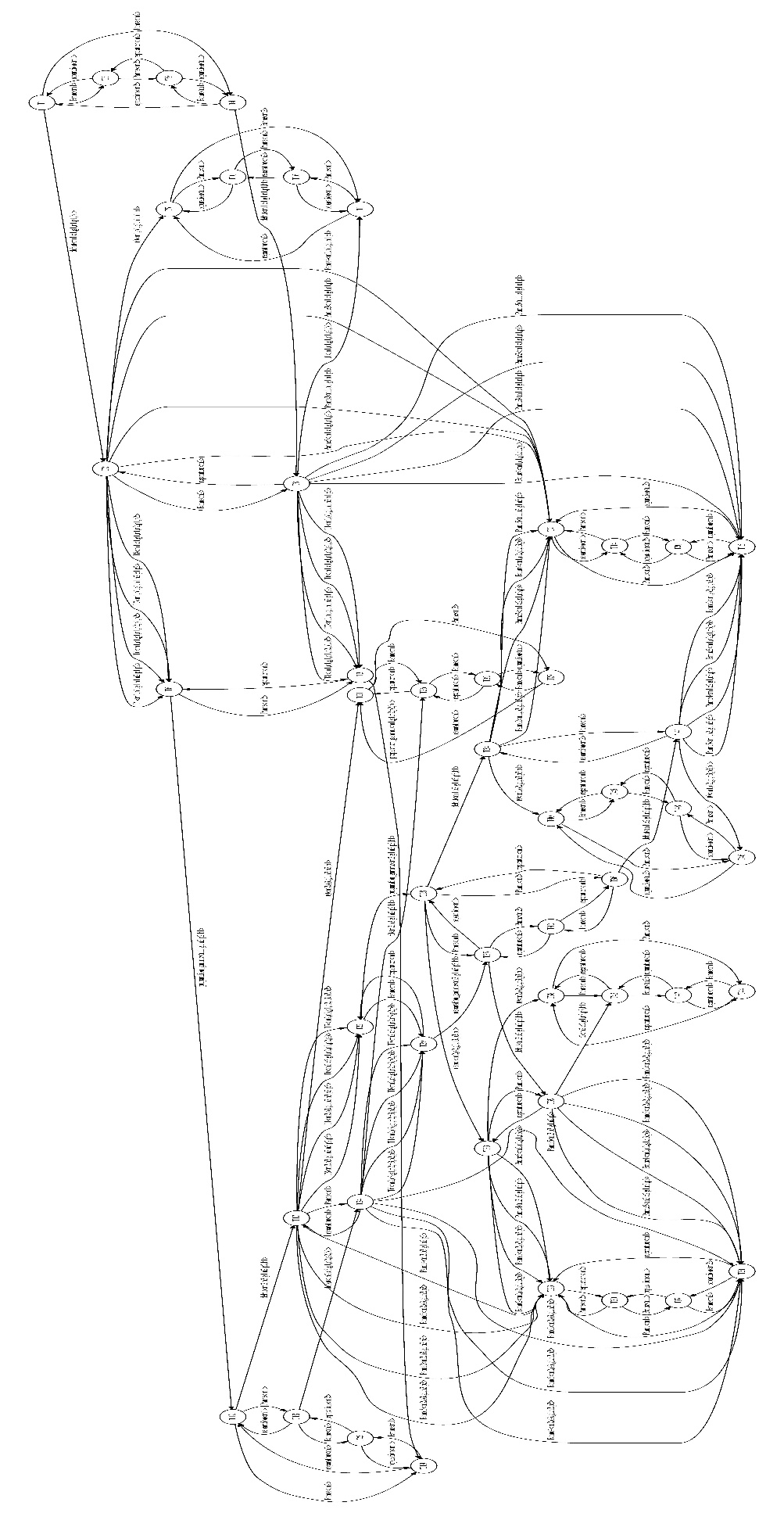


Figure 22 : Graphe de marquage

Tous les marquages T38, T40, T43, T44, T45, T46, T47 et T48 de ce graphe d’accessibilité (figure 22) représentent les marquages finaux qui nous ramènent au produit final P1 de la gamme G1 et la description de ces marquages est (la description complète dans l’index):

MARKING T1 (Marquage initial) Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p1,n1,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T38 Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T40 Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T43 En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T44

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T45 En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T46 En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T47 En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T48 En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

