

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa



Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de Master
Spécialité : Modélisation mathématique et techniques de décision

Thème

Évaluation de performances d'un système des Web services

Réalisé par :

— Sahli Ramzi

Soutenu le 06 juillet 2022 devant le Jury composé de :

	Nom et Prénom	Grade	
<i>M^{me}</i>	Bernine Nassima	MCB	Promotrice
<i>M^{me}</i>	Ziane Yasmina	MCA	Présidente
<i>M^r</i>	Djabri Rabah	MCB	Examinateur
<i>M^{me}</i>	Benouaret Zina	MCB	Examinatrice

Promotion 2021-2022

Résumé

Dans ce travail, nous avons modélisé un système de Web services simples et composites et évalué ses performances. premièrement nous avons proposé un modèle basé sur les réseaux de petri colorés avec synchronisation de deux files d'attente la première présente les Web services et la deuxième présente les demandes des clients, où les Web services et les clients sont défini par des couleurs. On a évalué ses performances à l'aide d'un simulateur "Colored Petri Nets Tools (CPN tools)", spécial pour les réseaux de petri colorés, où nous avons déterminé le nombre moyen des clients dans le système, nombre moyen des Web services et nombre moyen des client servis.

Mots clés : Web services, Évaluation des performances, Réseau de Petri, Réseau de petri coloré, simulateur CPN tools.

Abstract

In this work, we modeled a system of simple and composite Web services and evaluated the performances. first we proposed a model based on colored petri nets with synchronization of two queues the first presents the Web services and the second presents the requests of the customers, the Web services and the customers are defined by colors. Performance was evaluated using a special "Colored Petri Nets Tools (CPN tools)" simulator for colored Petri nets where we determined the average number of clients, average number of Web services and average number of clients served.

Keywords : Web service, Performances evaluation, Petri Net, Colored Petri Net.

Table des matières

Introduction générale	8
1 État de l'art des Web services	11
1.1 Introduction	12
1.2 Définition des Web services	12
1.3 L'intérêt d'un Web services	13
1.4 Caractéristiques	
des Web services	13
1.5 Architecture des Web services	15
1.5.1 Architecture de référence	15
1.5.2 Architecture étendue :	17
1.6 Les principales technologies	
des Web services :	17
1.6.1 XML (Extensible Markup Language)	18
1.6.2 SOAP (Simple Object Access Protocol)	19
1.6.3 UDDI (Universal Description Discovery and Integration)	19
1.6.4 Structure des données UDDI	20
1.6.5 WSDL (Web Service Description Language)	21
1.7 Composition des Web services	23
1.7.1 Définition composition des Web services	24
1.7.2 Cycle de vie d'une composition d'un Web service	24
1.7.3 Types de composition	25
1.8 Avantages et Inconvénients	
des Web services	26

1.8.1	Avantages des Web services	26
1.8.2	Inconvénients des Web services	26
1.9	Conclusion	26
2	Évaluation de performances	27
2.1	Introduction	28
2.2	Évaluation qualitative / quantitative	28
2.2.1	Formalisme qualitatif/ quantitatif	29
2.3	Évaluation de performances	29
2.3.1	Étapes d'évaluation de performances	29
2.4	Modélisation	30
2.4.1	Notations de Modélisation	30
2.5	Les méthodes d'évaluation	
	des performances	31
2.5.1	Méthodes Analytiques	31
2.5.2	Simulation	34
2.5.3	Prototype	36
2.6	Conclusion	36
3	Réseaux de Petri	37
3.1	Introduction	38
3.2	Notion de base	38
3.2.1	Matrice d'incidence	39
3.2.2	Notations matricielles :	40
3.3	Graphe associé	41
3.4	Rdp marqué	41
3.5	Évolution d'un RdP	43
3.5.1	Franchissement d'une transition	45
3.5.2	Conflit et parallélisme	46
3.6	Séquence de franchissement	47
3.7	Propriétés d'un réseau de Petri	49
3.7.1	Les propriétés dynamiques	49
3.7.2	Les propriétés structurelles	51
3.8	Réseaux de Petri particuliers	52
3.9	Extensions des RdP	53

3.9.1 Réseaux de Petri à temps discret	53
3.9.2 Les Rdp étendus	54
3.9.3 Réseaux de Petri hiérarchiques	56
3.9.4 Réseau de Petri coloré	56
3.10 Outils de modélisation des réseaux de Petri	57
3.11 Conclusion	57
4 Étude de l'existant sur l'évaluation des performances des Web services	59
4.1 Introduction	60
4.2 Travaux sur l'évaluation de performances des Web services	60
4.3 Étude comparative des travaux	62
4.4 Conclusion	63
5 Modélisation d'un système des Web services	64
5.1 Introduction	65
5.2 Modélisation des demandes des clients dans un système de Web services.	65
5.3 Étude du modèle du réseau	
de Petri coloré	66
5.4 Résultats	
de la simulation	69
5.4.1 Nombre moyen des client, des Web services dans le système et nombre moyen	
des clients servis	69
5.5 Position de notre travail par rapport à l'existant	70
5.6 Conclusion	71
Conclusion et perspectives	72
Annexes	78
A ANNEXE I	79
A.1 CPN TOOLS	79
A.1.1 Histoire	79
A.1.2 Définition (CPN TOOLS)	79
A.1.3 L'interface de CPN tools.	80

