REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : GLSD13/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Génie Logiciel et Systèmes Distribués (GLSD)**

Analyse en arrière de marquages accessibles dans les réseaux de Pétri colorés pour le diagnostic à base de modèles causaux

---

Par :

LESSMI AHMED

Soutenu le 26/06/2022 devant le jury composé de :

Ouaar Hanane	M.C.B	Président
Bennoui Hammadi	Prof.	Rapporteur
Hmidi Zohra	M.A.A	Examinateur

Année universitaire 2021-2022


بسم الله الرحمن الرحيم

Remerciement

Nous remercions avant tout Le Dieu

الحمد والشكر لله رب العالمين

أما بعد

Nous remercions Le Professeur .Hammadi

Bennoui qui nous a honorés son encadrement,

Nous remercions tous mes enseignants au

niveau Primaire, C.E.M, lycée, Université,

Nous remercions le membre de jury qui prit la

peine d'évaluer notre travail,

Nous remercions Ma femme et mes enfants,

ma famille et mes amis.

ملـــخص

في هذا التقرير، نقدم لكم أداة تستخدم نموذج شبكة بتري الملونة، اسمها CBPN (شبكة بتري السلوكية الملونة) لتشكيل نموذج للسلوك السببي، حيث نستغل تقنية التحليل العكسي أو التحليل إلى الوراء المعروفة بـ (CW-analyse) وهذا لإنشاء مخطط بياني للعلامات التي يمكن الوصول إليها للخلف، وهذا انطلاقا من علامات نهائية معطاة (التي تمثل الملاحظة)، ومنه يتم استخلاص الحلول عن طريق العلامات الأولية التي تم الحصول عليها في نهاية تشكيل المخطط البياني.

الكلمات المفتاحية:

- **التشخيص القائم على النموذج**
- **النماذج السببية**
- **شبكات بتري الملونة**

Résumé

Dans ce rapport, nous présentons un outil qui utilise un modèle Réseau de Petri coloré, nommé CBPN (Colored Behavioural Petri Net) pour représenter un modèle de comportement causal. Ensuite, exploiter une technique d'analyse en arrière (CW-analyse) pour construire le graphe de marquages accessibles en arrière à partir d'un marquage final donné (Manifestations, celui représentant l'observation). Les solutions seront extraites à partir des marquages initiaux obtenus à la fin de la construction d'un tel graphe.

Mots clefs :

- **Diagnostic à base de modèle,**
- **modèles causaux,**
- **Réseaux de Pétri Colorés**

Abstract

In this report, we present a tool that uses a Colored Behavioral Petri Net model, named CBPN (Colored Behavioral Petri Net) to represent a model of causal behavior. Then, exploit a backward analysis technique (CW-analysis) to build the graph of backward accessible markings from a given final marking (Manifestations, the one representing the observation). The solutions will be extracted from the initial markings obtained at the end of the construction of such a graph.

Keywords:

- **model-based diagnosis**
- **causal models**
- **Colored Petri Nets**

Table des matières

Liste des figures

Chapitre : 01 Diagnostic à base de RDP

Chapitre : 02 Diagnostic à base de RDPC

Chapitre : 03 Développement d'un outil de diagnostic

Liste des Algorithmes

Chapitre : 03 Développement d'un outil de diagnostic

Introduction Générale

Introduction Générale

Les demandes de performances des systèmes et de produits plus élevées, la qualité d'un côté et plus de rentabilité de l'autre côté, sont continuellement croissante. En plus des exigences insistantes pour plus de fiabilité des systèmes, de disponibilité et de sécurité. La prévention des pannes techniques graves est devenait alors un élément essentiel de l'étape de développement des systèmes et surtout pour la sécurité des sites critiques comme les usines chimiques, les réacteurs nucléaires, les engins spatiaux, les avions et ainsi de suite. Donc, le diagnostic des pannes est devenu un enjeu majeur exigé en raison de son importance en termes de fiabilité du système, sécurité et efficacité.

Un processus de diagnostic est concerné avec la génération automatisée d'un diagnostic qui explique un comportement incorrect du système considéré. Dans ce périmètre, le diagnostic basé sur un modèle (MBD), est un domaine de recherche de l'intelligence artificielle qui effectue le diagnostic des fautes à l'aide de modèles. L'idée principale du MBD consiste en la comparaison du comportement observé du système et celui qui peut être prédit à l'aide du modèle du système, qui peut être de tout type (basé sur la logique, les automates, les réseaux de Petri, … etc.).

En effet, les réseaux de Petri (PN) sont populaires dans ce domaine en tant qu'outil bien adapté qui offrent une vision claire et une description précise de la structure statique du système en plus de sa dynamique dans un modèle réseau via la structure du graphe et le jeu de jetons respectivement. Malheureusement, ces réseaux souffrent du problème de complexité ; c'est-à-dire qu'un modèle PN a tendance à devenir trop grand et donc difficile à analyser, même pour les systèmes de taille modeste. La raison derrière cela est l'utilisation d'un seul type de jetons (tous les jetons sont noirs), et cela peut être assez clair lorsqu'il s'agit de systèmes composés d'un certain nombre de processus similaires. En fait, un tel problème a déjà été rencontré en introduisant une classe de réseaux de haut niveau appelés réseaux de Petri colorés (CPN), où les processus similaires sont décrit de manière uniforme et succincte sans perdre la capacité de distinguer entre eux en introduisant la notion de jeton avec une couleur. Par conséquent, une transition dans un CPN peut se déclencher de plusieurs manières, et les arcs sont donc associés à des expressions, souvent des dépendances linéaires, pour déterminer les couleurs de jetons lorsqu'elle se déclenche.

Une analyse comportementale (accessibilité) des PN, et en particulier les CPN, peut être accomplis en avant ou en arrière selon la nature du domaine d'application. Evidemment, dans le diagnostic de panne, le processus de raisonnement nécessite d'être effectué à l'envers

en commençant des symptômes et à la recherche des causes possibles. Pratiquement, l'analyse d'accessibilité en arrière est le concept dual de celle en avant. Elle permet, à partir d'un marquage donné, de déterminer les marquages à partir desquels il est accessible. Remarquons qu'une telle analyse nécessite une inversion des transitions et par conséquent des expressions, qui est mathématiquement considérée comme un processus difficile et peut conduire à une explosion combinatoire.

Afin de faire face à un tel problème, une classe particulière de CPN appelée Colored Behavioral Petri Nets, CBPNs a été proposée dans [1, 2, 3]. Les CBPNs permettent de décrire le comportement causal du système à diagnostiquer. Ils sont caractérisés par des étiquettes associées aux transitions du modèle sous forme de matrices représentant leurs différentes manières de franchissement. Chaque ligne de la matrice associée à une transition correspond à une manière de son tirage. Ainsi, une expression d'arc devient une variable typée pour un arc d'entrée de transition et une fonction sélective pour un arc de sortie.

Comme méthode de diagnostic, les auteurs de [1, 2, 3] définissent une analyse dite Analyse-CW basée sur la construction en arrière du graphe de marquages accessibles. En commençant du marquage correspondant à l'observation, l'analyse-CW exploite les matrices associées aux transitions du modèle construit un tel graphe jusqu'à l'obtention de marquages initiaux où toute place marquée est une place source. Les diagnostics possibles sont extraits depuis ces marquages initiaux.

Le travail présenté dans ce mémoire a pour objectif d'implémenter cette technique d'analyse en arrière de marquages accessibles (Analyse-CW) comme méthode de diagnostic basée sur des modèles CBPNs.

Le reste du mémoire est organisé comme suit :

• Le chapitre 1 est consacré aux concepts préliminaires concernant le diagnostic basé-modèle et réseaux de Petri.

• Le chapitre 2 présente une extension de PNs utilisée dans ce document : les réseaux de Petri colorés (CPNs). Enfin, il passe en revue la littérature sur le diagnostic basé sur un modèle, présente les réseaux de Petri comportementaux colorés, citant les définitions formelles et une discussion informelle. Il fournit une description complète de l'analyse-CW détaillant l'utilisation des CBPN pour le diagnostic.

• Le chapitre 3 sera consacré à la réalisation de notre outil pour la modélisation et le diagnostic à base de CBPNs.

Enfin, ce mémoire terminera avec une conclusion générale qui donne un bilan et discute certaines perspectives du présent travail.

Chapitre 01 :
Diagnostic à base de RDP

1.1. Introduction

Ce chapitre a pour objectif de présenter les notions de base du diagnostic basé modèle (DBM) qui permet de déterminer les causes d'une défaillance entraînant la dégradation du système. En effet, le diagnostic établit un lien de cause-effet entre un symptôme observé et la défaillance constatée. Le chapitre sera orienté vers les travaux utilisant les RdPs comme outil de modélisation et de raisonnement.

1.2 Théorie du diagnostic basé-modèle

Généralement, deux types de connaissances de diagnostic ont été utilisés. Tout d'abord, la connaissance superficielle, ou connaissance experte, qui fait référence à des connaissances empiriques sur la base de nombreuses observations et expériences.

La spécification d'une telle connaissance peut prendre la forme d'un ensemble d'observations défauts ou symptômes qui leur sont attribués les diagnostics possibles, comme des règles si-alors. Les approches de diagnostic basées sur des connaissances superficielles sont connues aussi sous le nom de systèmes experts de première génération.

Deuxièmement, la connaissance approfondie qui fait référence à la connaissance de la structure interne du système en termes de composants et les interactions possibles entre eux. Il prend la forme d'un modèle mathématique, qui nous permet de prédire le comportement du système pour toute condition d'entrée admissible. Les approches basées sur l'utilisation de telles connaissances sont connues sous le nom d'approches basées sur des modèles.

Le diagnostic basé-modèle est devenu une approche générale qui à été adopté avec succès dans une variété de domaines d'application. Selon le point de vue de l'approche MBD, le processus de raisonnement est effectué sur la base d'un modèle mathématique du comportement du système étudié ainsi qu'une observation de la façon dont le système se comporte réellement. Un problème de diagnostic est alors pointé du doigt lorsqu'il y a des écarts entre l'observation obtenue et le comportement prédit (en utilisant le modèle du système), l'idée de l'approche est entièrement présentée dans la Fig. 1.1.

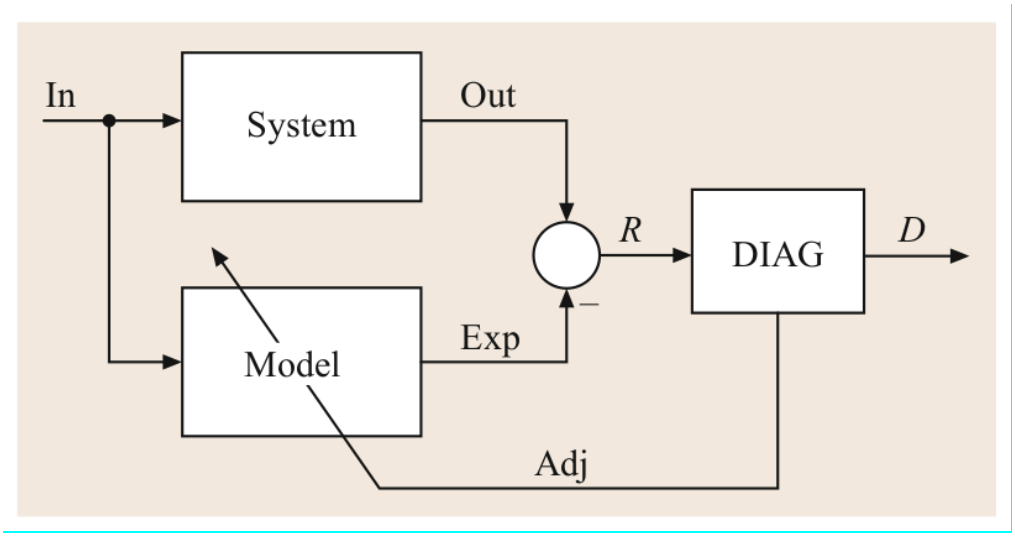


Figure 1.1: A presentation of diagnostic activity

Étant donné la même valeur d'entrée pour le système réel et son modèle. La sortie du système (obtenu par une observation) est comparée à celle attendue (générée en utilisant le modèle du système). Dans le cas d'aucune différence entre les résultats obtenus (sorties), on peut supposer que le système fonctionne correctement. Au cas où une différence significative entre le comportement observé et celui prédit prend place, il faut constater une incohérence entre le comportement réel du système et son modèle. Ici, un mauvais comportement du système est détecté, ce qui implique l'apparition d'un défaut (en supposant que le modèle du système est correct).