1. **Perte du produit**

D’après le graphe d’accessibilité il y a 3 chemin qui nous ramènent au deux état T7 et T23 dont lesquelles il y a une perte de produit P1, comme nous avons déjà vu dans le chapitre précèdent au cour de l’exécution de l’opération les machines peuvent avoir un aspect aléatoire (Panne des machine avec un taux μ), alors ces chemin sont :

**C1 : T1🡪 T2 🡪 T7  
C2 : T6🡪T12🡪T17🡪T23  
C3 : T6🡪T12🡪T17 🡪T22🡪T28🡪T34🡪T7**

1. **évaluation de performance des Systèmes de production**

**4.1 analyse de graphe d’accessibilité**

Le graphe d’accessibilité démarre à partir d’un marquage initial. Donc, à partir de ce marquage, toutes les transitions sensibilisées sont répertoriées, chacune d’entre elles est  
sélectionnée, et de nouveaux états sont générés. Cette procédure est répétée pour tous les marquages générés par le marquage d’origine. Tous les nouveaux marquages sont conservés et pour chacun d’entre eux, on explore les transitions sensibilisées qui en découlent, de façon itérative. Lorsqu’un des marquages générés correspond à l’état final recherché, les caractéristiques du chemin conduisant à cet état sont gardées en mémoire. D’une façon générale, le graphe des marquages accessibles est défini comme le graphe dont les nœuds sont les marquages accessibles et dont les arcs sont les noms des transitions impliquées dans les tirs menant d’un marquage à un autre, chaque arc relie un marquage à un autre qui est immédiatement accessible par une transition. L'état du système modélisé par le RdPS est représenté par le vecteur de marquage définissant le nombre de jetons que contient chaque place. L’évolution de l’état dynamique du système correspond donc à une évolution du marquage. Cette évolution se produit par le franchissement de transition : l'occurrence d'un événement correspond le franchissement d'une transition (si certaines conditions sur le marquage de ses places amont sont satisfaites). La figure 22 représente le graphe d’occurrence (accessibilité) pour la réalisation d’un produit de type P1 selon la gamme G1

Le fait d’avoir la liste des nœuds finaux est insuffisante pour la détermination du nœud le moins et le plus coûteux. Pour cela, nous devrons indiquer pour chaque nœud final son temps, qui indique le temps de la réalisation des produits commandés. Donc, une fois que nous avons les informations nécessaires à notre parcours, il nous reste que l’extraction des nœuds et cela en triant tous les nœuds finaux dans un ordre croissant par rapport à leur temps dans le but d’extraire le premier, la moyenne et la dernière nœud ordonné qui correspond au nœud le moins, la moyenne, et le plus coûteux.

Cette première étape permet de parcourir tous les nœuds finaux partir du graphe d’accessibilité, le tableau ci dessous représente les temps de tous les chemins possibles pour la fabrication du produit P1.

|  |  |
| --- | --- |
| Chemin | Temps/s |
| S1 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 5.8 |
| S2 : T1 🡪 T3 🡪 T1 🡪T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 9.1 |
| S3 : T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 8.7 |
| S4 : T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 6.8 |
| S5 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 9 |
| S5 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 7.6 |
| S6 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 6.5 |
| S7 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T40 | 6.9 |
| S8 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 8.4 |
| S9 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 9.7 |
| S10 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 8.2 |
| S11 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 7.9 |
| S12 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 10.5 |
| S13 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T40 | 9.4 |
| S14 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 12 |
| S15 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T40 | 9.7 |
| S16 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T43 | 10.8 |
| S17 : T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T44 | 11 |
| S18 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 15.7 |
| S19 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 12.6 |
| S20 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪T22 🡪 T28 🡪 T34 🡪 T40 | 11.9 |
| S21 : T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 13.5 |
| S22 : T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 14.7 |
| S23 : T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 10.6 |
| S24 : T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 12.5 |
| S25 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 20.3 |
| S26 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 14.7 |
| S27 : T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 15.9 |
| S28 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 18.3 |
| S29 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 17 |
| S30 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 19.1 |
| S31 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T40 | 16.4 |
| S32 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 18.6 |
| S33 : T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 21 |
| S34 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 17.7 |
| S35 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 19.9 |
| S36 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 23.2 |
| S37 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 23.9 |
| S38 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 25.5 |
| S39 : T1🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 26.5 |
| S40 : T1 🡪 T3 🡪 T1 🡪 T2 🡪T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 28.3 |
| S41 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 24.6 |
| S42 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 22.6 |
| S43 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 25.4 |
| S44 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 25.7 |
| S45 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 29 |
| S46 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T40 | 30.1 |
| S47 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 🡪 T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 | 36 |

Tableau 6 : les chemins de fabrication du produit P1

Pour avoir les nœuds le mieux et le plus coûteux on met tous les temps qui ramènent au produit finit dans un ordre croissant. Le premier temps dans la liste sera correspond au chemin le moins coûteux, le dernier temps sera correspond au chemin le plus coûteux.