Table des figures

1.1 Scénario d'utilisation des Web services par les acteurs client/fournisseur	16
1.2 Architecture en Pile	18
1.3 Structure d'un message SOAP	19
1.4 structure des informations dans un fichier WSDL	22
2.1 Étapes d'évaluation de performance	30
3.1 Exemple d'un réseau de Petri	39
3.2 La structure d'un RdP	39
3.3 Exemple d'un RdP	41
3.4 Un RdP marqué	42
3.5 Transition source	43
3.6 Transition puits	43
3.7 Avant Franchissement Après Franchissement	44
3.8 Franchissement d'une transition source	45
3.9 Franchissement d'une transition puits	45
3.10 Exemple d'un réseau de Petri marqué	49
3.11 Graphe de marquage	50
3.12 Représentation d'un arc inhibiteur.	55
5.1 Modèle de réseau de Petri coloré d'un Web services	66
5.2 Le modèle proposé sur l'interface de simulateur	67
5.3 Rapport 1	68
5.4 Rapport 2	68

A.1 interface de CPN TOOLS	80
A.2 la boîte à outils d'index	80
A.3 Palettes d'outils dans l'espace travail	81
A.4 Toutes les palettes dans l'espace travail	81
A.5 Outils auxiliaires	82
A.6 Outils de création	82
A.7 Outils hiérarchie	82
A.8 Outils net	83
A.9 Outils de simulation	83
A.10 Outils d'espace d'état	83

Liste des tableaux

4.1 Étude comparative des travaux de l'existence de l'évaluation de performances des	
Web services	62
5.1 Nombre moyen des client dans le système	69
5.2 Nombre moyen des Web services	69
5.3 Nombre moyen des client servis	70
5.4 Nombre des clients servis par les Web services.	70
5.5 Étude comparative des travaux de l'existence de l'évaluation de performances des	
Web services	71

Liste des Abréviations

RdP	<i>Réseau de petri</i>
RdPC	<i>Réseau de petri coloré</i>
W3C	<i>Word Wild Web Consortium</i>
IBM	<i>International Business Macine</i>
XML	<i>Extensible Markup Language</i>
SOAP	<i>Simple Object Access Protocol</i>
UDDI	<i>Universal Description Discovery and Interation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
WDSL	<i>Web Service Sescription Language</i>
CPN	<i>Colored Petri Net</i>
CP	<i>Colored Petri</i>
RdPS	<i>Réseau de petri stochastique</i>
XHTML	<i>Extensible HyperText Markup Language</i>
SVG	<i>Scalable Vector Graphics</i>

Conception

Les applications doivent souvent récupérer les données d'une autre machine sur le réseau. Parfois, un programme s'exécutant sur une machine particulière qui doit exécuter une opération très longue sur une machine plus puissante, par conséquent, il doit y avoir un autre programme dans la configuration client serveur qui lui réponde. Il existe plusieurs technologies qui permettent de résoudre le problème de communication à travers un réseau parmi eux les Web services.

Les Web Services sont une conséquence naturelle de l'évolution de Web. Permet de faciliter l'accès aux applications entre entreprise et ainsi simplifier les échanges de données, permettre aux applications d'inter-opérer à travers un réseau indépendamment de leur plate-forme et de leur langage d'implémentation[25].

Cette technologie se base sur des protocoles et des Languages standards qui facilitent leur mise en œuvre. Tel-que le SOAP(Simple Object Access Protocole), WSDL(Web services Description Language) et UDDI(Universal Description Discovery Integration).

Pour assurer un bon fonctionnement de cette technologie et avoir ses caractéristiques on fait appel à les méthodes d'évaluation de performances (méthodes analytiques, simulation, prototype) qui s'intéressent à la valeur quantitative d'un système comme : nombre moyen d'une entité donnée, temps moyen de réponse, débit, etc).

Les réseaux de Petri sont des modèles formels abstraits pour l'analyse de systèmes concurrents. Leurs parallélismes inhérents les rendent adaptés à la représentation de systèmes d'activités concurrentes, ils ont des propriétés analytiques telles que le comportement de système modélisé peut

être analysé de manière systématique. Les réseaux de petri colorées ont été défini pour enrichir l'information portée par les jetons et facilitent la modélisation de systèmes de grande taille.