1.3. Approches de diagnostic basé-modèle

Certes, le modèle du système à diagnostiquer est le plus recherché lors de la construction d'un système de diagnostic basé-modèle. Ce modèle peut décrire le comportement du système dans le cas sans fautes (comportement correct), ou il peut donner la connaissance sur quel défaut peut survenir et les conséquences qu'il peut provoquer (comportement fautif), ou éventuellement les deux. Généralement, un modèle de système est orienté composants, c'est-à-dire le système est considéré comme un ensemble de composants qui interagissent entre eux. C'est le cas où le modèle n'est pas conçu spécifiquement à des fins de diagnostic, aucune des informations sur le comportement du système en présence de défauts n'est donnée, ainsi diagnostiquer un système consiste à définir les parties du système responsables de certains comportements inattendus. Une autre vue possible du modèle de système pourrait être celle décrivant les relations causales entre les défauts et les symptômes, et avec cela il exploite le modèle causal du système à diagnostiquer.

Afin de représenter formellement l'approche du diagnostic basé-modèle, la logique du premier ordre avec égalité sera utilisée dans ce qui suit.

1.3.1 Diagnostic basé à cohérence

Dans l'article "A Theory of Diagnosis from First Principles" de Raymond Reiter, où il a proposé les notions de base du raisonnement du diagnostic sur la base d'analyse de l'incohérence entre le comportement réel du système et le comportement prédit (en utilisant son modèle). Dans cette théorie, le modèle du système est déjà disponible, il peut être construit pour des raisons autres que le diagnostic, comme dans le cas d'un artefact où les modèles créés lors de la conception pourraient être exploités pour le diagnostic. Ainsi, la description du système peut être composée de deux parties : la structure du système, et le comportement de chaque composant.

1.3.2 Diagnostic abductif

Une autre approche qui est le diagnostic abductif, où le raisonnement habituellement adopté dans le domaine médical consiste à utiliser le comportement défaillant du système. Expliquer une inconduite, un symptôme, c'est ici trouver un ensemble de causes qui impliquent logiquement le symptôme lui-même, non seulement en étant cohérent avec lui, comme dans le diagnostic basé sur la cohérence. À cet égard, le modèle comportemental (BM) du système décrit ce qui se passe en cas de défauts, lorsque le système s'écarte de son comportement. Nous définissons l'ensemble de modes de comportement défaillants pour chaque composant c comme suit : mode(c) = $\{m_1,..., m_n\}$, où $m_1,...,m_n$ désignent les modes de défaillance possibles. Ainsi, le BM du système contient, outre la structure du système, une description pour chaque mode de comportement défaillant de chaque composant (l'un de ces modes pourrait être le mode inconnu auquel aucun modèle n'est associé), il est donné comme des implications décrivant les relations causales entre les défauts et leurs causes. Il convient de noter qu'en raison de l'absence de comportement correct, aucune différence entre les comportements observé et prédit ne peut être détectée.

1.3.3 Approche d'intégration (unified framework)

Une idée concernant les deux approches était que l'exploitation des informations sur les défauts dans le diagnostic basé sur la cohérence, et des informations sur le comportement correct dans abductive diagnostic. Inclure dans un modèle orienté composants des informations sur défauts correspond à la description, accompagnée du bon comportement des composants du système, aussi leurs défauts éventuels et leurs conséquences. Ainsi, les

modèles de bon comportement ont commencé à être dotés de modèles de fautes, et les modèles de causalité ont commencé inclure une description du comportement nominal. De plus, les deux approches ont été intégrés, et se sont révélés être les extrêmes d'un large éventail de définitions possibles du diagnostic. Il convient de noter que la causalité les modèles ont été largement adoptés dans cette approche en tant que BM du système a diagnostiqué.

1.3.4 Modèles causaux

Dans les modèles causaux, le comportement d'un système peut être décrit au moyen d'un ensemble d'états et de relations entre ces états. Chaque état modélise une composante de l'état global du système, et il est caractérisé par un ensemble fini de valeurs admissibles. Les relations décrivent les relations de cause-effet entre ces états. à des fins de diagnostic, [9] indique qu'il est utile de distinguer au moins entre trois types d'états dans le modèle :

– États initiaux : ils correspondent à des entités qui n'ont pas de causes dans le modèle. Dans le cas d'un modèle de comportement anormal, ils représentent les perturbations initiales conduisant le système à un dysfonctionnement donné.

– États internes : correspondant aux conséquences des états initiaux. Ils sont attachés à des composants du système qui ne sont pas susceptibles de faire partie d'un diagnostic, car ils peuvent être expliqués par les éléments d'états initiaux ;

– Manifestations : correspondant à certains états finaux considérés comme observables ou résultats mesurables des états internes et représentent ainsi tous les symptômes attendus dans le cas de modèles défectueux.

Dans cette optique, accomplir un diagnostic signifie expliquer un ensemble de manifestations en termes d'états initiaux, en utilisant les relations de cause-effet décrites dans le modèle.

D'un point de vue logique, un modèle causal du type décris précédemment peut être représenté par un ensemble de clauses sans récursivité données dans un formalisme du premier ordre. Plus précisément, chaque instance d'un l'état ou manifestation est représenté par un atome clue comme s(v) où s est un symbole d'état ou de manifestation et v une valeur admissible. De plus, admissible_values(s) désigne l'ensemble de valeurs possibles que s peut prendre et modelled_values(s) ⊆ admissible_values(s), est l'ensemble de valeurs modélisées. Afin d'aborder le processus de diagnostic plus efficacement, [10] propose de traduire l'ensemble des clauses définissant un modèle de comportement causal à un réseau de Petri, et donc au lieu d'adopter une certaine forme de calcul symbolique, les auteurs exploitent les

techniques classiques d'analyse des réseaux de Petri comme le graphe d'accessibilité ou l'analyse d'invariants pour faire face au problème.

1.4. Diagnostic à base de réseaux de Petri

1.4.1. Réseaux de Petri

Historiquement, Vers les années 1960, les réseaux de Petri ont été introduits par Carl Adam Petri dans son thèse de doctorat sous forme de graphes orientés bipartis destinés à la modélisation de systèmes concurrents, asynchrones, distribués et/ou parallèles. Les réseaux de Petri sont bien adaptés comme outil offrant une description claire et précise de la structure statique du système outre sa dynamique dans un modèle via la structure du graphe et le jeu de jetons respectivement.

Un autre aspect intéressant à leur sujet est qu'ils sont conçus pour aider l'analyse du système, et donc différentes techniques, basées sur le graphe l'accessibilité et des techniques algébriques, ont été développés pour les étudier. Côte à côte avec la description formelle et analyse, ces réseaux fournissent une représentation graphique pratique du système étudié qui comprend : des places (cercles), des transitions (rectangles), jetons (points noirs) attribués aux places et arcs en tant que relations entre les places et les transitions. La définition formelle d'un modèle réseau de Petri et les concepts sous jacents sont présentés dans ce qui suit.

Définition 1.1 : Un **RdP** est un triplet $N = (P, T, F)$ où :

- $P \cap T = \emptyset$.
- $P \cup T \neq \emptyset$.
- $F \subseteq ((P \times T) \cup (T \times P))$.

P est un ensemble fini non vide de places (représentées graphiquement par des cercles),

T est un ensemble fini non vide de transitions (représentées graphiquement par des rectangles ou des barres),

F représente l'ensemble des arcs (représentées graphiquement par des flèches) reliant les places aux transitions et vice-versa.

Dans un ***RdP***, la fonction de poids W est définie par $W : F \rightarrow N^*$, si $\forall f \in$ F/ $W(f) = 1$ le réseau est dit ordinaire.

Pour tout $x \in P \cup T$, l'ensemble d'entrées de x est définit par ${}^{\bullet}x = \{y \in P \cup T \mid (y, x) \in F\}$ et l'ensemble de sorties $x^* = \{y \in P \cup T \mid (x, y) \in F\}$ et pour que x ne soit pas isolé

$|{}^*x| + |x^*| \geq 1$, si ${}^*x \neq \emptyset$ x est dit source, alors que si $x^* = \emptyset$ x est dit puits.

Le marquage d'un réseau de Petri est une fonction $\mu : P \rightarrow \mathbb{N}$ définissant pour chaque place le nombre de jetons existant dans celle-ci.

La dynamique du réseau est décrite par le déplacement de jetons dans les places selon la définition suivante de règles de sensibilisation et de franchissement:

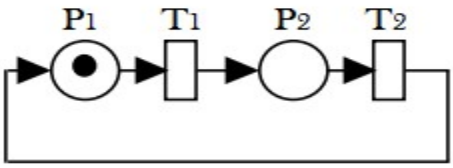


Figure 1.2 : Exemple d'un Réseau de Petri

Définition 1.2. Soit un ***RdP*** ordinaire marqué (P, T, F, μ); Une transition $t \in T$ est dite franchissable (ou sensibilisée) depuis μ si et seulement si: $\forall p \in {}^\circ t, \mu(p) \geq (p, t)$**.**

Si t est sensibilisée depuis μ, alors t peut être franchie produisant un nouveau marquage μ' (on écrit $\mu\ [t> \mu'$**)** tel que pour tout $p \in P$ on a : $\mu'(p) = \mu(p) - W(p, t) + W(t, p)$.

Définition 1.3. Soit $N = (P, T, F)$ avec $n = |T|$ et $m = |P|$, la matrice d'incidence de N est une matrice $C = [c_{ij}]$ d'entiers de taille $n \times m$ tel que $c_{ij} = W(i, j) - W(j, i)$, $(i \in T, j \in P)$.

Un T-invariant est un vecteur compteur de franchissement d'une séquence σ où le réseau est stable (ne change pas son état après le franchissement de $\sigma : \mu\ [\sigma > \mu)$.

- Soit I un vecteur où $|I| = |T|$, I est un T-invariant ssi : $I \neq 0$ et $C.I = 0$.
- Le support d'un T-invariant I est un sous-ensemble de T, noté $\| I \|$, où $\|I\| = \{t \in T | I(t) > 0\}$.
- On dit qu'un T-invariant I est minimal s'il n'existe pas un autre T-invariant J où : $||J|| \subseteq ||I||$.

Définition 1.4. Le graphe des marquages accessibles de N depuis M_0, noté G (N, M_0) est le graphe dont les états sont les marquages accessibles tel qu'il existe un arc entre deux sommets M_1 et M_2 si et ssi $M_1\ [t_i M_2]$

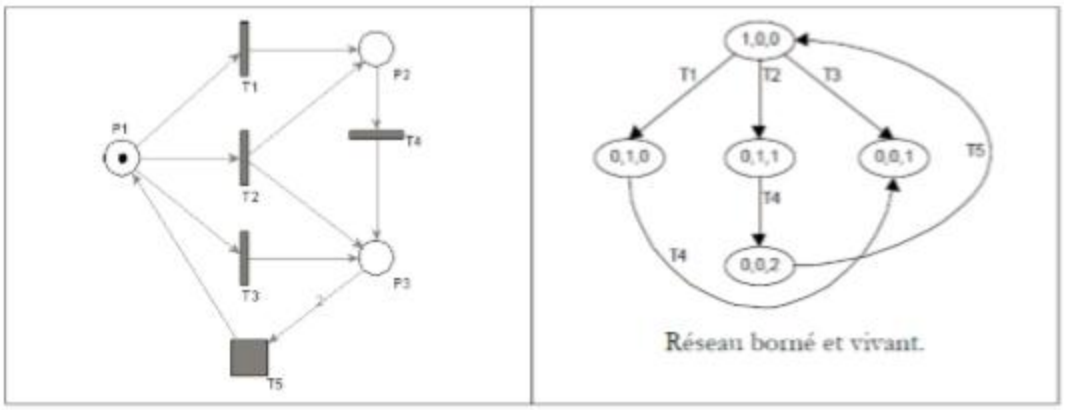


Figure 1.3 – Exemple de graphe de marquages d'un réseau de Petri.

1.4.2. Diagnostic avec PNs

Les réseaux de Petri ont été utilisés dans le diagnostic basé sur des modèles causaux via l'introduction d'une classe particulière de PNs appelée Behavioral Petri Nets (BPNs). En fait, les BPNs ont été proposés pour représenter le comportement causal du système à diagnostiquer. De plus, les notions formelles concernant les problèmes de diagnostic et leurs solutions ont été reformulées dans le cadre de BPNs. L'objet de cette section est de montrer comment un BPN est utilisé pour représenter un modèle de comportemental causal d'un système et comment accomplir un raisonnement de diagnostic basé sur un tel modèle.