Liste de temps par ordre croissant : 5.8, 6.5, 6.8, 6.9, 7.6, 7.9, 8.2, 8.4, 8.7, 9, 9.1, 9.4, 9.7, 10, 10.5, 10.6, 11.9, 12, 12.5, 12.6, 13.5, 14.7, 15.7, 15.9, 16.4, 17, 17.7, 18.3, 18.6, 19.1, 19.9, 20.3, 21, 22.6, 23.9, 24.6, 25.4, 25.5, 25.7, 26.5, 28.3, 29, 30.1, 36 .

On constate que la durée du chemin le moins coûteux est 5.8 qui correspondent au chemin S1 : T1 🡪 T2 🡪 T6 🡪 T12 🡪 T17 **🡪** T22 🡪 T28 🡪 T34 🡪 T40, et le temps le plus coûteux et 36 et son chemin S47 : T1 🡪 T3 🡪 T1 🡪 T2 🡪 T8 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 **🡪** T24 🡪 T17 🡪 T22 🡪 T29 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40 et à la fin le chemin avec un cout moyen est S27 avec un temps 15.9 et S 27 : T1 🡪 T2 🡪 T6 🡪 T13 🡪 T6 🡪 T12 🡪 T18 🡪 T12 🡪 T17 **🡪** T24 🡪 T17 🡪 T22 🡪 T28 🡪 T36 🡪 T28 🡪 T34 🡪 T41 🡪 T34 🡪 T40

On remarque que le chemin optimal est le chemin qui ne contient pas d’aspect aléatoire (aucun panne de machine) par contre le chemin le plus coûteux contient 8 pannes est c’est le pire cas.  
Pour voir l’influence de l’aspect aléatoire dans SdP nous allons comparer nos résultats avec un système de production flexible sans l’aspect aléatoire.

***4.* Comparaison avec un système de production flexible sans l’aspect aléatoire**

Après avoir calculé l’espace d’état dans notre outil de modélisation New GreatSPN nous devrons être capables de fournir à l’utilisateur la séquence d’ordonnancement optimal afin de la représenter graphiquement.  
Pour cela, nous considérons le même exemple de commande de production P1 avec un modèle qui ne considère pas l’aspect panne (figure 23)