Définition 1. 5 :

Un réseau de Petri comportemental (BPN) est un 4-uplet $N = (P, T_N, T_{OR}, F)$ tel que $(P, T_N \cup T_{OR}, F)$ est un réseau de Petri ordinaire acyclique qui satisfait les axiomes suivants :

1. $\forall p \in P\ (|\bullet p| \leq 1 \wedge |p\bullet| \leq 1)$
2. $\forall p_1, p_2 \in P\ ((\bullet p_1 = \bullet p_2) \wedge (p\bullet_1 = p\bullet_2) \rightarrow p_1 = p_2)$
3. $\forall t \in T_N\ (|\bullet t| = 1 \wedge |t\bullet| > 0) \vee (|\bullet t| > 0 \wedge |t\bullet| = 1)$
4. $\forall t \in T_{OR}\ (|\bullet t| \geq 2 \wedge |t\bullet| = 1)$

Un tel modèle de réseau se caractérise par les caractéristiques suivantes :

– chaque place correspond à une instance d'un des états du modèle et les transitions décrivent les relations de cause-effet entre les instances d'état correspondantes.

– une place source (c'est-à-dire $p : \bullet p = \emptyset$) correspond nécessairement à une instance d'état initial.

– une place de puits représente soit une instance d'état de manifestation, soit une instance d'état interne qui n'a aucune conséquence.

– le modèle est sûr, c'est-à-dire que n'importe quel place peut être marqué avec au plus un jeton.

– L'ensemble des transitions est partitionné en deux sous-ensembles T_N et T_{OR}. Les transitions de T_N (et-transitions) ont la même signification habituelle, tandis que les transitions dans T_{OR} (ou-transitions) sont destinées à représenter le connecteur logique OR et elles représentent des macros transitions dont la sémantique peut être donnée en termes de réseau de Petri avec des arcs inhibiteurs.

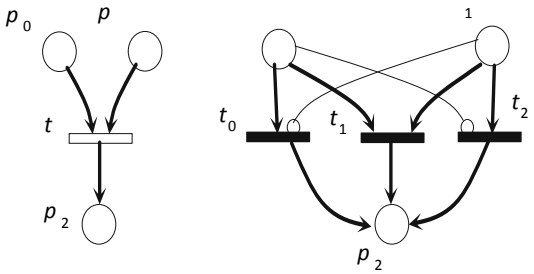


Figure 1.4 : Or-transition et son sémantique

De manière informelle, une transition $t \in T_{OR}$ (représentée graphiquement par une barre épaisse vide) est sensibilisée depuis un marquage si et seulement si au moins une de ses places d'entrée est marquée.

– une transition t est de type linéaire, fourchette ou jointure. Une transition t est une transition linéaire ssi elle a exactement une place d'entrée et une place de sortie. Une transition fourchette est celle avec seulement une place d'entrée et au moins deux places de sortie. Enfin, une transition de jointure est celle avec au moins deux places d'entrée et une seule place de sortie (par souci de simplicité et pour faire face au problème d'explosion, une transition fourchette a deux places de sortie et une transition de jointure a deux places d'entrée. Si $t \in T_{OR}$, alors il a deux places d'entrée et une place de sortie).

– La clôture transitive de la relation de flux (c'est-à-dire les arcs) est non réflexive.

– un marquage initial d'un BPN est un marquage sûr μ_0 tel que si $\mu_0(p) = 1$ alors p est une place source.

– Un BPN marqué est un couple (N, μ) où N est un BPN et μ est soit un marquage initial soit un marquage tel qu'il existe un marquage initial μ_0 et $\mu \in [\mu 0>$.

Dans cette vue, un problème de diagnostic à base de BPN (BPNDP) correspondant à un DP logique est défini comme BPNDP = (N, P^{ini}, P^+, P^-), où N est le réseau correspondant à un BM causal, P^{ini} est un ensemble de places sources dénotant les états initiaux dans BM, P^+ et P^- sont deux ensembles de places puits représentant les observations et correspondant donc respectivement à Ψ^+ et Ψ^-.

Avant de montrer comment caractériser les solutions pour un BPNDP donné, rappelons la définition suivante.

Définition 1.6 : un marquage μ d'un BPN tel qu'aucune transition n'est activée en μ est appelé un marquage final.

Théorème 1.1 : Dans un BPN marqué, il y a exactement un marquage final.

La notion de solution peut maintenant être appréhendée comme suit :

Définition 1.7 : Soit un BPNDP = (N, P^{ini}, P^{+}, P^{-}), un marquage initial μ^{ini} est une solution à BPNDP si et seulement si le marquage final μ de N couvre P^{+} et zéro-couvre P^{-}.

1.4.3 Analyse-BW

Une conséquence évidente de non réflexivité de la relation de flux est qu'elle définit un ordre partiel, noté "≺", sur les transitions d'un BPN ; étant donné deux transitions t_1 et t_2 : $t_1 \prec t_2 \Leftrightarrow t_1 F^{+} t_2$ où F^{+} désigne la fermeture transitive de la relation de flux. Pour mettre en place un raisonnement de diagnostic qu'est par nature hypothético-déductif sur un modèle BPN, la simple utilisation de l'analyse d'accessibilité vers l'avant n'est pas adéquate pour résoudre un problème de diagnostic donné, car un tel problème a été considéré comme abductif (c'est-à-dire comme un problème de post-diction) plutôt que comme un problème de prédiction. En conséquence, le problème de l'analyse de l'accessibilité vers l'arrière sera étudié, ce qui signifie que commencer à partir d'un marquage donné, il faut chercher les marquages d'où provient celui-ci accessible. Il devrait être clair que cela peut être accompli en tirant vers l'avant des transitions du réseau inverse correspondant (c'est-à-dire le réseau obtenu en inversant la direction des arcs). En particulier, étant donné un ensemble d'observations OBS représenté par des ensembles Ψ^{+} et Ψ^{-}, afin de déterminer une solution on part d'un marquage μ tel que $\mu(p) = 1 \rightarrow p \in P^{+}$.

A partir de ce marquage, il faut rechercher un marquage initial μ_0 ayant les caractéristiques d'une solution. Notez que cela ne signifie pas que μ doit être accessible à partir de μ_0 mais seulement que μ doit être un sous-marquage de μ , où μ est accessible à partir de μ_0 couvrant P^{+} et zéro couvrant P^{-}. Un tel processus doit construire un graphe d'accessibilité vers l'arrière à partir de μ en prenant en compte cet aspect.

Afin de déterminer efficacement ces marquages initiaux, une technique de rétro-analyse particulière appelée Analyse-BW est utilisée. Elle exploite deux types de jetons appelés respectivement jetons normaux et jetons inhibiteurs.

Le sens de la normalité de jetons est comme d'habitude : nous pouvons associer une condition à une place et un jeton normal dans cette place signifie qu'une telle condition est satisfaite. Au

contraire, les jetons inhibiteurs représentent les conditions qui ne sont certainement pas remplies dans le cas examiné (c'est-à-dire ; ils représentent toutes les valeurs de paramètres qui sont différentes de celles observées).

Si une place dénotant une condition C est marquée avec ce type de jeton, alors ¬C est vrai. Par conséquent, lorsqu'une place est vide, aucune contrainte n'est imposée sur la condition associée. Cela correspond à considérer une logique à trois valeurs dont les valeurs de vérité sont {vrai, faux, inconnues}.

En se référant à la section précédente, dans un BPN causal, une transition t est activée, en avant, à un marquage μ (et donc ça peut tirer) ssi t est sensibilisée en μ et $\nexists t' \prec t$ tel que t' est sensibilisée en μ. Cela correspond à imposer un ordre de priorité sur les transitions. Néanmoins, comme nous devons tirer les transitions à l'envers, la relation inverse de l'ordre partiel doit être considérée : étant donné deux transitions t_1 et t_2 : $t_1 \succ t_2 \Leftrightarrow t_2 \prec t_1$.

Marquage b-w. Le concept de marquage b-w a été défini comme une fonction μ de l'ensemble des place à l'ensemble {b, w, 0} ; où μ(p) = b signifie que la place p est marquée par un normal jeton (noir), μ(p) = w signifie que la place p est marquée d'un jeton inhibiteur (blanc) et μ(p) = 0 signifie que p est vide.

Une particularité de l'analyse-BW est la possibilité de forcer le tir arrière de transitions de type fourchette (T_N). De manière informelle, cela signifie que si une transition fourchette t a au moins une place de sortie marquée alors toutes ses places de sortie qui sont vides doivent être marquées avec le même type de jeton.

Nous avons besoin d'un ensemble de règles de tir qui seront appliquées à l'envers sur le modèle réseau. Ces règles partent d'un marquage b-w correspondant à la mauvaise conduite constatée (un marquage μ s.t $\mu(p) = 0 \Rightarrow p \in P^+ \cup P^-$) et se terminant dans un marquage b-w initial dont les places marquées appartiennent nécessairement à P^{ini} (c'est-à-dire ; correspondant aux états initiaux du modèle comportemental). Un tel marquage b-w initial représente la solution au problème de diagnostic donné. Dans ce cas, les places sources vides représentent des conditions initiales qui ne sont pas significatives pour le cas en cours d'examen, tandis que les places sources marquées par des jetons normaux et inhibiteurs représentent les conditions initiales qui se sont avérées vraies et fausses respectivement.

Afin d'obtenir un tel marquage b-w initial, nous devons construire le graphe de marquages en arrière en appliquant l'ensemble des règles de tir schématisées dans la figure suivante.

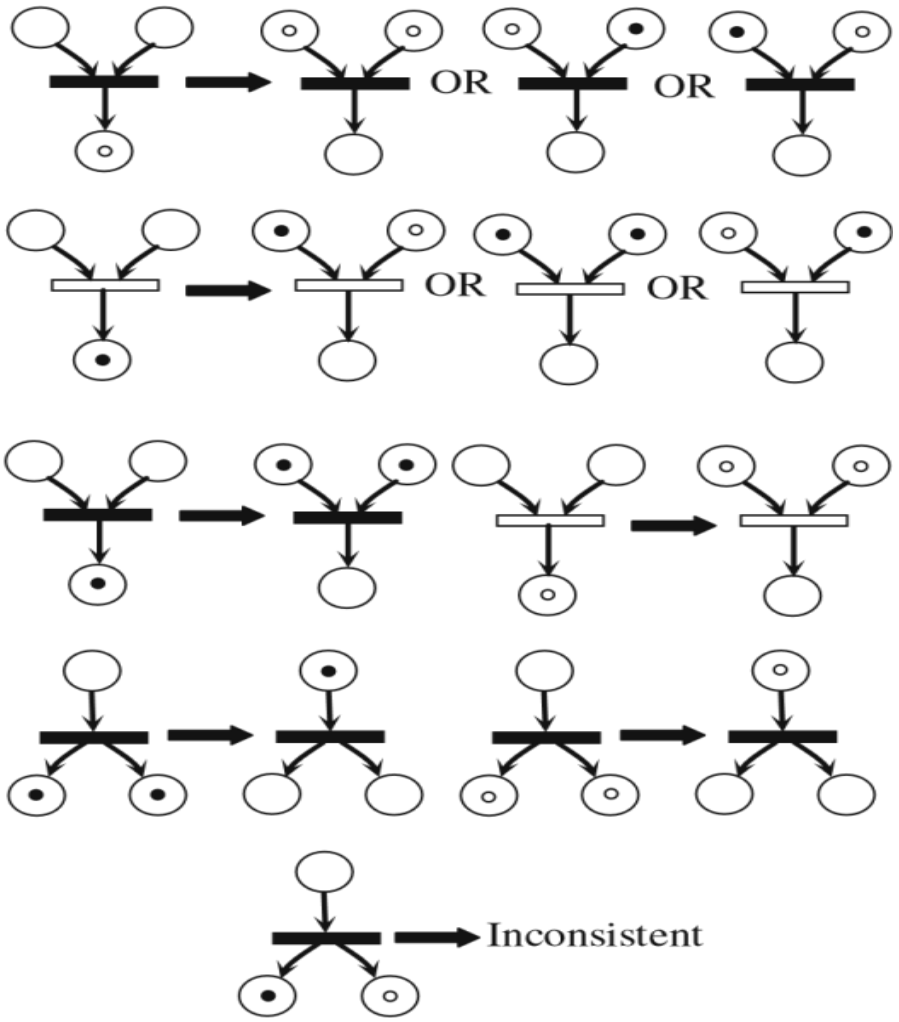


Figure. 1.5: Règle de franchissement en arrière

1.5. Conclusion

Le diagnostic des pannes des systèmes est une tâche très difficile. Dans ce chapitre, on a vu comment les réseaux de Petri sont utilisés comme outil de modélisation et de raisonnement. En particulier, on a détaillé une technique d'analyse d'accessibilité en arrière comme méthode de diagnostic basée sur des modèles BPNs.

Le prochain chapitre représente un pliage de cette technique aux réseaux de Petri colorés.

Chapitre 02 :

Diagnostic à base de RDPC

2.1. Introduction

Pour les problèmes où le modèle réseau est grand ou composé de certaines parties identiques, il est bien connu que les PN colorés (CPN) sont bien adaptés par rapport aux PN classiques.
Nous présentons le modèle CBPN défini par [1,2,3] pour la représentation du comportement causal du système) diagnostiquer, ainsi que la technique d'analyse d'accessibilité dite CW-analysis pour effectuer un raisonnement de diagnostic à base du graphe d'accessibilité.

2.2. Les Réseaux de Petri colorés (RDPCs)

Les réseaux de Petri colorés sont des réseaux de haut niveau. Les CPN conservent toujours, comme points forts des PN, le fondement de la notation graphique et les bases primitives pour modéliser la concurrence, la communication et la synchronisation. Comme les CPNs sont essentiellement des PNs, il est important de garder à l'esprit qu'un modèle CPN est défini par un ensemble fini de places, un ensemble fini de transitions et un ensemble fini d'arcs comme connexions entre les places et les transitions. Chaque place à un type associé et peut contenir un ou plusieurs jetons dont chacun porte une valeur donnée appartenant au type de la place. Par convention, les types sont des jeux de couleurs et donc les jetons ont des couleurs.

Chaque place a son propre marquage qui est un multi-ensemble de jetons colorés qui sont présents dans une telle place. La somme des marques de places individuelles donne le marquage du modèle CPN. Le marquage change au cours de l'exécution du modèle par l'exécution d'une transition. Lorsqu'une transition est franchie, il supprime les jetons de ses places d'entrée et il ajoute des jetons à ses places de sortie.

Dans un CPN, les jetons supprimés (resp. ajoutés) sont déterminés par une expression d'arc associé à l'arc sortant (resp. entrant) de (resp. vers) une place. Une expression d'arc est construite à partir de variables typées, de constantes, d'opérateurs et de fonctions et elle est évalue à un multi-ensemble de jetons colorés. De plus, il peut être attaché à chaque transition une expression booléenne (avec des variables) appelée une garde qui spécifie les liaisons pour lesquelles elle est évaluée à vrai. Une liaison est une affectation de valeurs de données aux variables libres apparaissant dans l'expression d'un arc entrant ou un garde d'une transition. Une liaison d'une transition peut s'écrire sous la forme : ($v_1 = d_1; v_2 = d_2, \ldots, v_n = d_n$) où pour i $\in$ 1..n : v_i est une variable et d_i est la valeur attribuée à v_i.
Une transition *t* est activée, prête à se produire, s'il existe une liaison telle que :

1) le résultat d'évaluation de chacune des expressions d'arc d'entrée est présent sur la place d'entrée correspondante,

2) le garde (s'il existe) est satisfait. Le tir de t ajoute à chaque place de sortie un multi-ensemble de jetons colorés auquel l'expression sur l'arc de sortie correspondant est évaluée.

Avant de rappeler la définition formelle d'un CPN, on rappelle le concept de multi-ensemble. Un multi-set est un ensemble où les éléments individuels peuvent apparaître plus d'une fois.

Définition 2.1. (Multi ensemble) Un multi ensemble m, sur un ensemble non vide S, est une fonction $m \in [S \rightarrow N]$ représenté formellement par : $\sum_{s \in S} m(s)'s$.

- $\forall s \in S: s \in m$ ssi $m(s) \neq 0$.
- S_{MS} est l'ensemble de tous les multi-ensembles finis sur S.

Un ensemble d'opérations applicables sur les multi-ensembles est défini comme suit :

$\forall m, m_1, m_2 \in S_{MS}$ et $\forall n \in N$:

$|m| = \sum s \in S \; m(s)$.

$m_1 + m_2 = \sum s \in S \; (m_1(s) + m2(s))'s$.

$m_1 \neq m_2 = \exists s \in S: m_1(s) \neq m2(s)$.

$m_1 \leq m_2 = \forall s \in S: m_1(s) \leq m_2(s)$, (défini de manière analogue pour $\geq$).

Ainsi, un CPN est donné par :

Définition 2.2. Un réseau de Petri coloré (CPN) est un 6-uplet $N = (\sum, P, T, A, C, G)$ où:

- $\sum$ est un ensemble non vide de types (ensembles de couleurs).
- P est un ensemble non vide de places.
- T est un ensemble non vide de transitions.
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$.
- A est un ensemble non vide d'arcs tels que : $A \subseteq (P \times T) \cup (T \times P)$.
- $C : P \rightarrow \sum$ est une fonction de mappage de couleurs, chaque place a un ensemble de couleurs.
- G est une fonction de mappage de gardes. Chaque transition a une expression booléenne.

Définition 2.3. Un CPN marqué est un couple (N; μ) où N est un CPN et μ est une fonction définie sur P telle que : $\mu(p) \in C(p)_{MS}, \forall p \in P$.

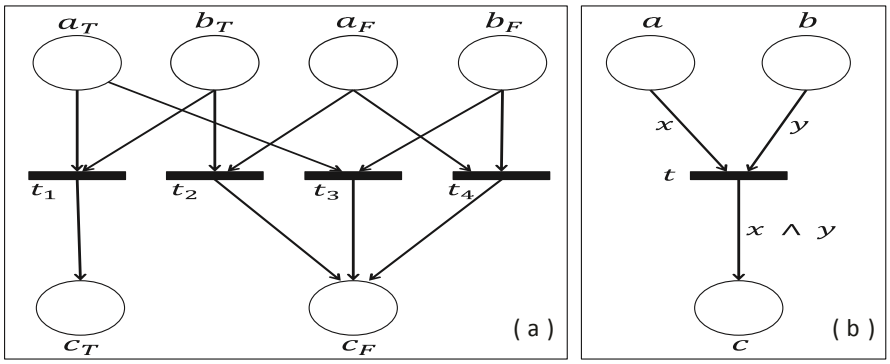


Figure 2.1. PN/CPN correspondant à l'exemple 2.1. (a) un modèle PN représentant une expression logique. (b) représentation CPN de (a).

Exemple 2.1 RDPC :

À titre d'exemple, considérons l'expression logique α ∧ b → c, où α, b et c prennent des valeurs booléennes (T pour vrai et F pour faux) et ∧ est le connecteur logique et, c est vrai uniquement lorsque α et b sont vrais, sinon il est faux. L'expression c peut être décrite par le réseau de Petri montré dans **Fig. 2.1(a)**. Comme α et b peuvent être réglés de quatre manières différentes, le PN correspondant à c a quatre transitions, par exemple, t_1 représente le paramètre lorsque α et b sont vrais (donné par les places α_T et b_T, respectivement), et donc la valeur de c est également vraie (représenté par la place c_T). Un modèle CPN de la même expression est affiché sur la **figure 2.1(b)**, où C(α) = C(b) = C(c) = {T, F}, et une expression est attachée à chaque arc rappelant que les x et y sont des variables sur un multi-set. Ainsi, l'expression x ∧ y indique que dans le cas où la transition t se déclenche, un jeton de couleur x et un jeton de couleur y sont détruits respectivement de α et b. En conséquence, les réglages possibles de l'expression d'arc x ∧ y sont T ∧ T, T ∧ F, F ∧ T et F ∧ F.

2.3. Réseaux de Petri coloré causal

Les CPNs sont utilisés pour représenter et analyser une grande variété de systèmes. Le concept de jeu de couleurs nous permet d'obtenir un modèle réduit. Alors qu'à travers l'expression associées aux transitions (gardes) ou aux arcs on définit les possibles combinaisons de couleurs de jetons d'entrée et de sortie pour une transition soit franchissable, et ainsi, décrivant l'évolution du système étudié. Toutes les techniques d'analyse définies pour les PNs classiques sont étendues aux CPNs. En particulier, l'analyse en arrière d'un CPN apparaît comme une inversion du modèle réseau et donc la réalisation de l'analyse vers l'avant. Le problème principal produit ici, c'est le tirage d'une transition. Lors de l'inversion des arcs, leurs expressions doivent être inversées aussi, et même pour la garde de la transition.

Une telle inversion dépend des expressions d'arcs d'entrée et de sortie. Cela peut conduire à une explosion combinatoire dans certains cas. Afin d'affronter ce problème, les travaux de [1, 2, 3] proposent d'associer à chaque transition une matrice décrivant les combinaisons possibles de couleurs des jetons d'entrée et de sortie. En conséquence, les expressions des arcs d'entrée d'une transition sont des variables typées et celles de des sortie sont des expressions simples qui sont extraient d'une matrice donnée et selon les jetons d'entrée. Dans ce qui suit, nous rappelons définition des CPNs causaux (CBPNs) caractérisés par des matrices associées aux transitions. Les CBPNs sont utilisés pour représenter le comportement causal du système sous étude.

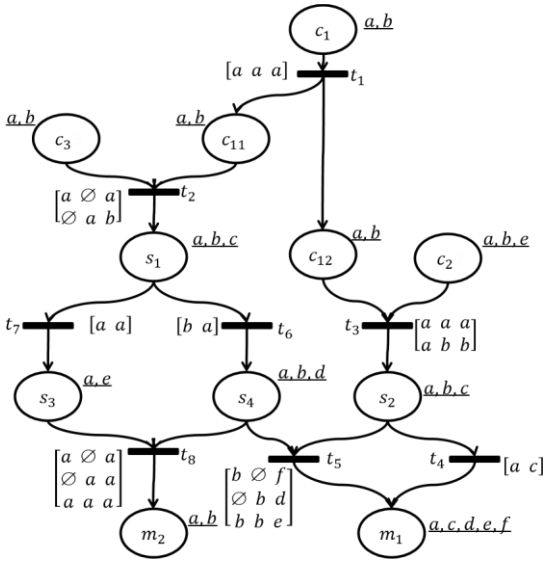


Figure 2.2. Exemple de CBPN.

2.4. Réseaux De Petri Comportemental Coloré (COLORED BEHAVIORAL PETRI NET)

Les modèles de réseaux de Petri sont des outils utiles bien connus, qui offrent une description claire et précise du système étudié. Néanmoins, l'utilisation d'un seul type de jeton provoque un problème de taille du modèle, qui devient assez complexe même pour les petits systèmes. On peut considérer ici comme un exemple, l'état d'une température moteur. Celui-ci peut prendre une valeur élevée, moyenne ou faible à un moment donné. Dans les PNs, chacune de ces valeurs est représentée par une place, et il peut avoir son propre chemin d'exécution, en tant que sous-réseau, à gérer. Des PNs colorés sont introduits, en tant que modèles de réseau de haut niveau, pour simplifier la structure du modèle. Généralement, les réseaux de Petri de haut niveau sont maintenant largement utilisés dans l'analyse théorique et pratique de divers

systèmes. La principale raison du succès de cette classe de modèles de réseaux est qu'ils permettent d'obtenir des descriptions beaucoup plus succinctes et gérables qui peuvent être obtenus depuis des PNs classiques. D'ailleurs, ils offrent toujours un large éventail de méthodes et d'outils d'analyse. Nous visons à utiliser les CPN pour saisir le comportement causal du système à analyser. En fait, cela est l'objet des CBPNs où les transitions sont étiquetées avec des matrices comme alternative pour déterminer les différentes façons dont une transition peut se déclencher. Une description complète des CBPNs est donnée ci-dessous.

2.4.1. Le Modèle CBPN

Définition 2.4. Un CBPN est un 6-uplet N = (∑, P, T, A, C, FW) où :

1) (∑, P, T, A, C) est un CPN.

2) FW : T → $MAT_{n,m}$(∑∪ {ε}).

3) A^+ (fermeture transitive de A) est non réflexive.

Un CBPN est un CPN particulier introduit principalement pour décrire le comportement causal du système étudié. Par conséquent, l'ensemble des places peut être partitionné en trois sous-ensembles : 1) causes initiales Ic, 2) états internes Is, et 3) manifestation Mn (P = Ic ∪+ Is∪+ Mn, où Ic = {p/p ∈ P, •p = ∅}, Mn ⊆ {p/p ∈ P, p• = ∅}, et Is = P \ (Ic ∪ Mn)). Les transitions peuvent être classées comme des transitions fourchettes, t ∈ T : |t•|>1, ou transitions de jointure, t ∈ T : |•t| ≥ 1. De plus, elles sont étiquetées avec leurs matrices de voies de tir.