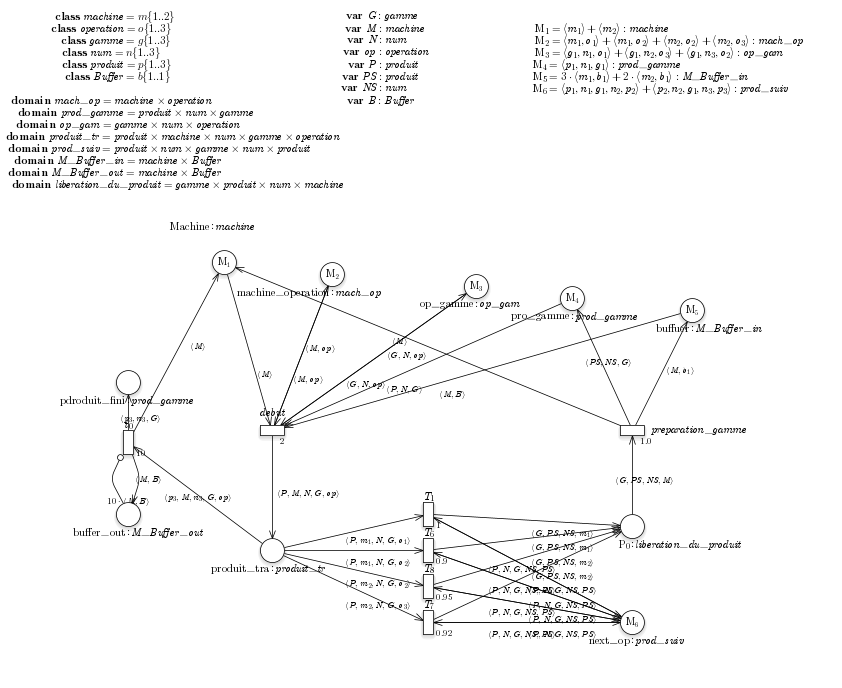
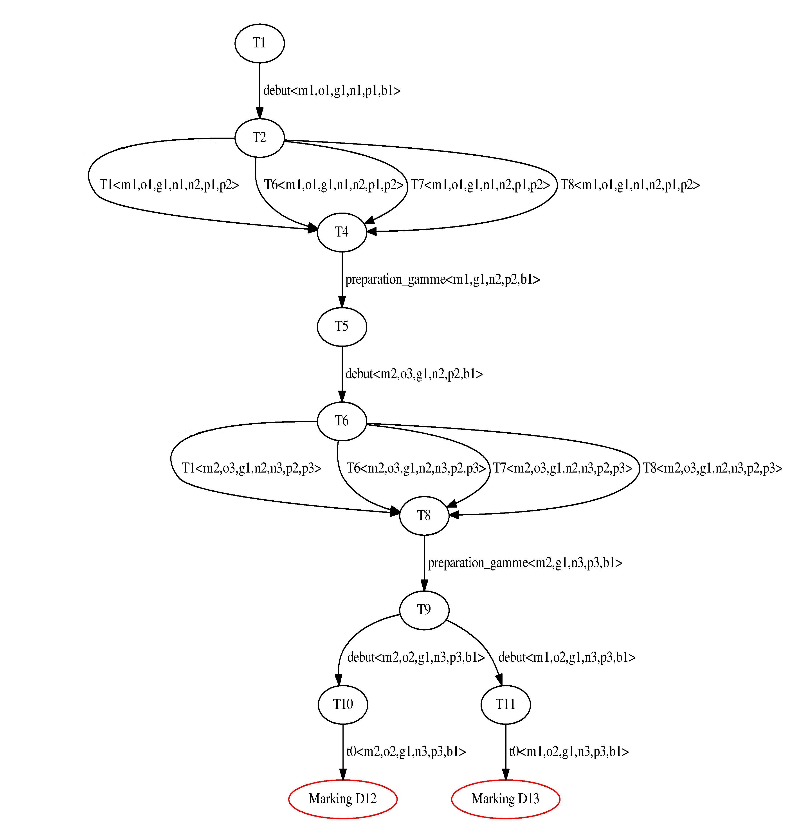


Figure 23 : Model de SdP sans l’aspect aléatoire

Pour ce model et d’après le graphe d’accessibilité (figure 24) il n’y a que deux chemin possible qui nous ramènent à la réalisation de produit final : S1 :T1🡪T2🡪T4🡪T5🡪T6🡪T8🡪T9🡪T10 et S2 : T1🡪T2🡪T4🡪T5🡪T6🡪T8🡪T9🡪T11 se sont les chemins optimaux pour l’évacuation du produit final. Et avec une simulation de ce model le temps de fabrication 5.5s presque le même temps minimum du model précèdent.

  
figure24 : graphe d’accessibilité

On remarque que le graphe d’accessibilité de l’atelier qui ne considère pas l’aspect aléatoire contient un nombre minimal d’états et que deux chemin pour avoir un produit final par contre le graphe d’accessibilité de l’atelier qui considère l’aspect aléatoire contient un grand nombre d’état et une infinité de chemin pour la réalisation de produit final qui permet de nous décrire tous les scénarios possibles dans l’atelier.

1. **Autre analyse de performance**

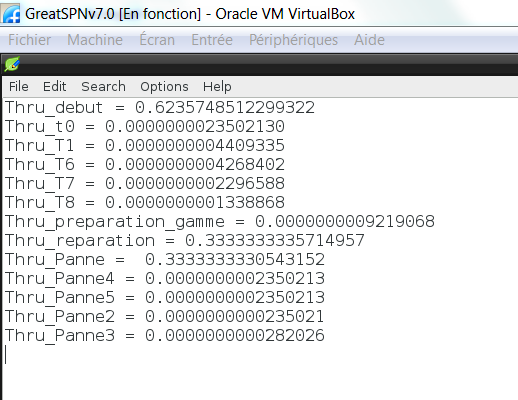
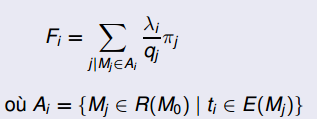