Soit t une transition avec n voies de tir et m places auxquelles elle est connexe (m = |•t|+|t•|).

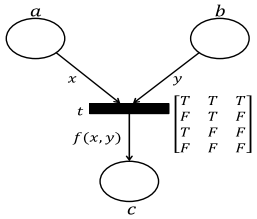


Figure 2.3. Exemple de CBPN.

La matrice des voies de tir d'une transition t, FW = [C_{ij}], est une matrice n x m de couleurs C_{ij} ∈$∪_ω$∈$∑^ω$, comprenant la couleur vide ε. Notez qu'une matrice FW ne contient que des couleurs. L'ensemble vide (qui est en fait un ensemble) ne peut pas être inclus. On nous impose donc d'introduire une couleur particulière que nous appelons la couleur vide ε pour représenter l'ensemble vide. La matrice FW contient une ligne pour chaque transition du modèle déplié (chaque façon de tirage) et une colonne pour chaque place connectée. Ainsi, le C_{ij} se compose de la couleur retirée de (ajoutée à) la place j lorsque la transition t est franchie

par rapport à la $i^{ème}$ manière de franchissement. Remarquons que si t est une transition fourchette alors il est prévu de diviser le jeton coloré de la place d'entrée (où plusieurs conséquences sont présentes). C'est pourquoi les couleurs d'une même ligne sont similaires. Alors que s'il s'agit d'une transition de jointure, on rencontre trois cas d'utilisation possibles. Le premier est comme usuel lorsque t est une conjonction de causes. La seconde consiste du Ou logique, où t devienne franchissable si au moins une de ses place d'entrée est marquée. Par conséquent, nous pouvons dériver le dernier cas, c'est-à-dire le Ou exclusif, où t devienne franchissable si et uniquement si une seule de ses places d'entrée est marquée. D'ailleurs, le FW(t) peut être décomposé en $FWin_n \times |\bullet t|$ et $FWout_{nx}|t\bullet|$ comme les sous-matrices d'entrée et de sortie, respectivement.

Définition 2.5. Un marquage initial d'un CBPN est un marquage sûr μ_0 ssi :
$\forall p \in P : \mu_0(p) = \emptyset \rightarrow p \in Ic$.

En tant que CPN, un CBPN marqué est une paire (N,μ), où N est un CBPN et μ est un marquage avec la propriété suivante : $\forall p \in P : |\mu(p)| \leq 1$.

Définition 2.6. Un CBPN marqué (N,μ) est dit sûr ssi :

$$\forall p \in P \; \forall \mu \in R\,(N, \mu_0) : |\mu(p)| \leq 1.$$

On dit que μ' (avec $\mu' \neq \mu$) est un sous-marquage de μ, on écrit $\mu \subseteq \mu'$, si $\mu'(p) = c \rightarrow \mu(p) = c$, pour $p \in P$, $c \in C(p)$.
Dans ce qui suit, nous définissons les règles de sensibilisation et de tirage. Il convient de noter qu'un arc d'entrée d'une transition t est étiqueté avec une variable typée, alors que la sortie se compose d'une fonction qui définit simplement le jeton coloré approprié à une place $p \in t\bullet$ en utilisant FW(t). Pour qu'une transition t soit sensibilisée, il doit être possible de trouver une liaison des variables qui apparaissent dans les expressions d'arc environnants de t. Soit b un $|\bullet t|$-vecteur dont la composante v_p, notée b(vp), représente la valeur affectée à une variable d'arc d'entrée E(p, t) qui est présent dans la place p. On note b une liaison si elle correspond à une ligne de FWin(t). Quand t se déclenche avec une liaison donnée, en utilisant FW(t), il supprime de chaque place d'entrée le couleur de jeton à laquelle la variable d'arc d'entrée correspondante est évaluée. De manière analogue, il ajoute à chaque place de sortie le jeton coloré à laquelle l'expression sur l'arc de sortie est évaluée. Il devrait être clair que A^+ dénote

la fermeture transitive de la relation de flux. La non réflexivité de A^+ signifie qu'un CBPN est acyclique, on définit donc un ordre partiel, noté « $\prec$ », sur les transitions d'un CBPN comme suit : Soit $t_1, t_2 \in T : t_1 \prec t_2 \Leftrightarrow t_1 A^+ t_2$.

Définition 2.7. Une transition $t \in T$ est activée dans un marquage μ, noté $\mu[t>$, ssi :
$\forall p \in \bullet t / E(p, t)<b> \leq \mu(p)$ et $\nexists t' \in T : t' \prec t$ s.t $\mu[t' >$.

Une transition activée $t \in T$ peut en outre se déclencher dans un marquage μ selon une manière de franchissement donnant un nouveau marquage μ', noté $\mu[t>\mu'$ avec

$$\mu'(p) = \mu(p) - E(p, t)<b> + E(t, p) <b>.$$

Deux transitions sont dites concurrentes ssi à chaque fois qu'elles sont toutes les deux activées, le déclenchement de l'une ne désactive pas l'autre. Formellement :

$$\exists \mu \in R(N, \mu_0) : \mu [t_1> \wedge \mu[t_2> \wedge t_1 \nprec t_2 \wedge t_2 \nprec t_1).$$

Définition 2.8. Un CBPN marqué est déterministe ssi $\forall \mu \in R(N, \mu_0) \forall t_1, t_2 \in T$
si μ [t1> et μ [t2> alors t_1 et t_2 sont concourantes.

Ainsi, étant donné un CBPN marqué (N,μ), on note par un pas l'ensemble des transitions activées et concurrentes en μ.

Définition 2.9. Un pas $s = \{t_1, \ldots, t_n\}$ peut se déclencher en μ donnant un nouveau marquage μ' tel que :

$$\mu' = \sum_{i=1}^{n} \mu i, \mu[\text{ti}>\mu\text{i} .$$

On appelle un marquage final μ_f le marquage où aucune transition n'est sensilisée.

Exemple 2.2. Afin d'aborder les principaux concepts d'un modèle CBPN, considérons comme exemple le réseau représenté dans la **Fig. 2.4**. Formellement, il est donné par
$N = (\sum, P, T, A, C, FW)$, où $\sum = \{\alpha, b, ré, n, h, l, r\}$, $P = \{p_1, \ldots, p_{10}, A, B, C, D\}$ avec $Ic = \{p_1, \ldots, p_3\}$ et $M_n = \{p_7, \ldots, p_9\}$, et $T = \{t_1, \ldots, t_8\}$. Les ensembles de couleurs attachées aux places sont les suivants :
$C(p_1) = C(p_3) = C(p_8) = \{h, l\}$, $C(p_2) = C(p_9) = \{\alpha, b\}$,

C (p_4) = C (p_6) = C (p_7) = {n, d}, et C (p_5) = {r}.

Chaque transition a sa matrice FW qui montre les différentes façons de tir à respecter. Il est à noter qu'on peut déterminer le type (fork et join) de chaque transition en inspectant uniquement la matrice associée. Par exemple, t_2 et t_3 sont des transitions de type fork, elles sont utilisées pour diviser certaines couleurs d'entrée. Au fait, t_2 et t_3 sont des transitions fictives, et donc, les places A, B, C, et D sont aussi fictives. Ainsi, les jeux de couleurs associés à ces places selon les ensembles de couleurs sont les suivants :

C(A) = C(B) = C (p2) et C(C) = C(D) = {n}. Les autres transitions sont t_5, t_7 et t_8 qui sont toutes des transitions de jointure.

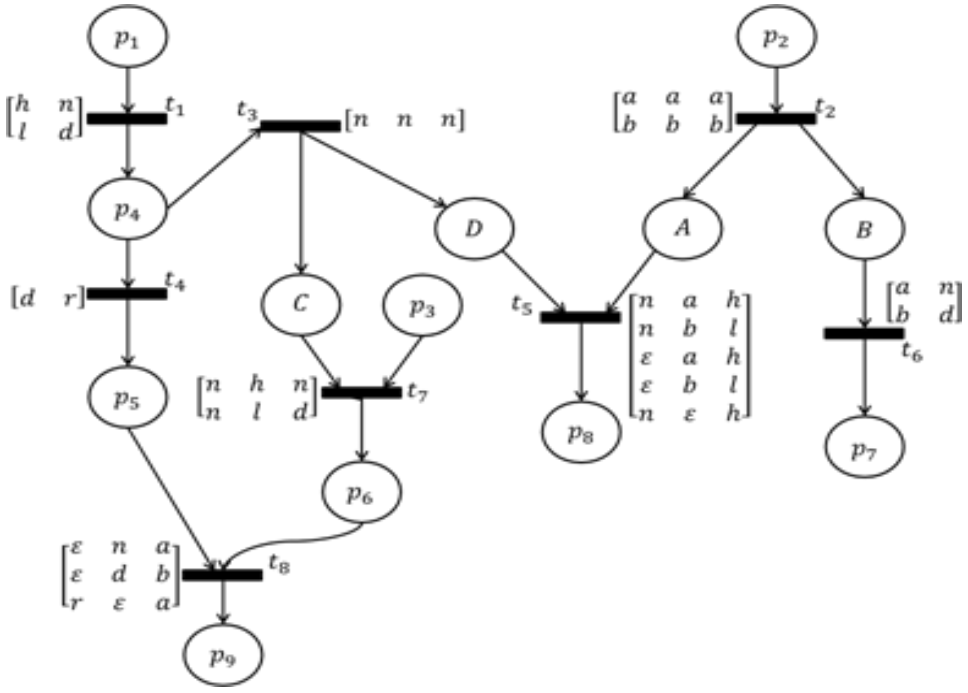


Figure 2.4. Exemple de CBPN.

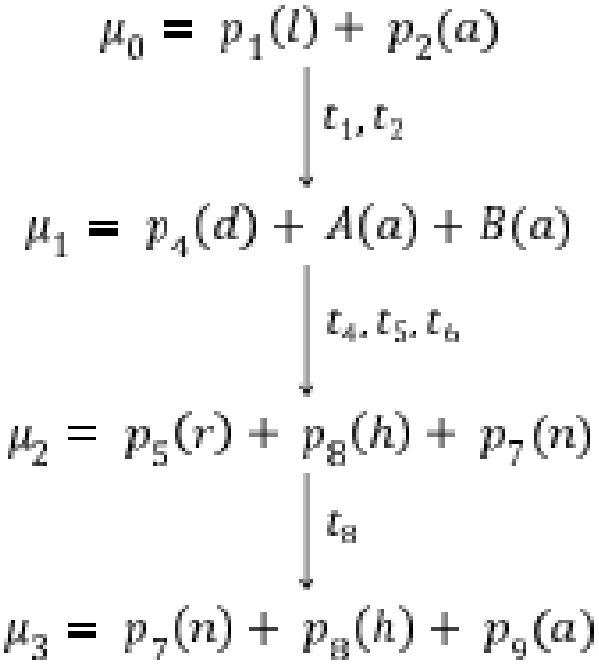


Figure 2.5. Graphe d'accessibilité du CBPN de la **Figure2.4.**

Réellement, il faut remarquer la différence entre ces transitions. La transition t_7 est prévue de la manière habituelle. Alors que la transition t_5 ressemble au connecteur logique OR, son activation nécessite le marquage d'au moins une de ses places d'entrée A ou D, et nous exclure la possibilité d'activer la transition t_8 lorsque les deux places p_5 et p_6 sont marquées, car p_5 et p_6 sont mutuellement exclusives.

Vient maintenant la simulation du comportement réseau. Considérez un marquage initial $\mu_0 = p_1(l) + p_2(\alpha)$, la notation p(c) signifie que la place p est marquée par un jeton de couleur c. Une étape $s = \{t_1, t_2\}$ est activée à μ_0. Le tir de s par rapport aux deuxième et première lignes de $FW(t_1)$ et $FW(t_2)$, respectivement, conduit à un nouveau marquage $\mu_1 = p_4(d)+A(\alpha)+B(\alpha)$. La **figure 2.5** montre le graphe d'accessibilité correspondant à μ_0. Notez que t_5 est activée à μ_1 même si seule la place A est marquée, t_5 est tirée par rapport au troisième ligne de $FW(t_5)$ obtenant la couleur α de A et fixant la couleur h dans p_8. Même chose pour t_8 lorsque p_5 devienne marquée. Enfin, μ_3 est un marquage final, où seules les places de Mn sont marquées.