Figure25 : taux de franchissement des transitions

Le fichier production.sta (figure 25) générer par New GreatSPN permet de donner le taux de franchissement de chaque transition (transition throughput), ce taux est calculer selon l'équation: 

On constate qu'a l'état idle les machines peuvent tomber en panne avec 33% puisque Thru\_debut = 0.62 et Thru\_Panne = 0.33 et ou cour de l'exécution de l'opération les machinent peuvent tomber en panne avec 7.5% Puisque la somme de moyen de franchissement de panne est 28.10-9 et la somme de bon fonctionnement 35.10-8.rappelons que la disponibilité d'une machine à être en état d'accomplir sa tâche dans des conditions de temps déterminées sous les aspects combinés de sa fiabilité, maintenabilité et de l'organisation de sa maintenance.

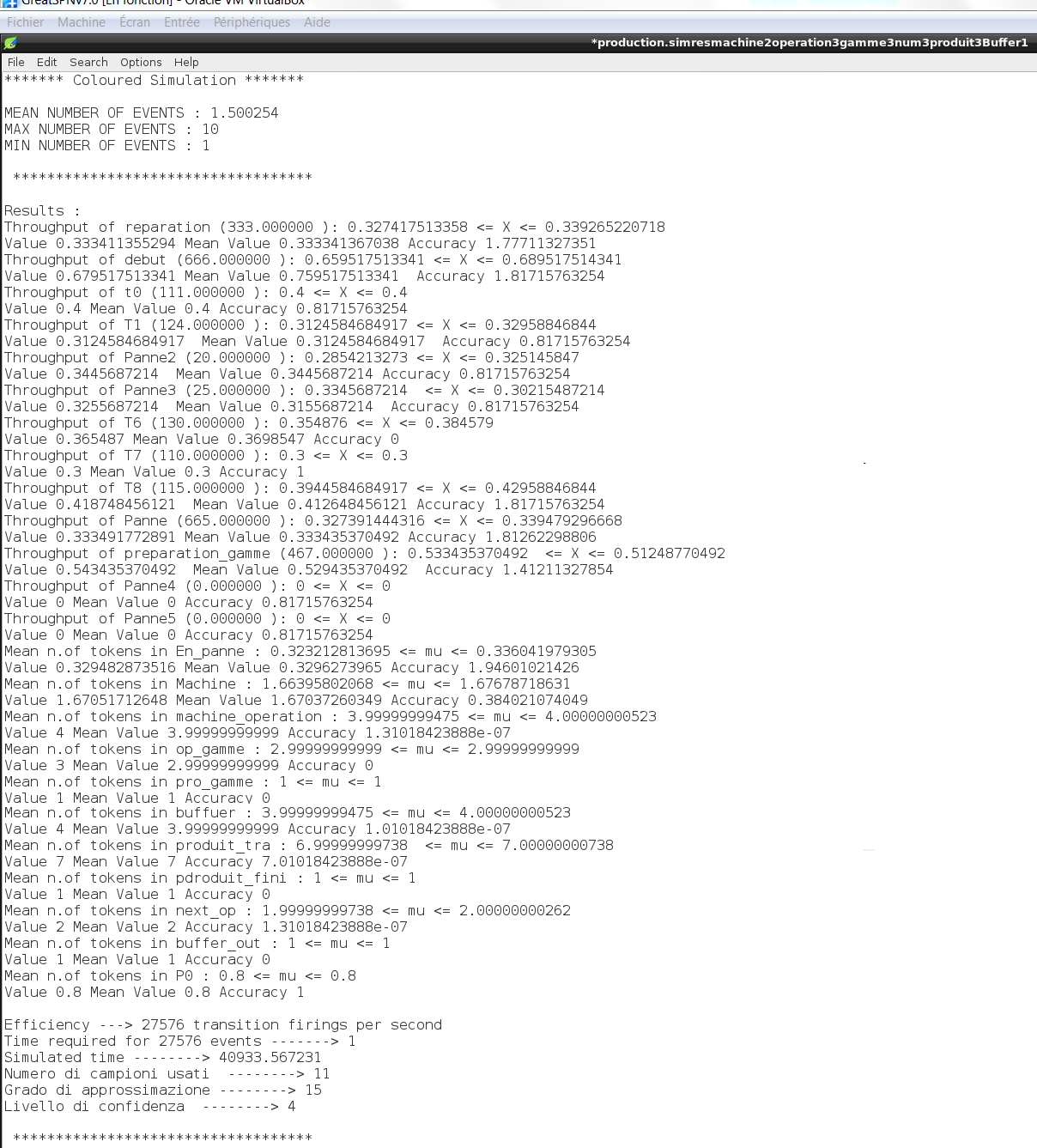


Figure 26 : indice de performance

Ce fichier représenter par le figure 26 permet de déterminer le temps de franchissement de chaque transition (le temps maximum, temps minimum et le temps moyen) aussi il permet de gêner les taux aléatoire pour chaque place.

Par ailleurs, il est important de rappeler que ces indicateurs ne permettent pas d’assurer la  
compétitivité et la réussite de l’entreprise. Ils ne sont qu’un outil de compréhension, maîtrise,  
pilotage, autrement dit un outil d’aide à la décision dans l’entreprise. Donc ces divers indicateurs  
constituent une aide précieuse pour suivre les performances de l’outil de production, et pour  
prendre des décisions et pour mesurer les améliorations.

**Conclusion**

Dans ce chapitre, nous avons exploité, dans une première étape, le modèle RdPCT que nous avons proposé pour la détermination de l’ensemble des séquences d’ordonnancement en calculant l’espace d’état (graphe de marquage accessible). Par la suite, nous avons comparé les résultats avec un système de production flexible sans l’aspect aleatoire. Ensuite nous avons présenté l’importance du l’aspect aléatoire dans l’atelier flexible. Finalement, nous avons donné une analyse de performance de SdP flexible.

**Conclusion générale**

**Index**

**La description du graphe d’accessibilité (figure)**

MARKING T1

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p1,n1,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T2

Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)produit\_tra(1<p1,m1,n1,g1,o1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T3

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p1,n1,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T4

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p1,n1,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T5

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p1,n1,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T6

Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)P0(1<g1,p2,n2,m1>)

MARKING T7

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T8

En\_panne(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)produit\_tra(1<p1,m1,n1,g1,o1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T9

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p1,n1,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T10

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p1,n1,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T11

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p1,n1,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T12

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p2,n2,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T13

En\_panne(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)P0(1<g1,p2,n2,m1>)

MARKING T14

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T15

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T16