2.4.2. La Technique d'Analyse-CW

Dans cette section, nous présentons une technique d'analyse particulière pour les CBPN que appelée ***CW-analysis***. Sans avoir besoin d'une inversion réseau, l-analyse-CW consiste en une accessibilité vers l'arrière effectuée par une simple manipulation des matrices FW avec l'utilisation d'une couleur inhibée. La raison derrière l'utilisation d'une couleur inhibée est que, généralement, l'analyse d'accessibilité en arrière peut conduire à une large gamme de marquages, certains d'entre eux sont inaccessibles dans le réseau d'origine, alors que d'autres n'ont aucun sens avec le cas étudié. Depuis chacun des marquages obtenus, une analyse en avant doit être réalisée pour vérifier si ce marquage est bien celui requis un ou pas. Pour faire ceci en une seule phase, le concept de couleur inhibée à été introduis. On cherche par ceci le blocage de la production de certaines couleurs pendant le processus d'analyse.

Définition formelle

Plus précisément, on entend par une couleur inhibée, appelée c^w pour $c \in \sum$, pour une condition qui est certainement insatisfaite dans le cas sous examen. Ainsi, une description supplémentaire du concept de marquage peut être utilisée. Soit $p \in P$ avec $C(p) = \{c_i, c_j\}$ et $\mu(p) = c_i$, le marquage de p peut être donné par $\mu(p) = c^w_j$. Une remarque vaut la peine sur de telles descriptions. Notez que le premier indique exactement le marquage réel de p (au moyen de couleurs normales). Alors qu'en utilisant les couleurs inhibées, un spectre de marques possibles de p est possible, $\mu(p) = c^w_j \rightarrow \mu(p) = c_i$ ou $\mu(p) = \emptyset$. En d'autres termes, nous nous concentrons, par là, sur telle couleur absente, donc, on s'en fiche quel que soit le marquage exact de p.

Nous utilisons l'exposant ($\_^B$) pour désigner le processus d'analyse en arrière. Étant donné une transition t, rappelons que la matrice FW(t) peut être décomposée en sous-matrices $FW_{in}(t)$ et $FW_{out}(t)$. Ainsi, $FW^B(t)$ correspond à FW(t) lorsque

$$FW^B{}_{in}(t) = FW_{out}(t) \text{ et } FW^B{}_{out}(t) = FW_{in}(t).$$

Pour une expression d'arc, $E^B(x, y)$ est une variable typée, v_y, si E(x, y) est une fonction sélective et vice-versa, avec $x, y \in P \cup T$.

En utilisant la matrice FW^B et les expressions E^B, une transition t est validée en arrière selon la définition suivante.

Définition 2.10. Étant donné un CBPN marqué (N, μ), une transition t ∈ T est activé en arrière à μ, (noté $\mu[t>^{B}$), ssi $\forall p \in t\bullet : E^{B}(t, p)<b> \leq \mu(p) \wedge \nexists t'> t : \mu[t' >^{B}$.

Il doit être clair que b fait référence à une liaison définie sur $FW^{B}(t)$, alors que « > » dénote la relation inverse de l'ordre partiel « ≺ » défini précédemment. Le tirage en arrière de t au marquage μ renvoie la progression réseau un pas en arrière, donnant un nouveau marquage μ' donné par: $\mu'(p) = \mu(p) - E^{B}(t, p)<b> + E^{B}(p, t)<b>$. Remarquez que la règle de tirage en arrière n'a aucun souci avec le type de couleurs qu'elles soient normales ou inhibées. Cependant, lorsqu'une transition se déclenche, elle produit le même type de couleurs qu'elle consomme.

Définition 2.11. Soit (N, μ) un CBPN marqué, on dit que μ est incohérent ssi :

$$\exists t \in T, \exists p, p' \in t\bullet : \mu(p) \neq \mu(p').$$

L'incohérence est un concept pertinent pour l'analyse d'accessibilité en arrière qui apparait quand certaines places en sortie d'une transition fork sont marquées de couleurs différentes, et en particulier la situation d'une place avec des marquages distinctifs.

Soit t ∈ T avec t• = {p, p'} et $\mu(p) = c_i$, la consistance de μ dépend du marquage de p'. Pour le cas où $\mu(p) = c_i^{w}$, μ est évidemment incohérent μ(p) ≠ μ(p'), il n'y a pas de couleur normale en commun ($\mu(p') = c_i^{w} \rightarrow \mu(p') = c_j$ ou μ(p') = ∅ tandis que $\mu(p) = c_i$).

Alors que si $\mu(p') = c_j^{w} \rightarrow \mu(p') = c_i$ ou μ(p') = ∅, alors il y a est une couleur normale c_i en commun entre μ(p) et μ(p'), ici μ est dit cohérent.

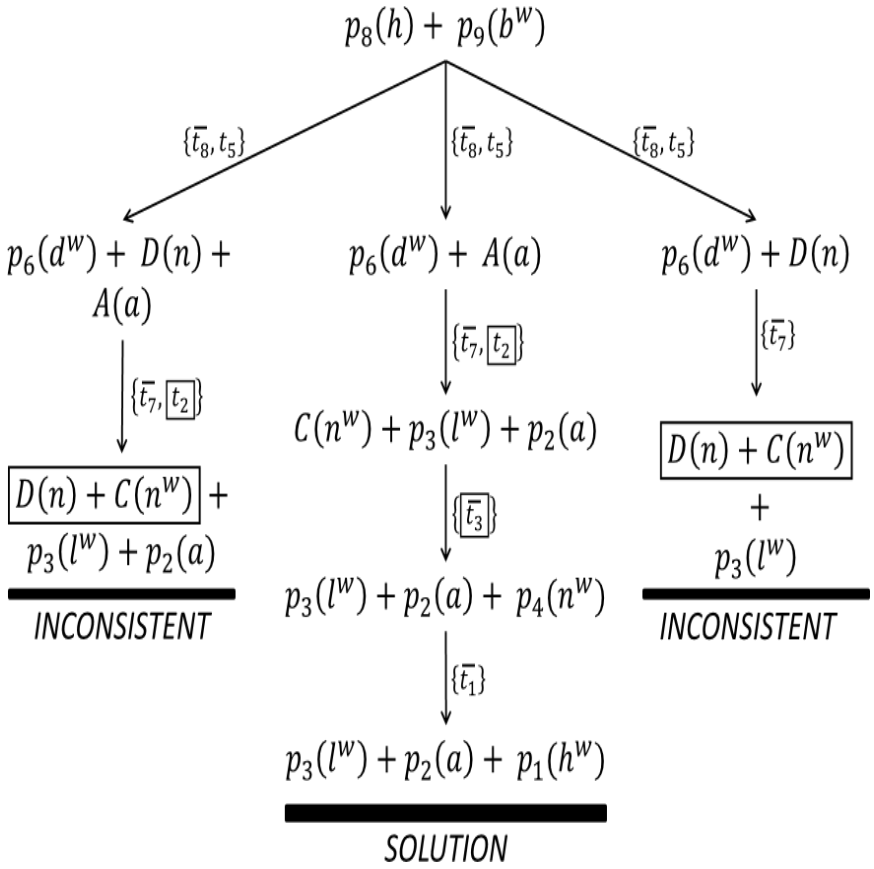


Figure 2.6. Graphe de l'analyse CW de l'exemple 2.2.

Dans le cas où certaines places sont marqués de la même manière alors que d'autres sont vides, on est devant le cas nommé transition forcée. Ici, on suppose que toutes les places non marquées, nécessaires au franchissement en arrière de la transition en question sont marquées par la même couleur que celles déjà marquées.

Définition 2.12 Soit un CBPN marqué (N, μ) et une transition fourchette t telle que •t = {p} et t• = {p_1, . . . , p_m}, on dit que t est forcée au marquage μ ssi :

1) t n'est pas activé en arrière à μ,

2) $\exists p_i (1 \leq i \leq m) | \mu(p_i) \neq \emptyset$,

3) μ n'est pas incohérent,

4) $\nexists t' > t$ où t est franchissable en arrière ou forcée à μ.

Si t est forcée en μ alors ∀p, p' ∈ t• tel que μ(p) = ∅ et μ(p') ≠ ∅, considérons μ(p) = μ(p').

Dans le cas des couleurs inhibées, une transition devienne forcée à un marquage μ s'il existe au moins une couleur normale en commun entre les places marquées. Ensuite, les places vides sont censées être marquées de cette couleur. Pour différentes couleurs en commun, un chemin dans le graphe d'accessibilité est construit pour chacun.

2.5. Conclusion

Le diagnostic des pannes des systèmes est une étape cruciale et une tâche difficile. Dans ce chapitre, nous avons présenté une approche basée sur les CPN pour diagnostiquer un système donné à base de son modèle de comportement causal.

L'objet du chapitre suivant est de développer un outil logiciel implémentant de telle approche.

Chapitre 03 :
Développement d’un outil de diagnostic

3.1. Introduction

Après avoir vu dans les chapitres précédents les différents concepts nécessaires à l'accomplissement de notre travail, nous passons maintenant à la partie conception et développement de notre système.

Pour construire un tel système, on doit passer par plusieurs étapes, et comme il s'agit d'un outil logiciel, il est préférable de passer par un cycle de développement d'un logiciel.La première étape dans le cycle de développement d'un logiciel est l'analyse des besoins. Sa fonction principale est de préciser les services qui seront rendus par le logiciel à l'utilisateur ainsi que l'ensemble de contraintes sous lesquelles ce logiciel devra fonctionner. Cette phase est suivie par la phase de conception (Conception générale etconception détaillée). Une bonne conception est la clé d'un développement de logiciel efficace. Un système bien conçu est facile à réaliser, à maintenir et à comprendre.

La dernière phase consiste en une réalisation, lors de cette étape on réalise un ensemble d'unités de programme, écrites dans un langage de programmation exécutable.

3.2. Analyse des besoins

L'analyse des besoins est la première phase dans le cycle de vie d'un logiciel. Elle consiste à définir les services qui seront rendus à l'utilisateur et les contraintes sous lesquelles ce logiciel devra fonctionner. Donc les services principaux offerts par notre outil sont :

1- Une interface graphique simple pour l'utilisateur permettant la saisie d'un modèleCBPN censé représenté le comportement causal d'un système à diagnostiquer.
2- Offrir de manière simple la saisie de l'ensemble des manifestations pour que la tâche du diagnostic s'effectuée.
3- La tâche de diagnostic doit être passée par :
   - ✓ Construction d'un graphe de marquages en arrière pour déterminer les marquages initiaux μ_{ini}.
   - ✓ Extraction des solutionsau problème donné depuis les marquages obtenusμ_{ini}.

3.3. Conception

La conception est un processus itératif à plusieurs niveaux.Elle commence par une conception générale qui peut être raffinée jusqu'à aboutir à un niveau moins abstrait à partir duquel il est simple de générer le code source (conception détaillée).

3.3.1. Conception générale

Notre système dans le diagnostic à l'architecture distribué peut être vu comme une collection des diagnostiqueurs attachés à un ensemble desous-systèmes qui ont des communications entre eux par un réseau de communication. Ces communications peuvent être spécifiées par le schéma suivant:

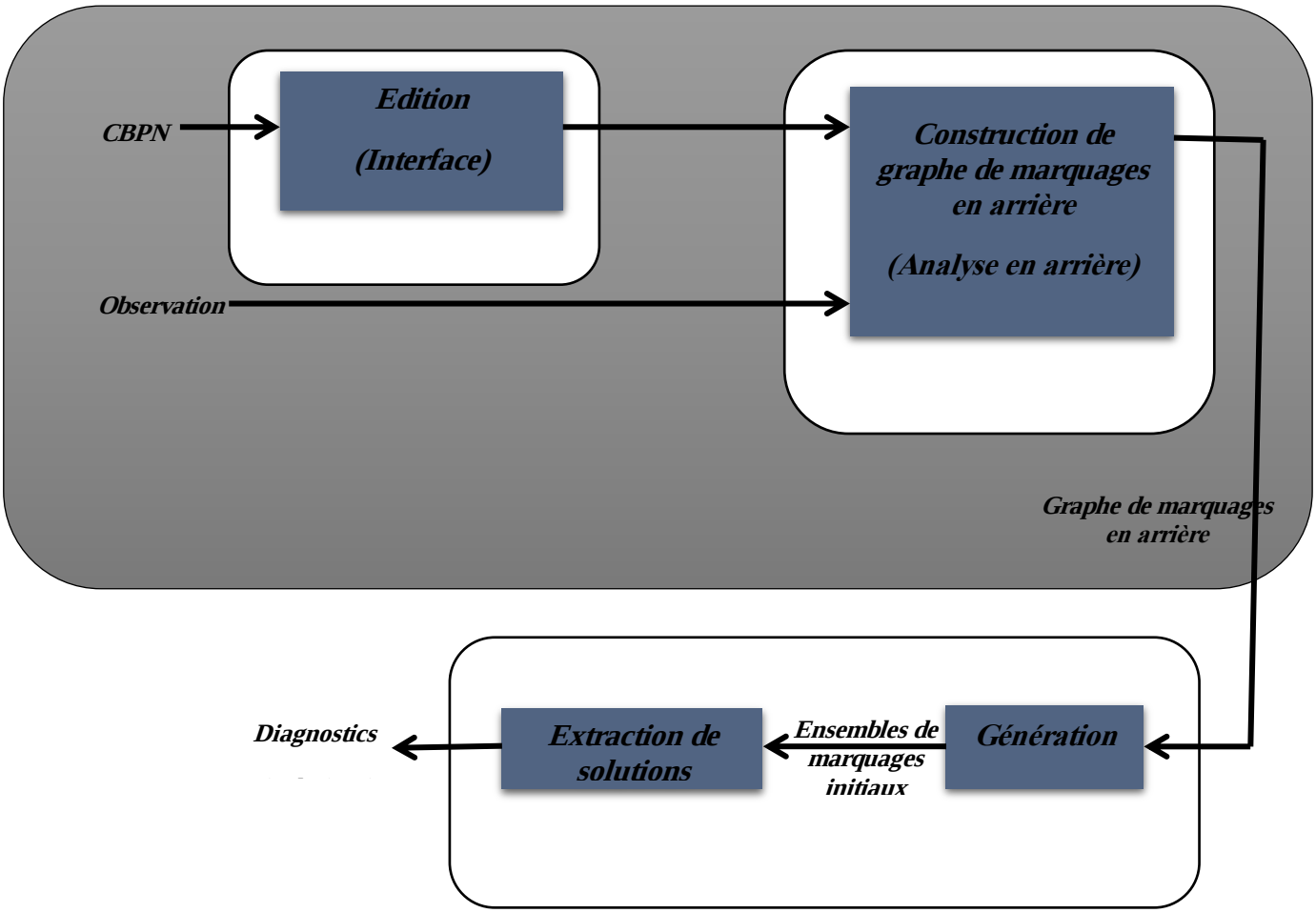


Figure 3.1 - L'architecture générale du système.

3.3.2. Conception Détaillée

La conception détaillée consiste à :

- ✓ La conception détaillée fournit pour chaque composant (module) de trois composants nécessaires (Edition, Calcule, Diagnostic) de notre système de diagnostic.
- ✓ Déterminer les structures de données.
- ✓ Déterminer les algorithmes pour définir chaque fonction logicielle.

3.3.2.1. Les Composants (Modules)

On a trois composants nécessaires **:**

Edition : le module d'édition prend en entrée le modèle du système sous forme d'un graphe CBPN représentant le comportement causal du système à diagnostiquer, et produit en sortie une représentation interne à stocker d'un tel modèle.

Calcul : Nous avons indiqué que notre outil construit le graphe de marquages en arrière sur lequel le diagnostic est accompli. Pour cette raison, le module calcul prend en entrée lemodèle CBPN généré par le module édition et l'ensemble des observations sous la forme d'un marquage final (manifestations) pour lesquelles le système est considéré défaillant et donne comme résultat le graphe de marquage en arrière de ce modèle marqué en appliquant la technique d'analyse CW décrite dans le chapitre précédent.

Diagnostic : c'est la tâche principale de notre projet. Le module de diagnostic a pour but de déterminer l'ensemble des états initiaux menant le système d'arriver à ce dysfonctionnement, en prend en entrée le graphe des marquages obtenu du module calcul. En fait, on cherche ici à extraire depuis chaque marquage initial obtenu dans le graphe construis le marquage des places dénotant des états initiaux (places sources).

3.3.2.2. Structure de données

Les structures de données utilisées est qui doivent représenter les CBPNs, les observations et le graphe de marquages sont comme suivant :

Un Réseau de Petri Coloré Causal (Causal Coloured Petri net (CBPNs)) est représenté par une matrice :

Classe matrice :

- Places c'est l'ensemble des places.
- Transitions c'est l'ensemble des transitions.
- Vale c'est la valeur de la matrice d'incidence.

Et chaque place est représentée par :

Classe place :

- Id_placereprésente l'identificateur de la place.
- Nom_placereprésente le nom de la place.
- Jeton représente le marquage actuel de la place oùchaque jeton

est une chaine de caractères.

Et pour les transitions, on a :

Classe transition :

- Id_transc'est l'identificateur de la transition.
- Nom_transreprésente le nom de la transition.
- Mat c'est une matrice $n \times m$ décrivant les combinaisons possibles decouleurs de jetons d'entrée et de sortie où n est le nombre de façons dont t peut tirer et m définit le nombre de places liées à t.

Chaque nœud (marquage) d'un graphe de marquages contient une ou plusieurs places :

Classe marquage :

- Id_marquagec'est l'identificateur de marquage.
- Nom_placereprésente la place marquée.
- Jeton représente le marquage de la place.

Les observations sont représentées par un marquage final μ^{OBS} où les places marquées sont celles appartenant aux manifestations.

3.3.2.3.Les Algorithmes

Dans cette section, on va fournir les algorithmes que nous jugeons importants :

- L'algorithme suivant (Algorithme 1) donne une méthode pour définir récursivement l'ensemble SF correspondant à s.

Algorithme 1 FB-Transition

Exiger: A transition s, a marquageμ
Assurer:l'ensemble de transition en arrièrede s
Si s = $\emptyset$**alors**
    retour{$\emptyset$}
Sinon
    Choisir t $\in$ s
    S$\leftarrow$FB-Transition (s-{t}, μ)
    $T_F \leftarrow$FB-Transition (t,μ) // transition en arrière de t àμ
    $S_{Fnew} \leftarrow \emptyset$
    Pour $t_f \in T_F$**Faire**
        Pour $s_f \in S_F$**Faire**
            $S_{Fnew} \leftarrow S_{Fnew} \cup \{\{t\ f\} \cup s\ f\}$
        Fin Pour
    Fin Pour
    retour S_{Fnew}
Fin Si

- Dans le module calcul on commence par l'algorithme 2 qui sert à calculer les places post de chaque transition

Algorithme2 : calcule post place

Entrée : matrice incidence, transition t.
Sortie : vecteur des places.

```
Début
    p← { };
    Pour chaque j allant de 1 a m-1 faire
        Si incidence [0][j].t==t alors
            Pour chaque i allant de 1 a n-1 faire
                Si incidence [0][j].Vale==1 alors
                    p← incidence [i][j].t
                Fin Si
            Fin Pour
        Fin Si
    Fin Pour
    retour p.
```

- **L'algorithme 3 :**(test) assure que toutes les places post de la transition sont marquées

Algorithme3 : test

Entrée : EnsP, liste de marquage L.
Sortie : vecteur des places.

```
1 : Début
2 :     k←0
3 :     Pour p∈EnsPFaire
4 :         Pour M ∈ L Faire
5 :             Si p.nom == m.nom Alors
6 :                 k← k+1
7 :             Fin Si
8 :         Fin Pour
9 :     Fin Pour
10 :    Si (k== taille de L) Alors
11 :        return vrai
12 :    Sinon
13 :        return vrai
```

- **L'algorithme4 :**(transition activée) revoit les transitions franchissables (activées)

Algorithme4 :transitionactivée

Entrée : matrice incidence[n][m] , liste de marquage L.
Sortie : vecteur de la transition activée.

```
1 : Debut
2 :     t← { }
3 :     p← { }
4 :     Pour l∈ L faire
5 :     Pour i allant de 1 à n-1 faire
6 :     Sil.nom== incidence [i][0].nom_placealors
7 :         Pourj allant de 1 à m-1 faire
8 :         Si incidence [i][j].vale= =1 Alors
9 :             p←calculpos(incidence,dence[i [j .t)
10 :            b←faux
11 :            Si b== vrai Alors
12 :                Pour k allant de 1 à m-1 Faire
13 :                    Sil.jeton==incidence[i][j].t.mat[k][0] Alors
14 :                        t← incidence [i][j].t
15 :                    Fin Si
16 :                Fin Pour
17 :            Fin Si
18 :        Fin Si
19 :        Fin Pour
20 :    Fin Si
21 :    Fin Pour
22 :    Fin Pour
23 :    return t
24 : Fin
```

- **L'algorithme5 :** algorithme récursif entendu pourle calcul de l'ensemble μ^{Ini} desmarquages initiaux.

Algorithme 5 CalculeMarquageInitial

Exiger: A marquage μ, L'ensembleμ^{Ini} (initial vide)
Assurer:ensembleμ^{Ini}de marquage initial.

```
Si∃ a fork transition t ∈ T,∃p, p'∈t* : μ(p) ≠μ (p') tel queμ (p) ≠∅etμ (p') ≠∅Alors
    μest incohérent
Sinon
    s←transitionactivé (μ)
    Si s = ∅Alors
        μ^Ini←μ^Ini∪ {μ}
    Sinon
        laisser s = {t1,..., tn}, n ≥ 1
        SF←FB-Transition (s,μ)
        Pour sf∈ SFFaire
            laisser sf= {(t1, i),...,(tn, j)}
            μ'←BFire (sf,μ)
            CalculeMarquageInitial(μ',μ Ini)
        Fin Pour
    Fin Si
Fin Si
```

3.4. Implémentation (Réalisation)

Après l'analyse et la conception de l'outil, nous devons passer au codage et le test.

3.4.1. Outils et langages de développement

On utilise les outils et langages de développement suivants ;

3.4.1.1 Langage de programmation Python

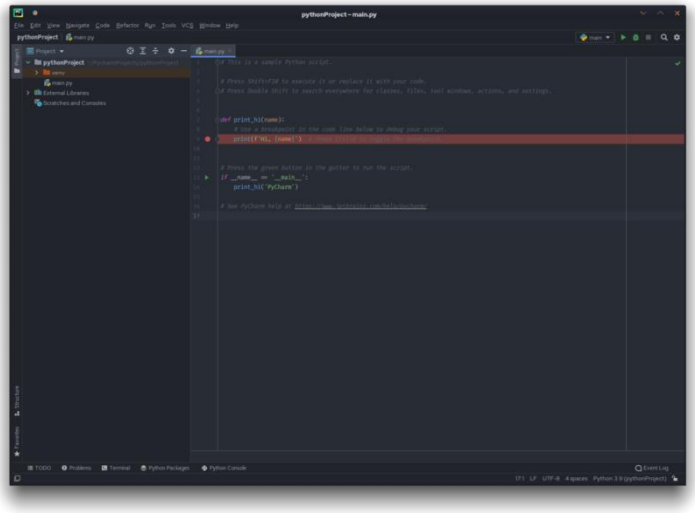
Le langage de programmation Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'unhommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de celangage a été publiée en 1991.
La Python Software Foundation 1 est l'association qui organise le développement de Python et anime la communauté dedéveloppeurs et d'utilisateurs.

Ce langage de programmation présente de nombreuses caractéristiques intéressantes :

- ✓ Il est multiplateforme. C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.
- ✓ Il est gratuit. Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !).
- ✓ C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateurpour être utilisé.
- ✓ C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à deslangages comme le C ou le C++.
- ✓ Il est orienté objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel(une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.
- ✓ Il est relativement simple à prendre en main.
- ✓ Toutes ces caractéristiques font que Python est désormais enseigné dans de nombreuses formations, depuis l'enseignementsecondaire jusqu'à l'enseignement supérieur.

3.4.1.2 Éditeur de programmation PyCharm

PyCharm est un environnement de développement intégré utilisé pour rogrammer en Python.Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.



Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux. Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache.

3.4.1.3La bibliothèqueSnakes

Snakes est une bibliothèque de réseaux de Petri à usage général pour le langage de programmation PythonSnakes peuvent créer des réseaux de Petri, les transformer (ajouter/supprimer/. . . nœuds,Ajouter enlever/. . . arcs, etc.), manipuler leurs marquages, et aussi transitions. Snakes utilise une variante très générale des réseaux de Petri colores.

Les jetons peuvent transporter des objets Python.

3.4.1.3.1. Quelque Modules de snakes :

L'ensemble de la bibliothèque se présente sous la forme d'un package Python organisé en une hiérarchie de modulesdont les principaux sont :

- ✓ snakes est le module de niveau supérieur qui dé nit les exceptions généralement utilisées
- ✓ snakes.data définit les structures de données comme les multi-sets, etc.
- ✓ snakes.typing définit un système de type utilisé pour restreindre les jetons par places.
- ✓ snakes.plugins regroupe tous les plugins fournis avec snakes.