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p1,n1,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T17

Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)produit\_tra(1<p2,m2,n2,g1,o3>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T18

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p2,n2,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T19

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p2,n2,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T20

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T21

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p2,n2,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T22

Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)P0(1<g1,p3,n3,m2>)

MARKING T23

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T24

En\_panne(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)produit\_tra(1<p2,m2,n2,g1,o3>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T25

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p2,n2,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T26

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p2,n2,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T27

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p2,n2,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T28

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p3,n3,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T29

En\_panne(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)P0(1<g1,p3,n3,m2>)

MARKING T30

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T31

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T32

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p2,n2,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T33

Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)produit\_tra(1<p3,m2,n3,g1,o2>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T34

Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)produit\_tra(1<p3,m1,n3,g1,o2>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T35

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p3,n3,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T36

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p3,n3,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T37

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T38

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T39

En\_panne(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)produit\_tra(1<p3,m2,n3,g1,o2>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T40

Machine(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T41

En\_panne(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)produit\_tra(1<p3,m1,n3,g1,o2>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T42

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p3,n3,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T42

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)pro\_gamme(1<p3,n3,g1>)buffuer(3<m1,b1>2<m2,b1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)

MARKING T43

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T44

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T45

En\_panne(1<m1>)Machine(1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T46

En\_panne(1<m2>)Machine(1<m1>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)

MARKING T47

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(3<m1,b1>1<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m2,b1>)

MARKING T48

En\_panne(1<m1>1<m2>)machine\_operation(1<m1,o1>1<m1,o2>1<m2,o2>1<m2,o3>)op\_gamme(1<g1,n1,o1>1<g1,n2,o3>1<g1,n3,o2>)buffuer(2<m1,b1>2<m2,b1>)pdroduit\_fini(1<p3,n3,g1>)next\_op(1<p1,n1,g1,n2,p2>1<p2,n2,g1,n3,p3>)buffer\_out(1<m1,b1>)