- ✓ snakes.pnml import/export functions to/from pnml.
- ✓ snakes.nets est le module principal qui dé nit toutes les structures liées aux réseaux de Petri comme les places, les transitions, les arcs, le graphe de marquage, etc.

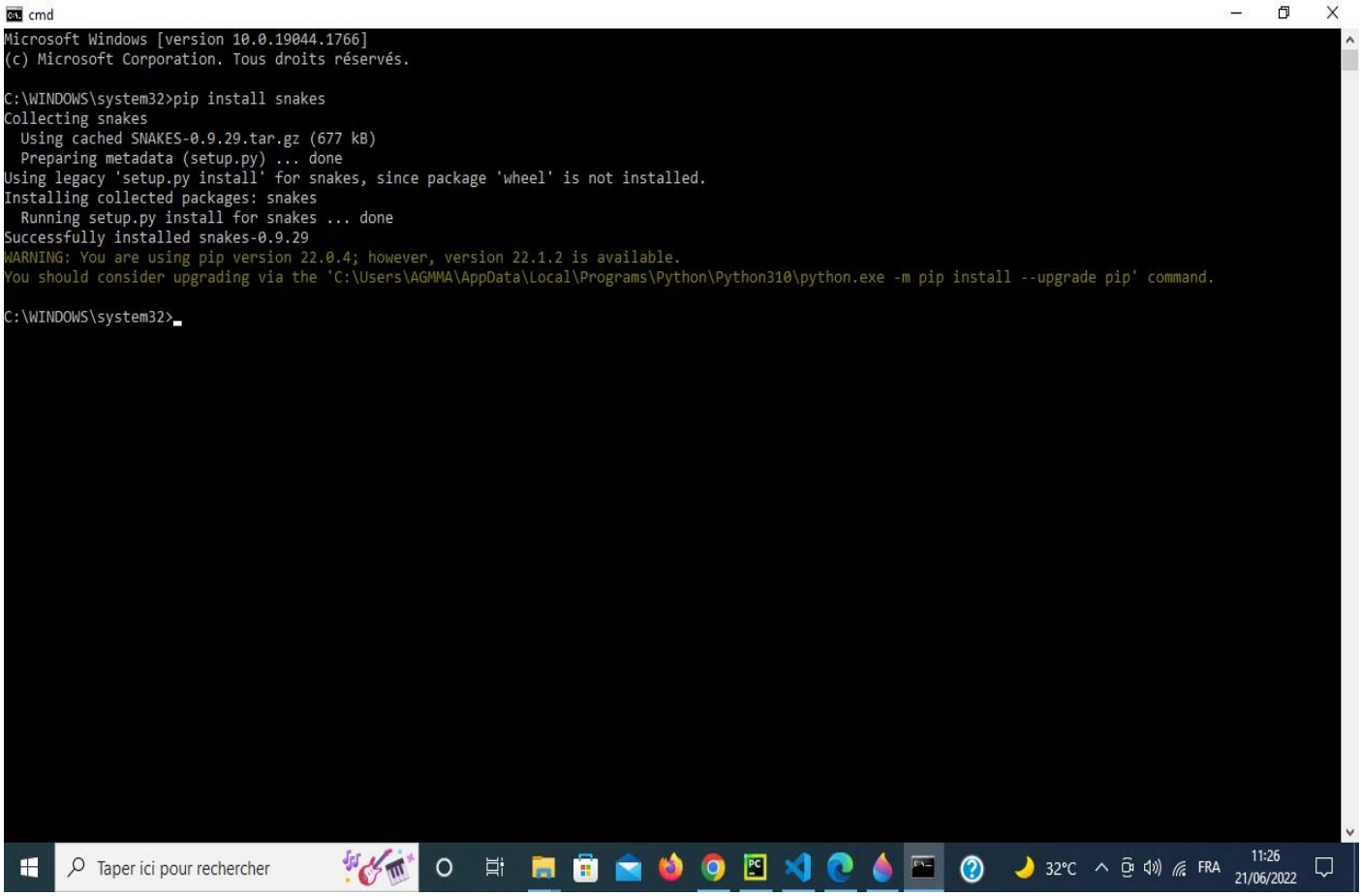


Figure 3.2. Comment ajouter la bibliothèque snakes a python.

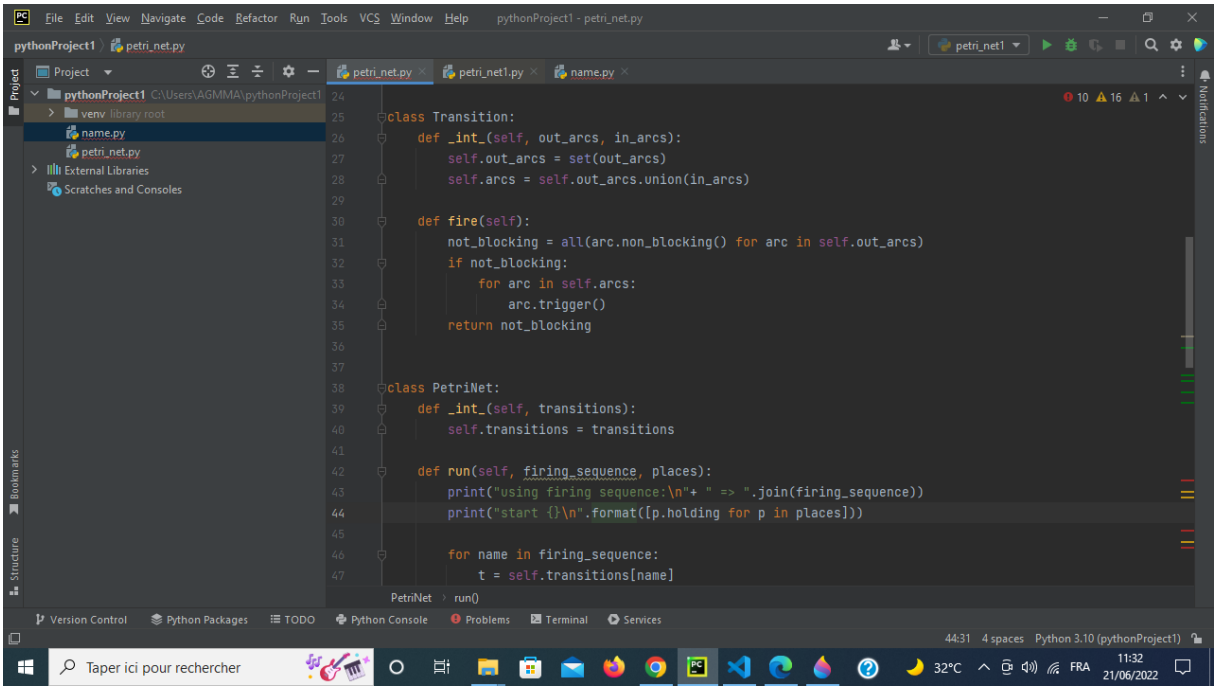


Figure 3.3. Une partie de code source de notre Application.

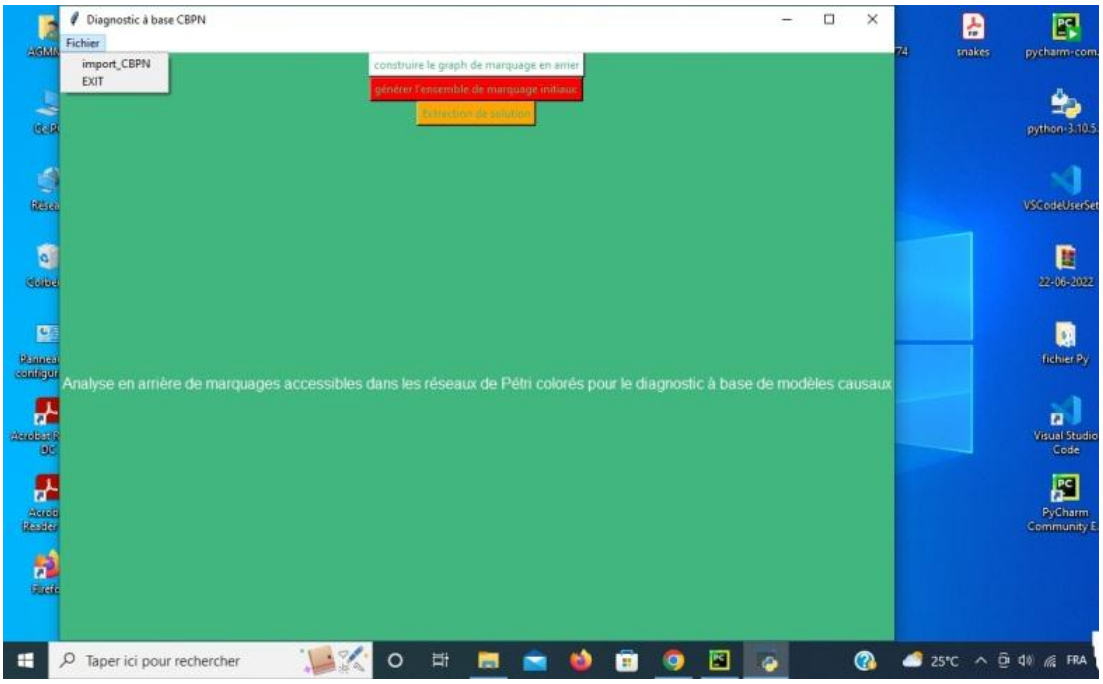


Figure 3.4.L'interface de notre Application.

3.5. Conclusion

Dans ce chapitre, nous avons montrons l'analyse des besoins et les conceptionsnécessaires comme une première étape qui nous aide pour la réalisation denotre outil et après nous avons présenté les outils que nous avons utilisés pourimplémenter notre application (Python et PyCharm), et à la fin nous avons affiché ce modèle dans une interface graphique.

Conclusion générale

Conclusion générale

Le diagnostic des pannes des systèmes est une étape décisive et tâche difficile. Dans ce document, nous avons présenté une approche basée sur les CPN pour diagnostiquer un système donné par son comportement causal. Notre objectif à été de développer un outil basé sur l'utilisation d'une classe particulière de réseaux de Petri colorés dite CBPNs. Sa caractéristique principale est l'étiquetage de transitions du modèle par des matrices représentants les différentes façons de franchissement. Comme technique de raisonnement, une analyse d'accessibilité en arrière du graphe de marquages est exploitée pour déterminer les marquages initiaux à partir desquels les solutions vont être générées.

En fait, les CBPN sont introduits principalement pour le diagnostic de dysfonctionnements du système, mais ils ne peuvent pas être limités uniquement dans ce but. Ils peuvent fournir un outil très utile pour traiter des problèmes nécessitant une analyse en arrière sur l'accessibilité des CPN pour le raisonnement.

Au niveau de notre réalisation, certaines fonctions restent incomplètes et ceci suite aux contraintes temporelles. On aimerait faire des expérimentations de l'outil développé sur des cas plus réalistes.

Bibliographie

1. S. Mancer and H. Bennoui, “Distributed Diagnostic Problem Solving With Colored Behavioral Petri Nets”. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 51, n°. 6, pp. 3380-3301, June 2021. doi: 10.1109/TSMC.2019.2924144

2. S. Mancer, A CPN Approach for Distributed Abductive Reasoning: Application to Causal Model-Based Diagnosis, Phd Thesis, Mohamed Khider University – Biskra 2020.

3. S. Mancer and H. Bennoui.”Coloured Petri Nets based diagnosis on causal Models”. In Proc. of *Petri Nets and Software Engineering* (*PNSE’2017*), pp:127-140, Zaragoza, June 2017.

4. H. Bennoui. “Interacting behavioral Petri nets analysis for distributed causal model-based diagnosis.” *Journal of Autonomous Agents and Multi Agent Systems (JAAMAS)*, vol. 28, n°. 2, pp: 155-181, Springer, March 2014. DOI : 10.1007/s10458-013-9221-5

5. Hugues Bersini, La programmation orientée objet, ÉDITIONS EYROLLES , Paris , 2009.

6. François Vernadat, Introduction aux Réseaux de Petri, INSA-DGEI - LAAS-CNRS, toulouse.

7. Jerome Delatour et Adel Benzina, UML et réseaux de Petri, LAAS-CNRS, Toulouse, 2004.

8. Patrick Fuchs et Pierre Poulain, Cours de Python, Université de Paris, France, version du 17 novembre 2021.

9. L. console and P. torasso, hypothetical reasoning in causal models, international journal of intelligent systems, vol.5, pp.83-124, 1990.

10. L. Portinale, Behavioral petri nets : a model for diagnostic knowledge representation and reasoning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 27 (1997), pp. 184–195.