La simulation qui est l'un des outils largement utilisé dans l'évaluation de performances des systèmes. De ce fait, notre travail propose un modèle basé sur les réseaux de petri colorées (RdPC) pour évaluer les performances d'un système de Web service, le modèle constitué de deux files d'attente; la première présente les Web Services et la deuxième présente les requêtes des clients. Les performances de ce modèle sont calculées à l'aide du simulateur (colored petri net (CPN-TOOLS)).

Problématique

Dans ce travail nous nous intéressons à la modélisation, l'évaluation de performances d'un système de Web services qui est basé sur deux entités :

1. Les clients qui cherchent le Web dont ils ont besoin.
2. Les Web services qui fournissent un service.

Dans la modélisation nous avons pris en compte des arrivées des clients et des Web services, et de spécifier qu'un client peut être servi par un Web service simple ou composite. Pour faire la modélisation et d'analyser les performances d'un système de Web services nous avons proposé un modèle basé sur les réseaux de petri colorés, pour satisfaire les différents clients avec les Web services spécifiques simples ou composites.

Plan de mémoire

Ce mémoire est organisé en 5 chapitre :

- Chapitre 1 : On présente les notions de base des Web services.
- Chapitre 2 : On présente les méthodes d'évaluation de performances.
- Chapitre 3 : On présente les notions de base des réseaux de petri et quelques définitions et propriétés.
- Chapitre 4 : On cite certains travaux de l'existence sur l'évaluation des performances des Web services.

- Chapitre 5 : On Présente notre contribution qui consiste à modéliser un système des web services par les réseaux de petri colorés.

1

État de l'art des Web services

1.1 Introduction

Les Web services doivent leur origine à l'informatique distribuée et au développement du Web. L'accès aux systèmes d'information s'appuie aujourd'hui de plus en plus sur des technologies internet. Les efforts de standardisation dans ce contexte ont accentué l'engouement des personnes et des organisations, pour l'utilisation de l'internet et ont permis l'émergence des Web services comme support de développement des applications accessibles par Internet. L'apparition des Web services a permis la création et l'utilisation des application qui permette de réaliser un système d'information rapide et efficace distribué sur Internet.

1.2 Définition des Web services

On retrouve plusieurs définitions des Web services, parmi ces définitions on cite :

Les Web Services sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standards. Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un internet, par et pour des applications ou machines, sans intervention humaine, et en temps réel. [54]

Un Web Service est une application qui permet d'échanger des données avec d'autres applications web. Même si ces dernières sont construites dans des langages de programmation différents. Parmi les Web Services les plus connus on peut citer SOAP, REST ou HTTP. [53]

Citation W3C (World Wide Web Consortium) [49]

Un Web service est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les Web services peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles internet.

Citation Dico du Net [50]

Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

Citation IBM (International Business Machine) [51]

Un service Web est un ensemble de fonctions d'applications associées pouvant être appelées par programme via Internet. Les entreprises peuvent associer différents services Web de manière dynamique afin d'exécuter des transactions complexes demandant un minimum de programmation. Les services Web permettent aux acheteurs et aux fournisseurs du monde entier de se trouver, d'entrer en contact de manière dynamique, et d'exécuter des transactions en temps réel, le tout avec très peu d'interaction humaine.

Les services Web sont des applications modulaires autonomes et auto-descriptives qui peuvent être publiées, localisées et appelées à travers Internet.

1.3 L'intérêt d'un Web services

- Les Web services fournissent un lien entre applications.
- Les Web services sont normalisés, car ils utilisent les standards XML et HTTP pour transférer des données.
- les Web services ne sont pas concernés par la façon dont les messages sont produits ou consommés par des programmes. Cela permet aux vendeurs d'outils de développement, d'ouvrir différentes méthodes et interfaces de programmation au-dessus de n'importe quel langage de programmation.
- Les Web services représentent donc la façon la plus efficace de partager des méthodes et des fonctionnalités.

1.4 Caractéristiques des Web services

La technologie des Web services repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web.

Un Web service possède les caractéristiques suivantes :[34, 27]

- **Les Web services sont basés sur des standards ouverts :** Le fait que le Web est constitué de plates-formes totalement hétérogènes où les intérêts des différents acteurs du marché s'entremêlent ne l'a pas empêché de se développer et d'être universel, Ce succès est dû essentiellement à l'utilisation d'un ensemble de standards ouverts, dont le plus connu est le protocole HTTP. Les Web services suivent la même approche que le Web en se basant sur des standards ouverts. Ceci permet de réduire le coût d'intégration des applications qui est essentiellement dû, au fait que les modules interactifs ont des interfaces différentes, supportent différents protocoles de communication et différents formats de données et modèles d'interactions.
- **Les Web services sont faiblement couplés :** Un consommateur d'un Web service n'est pas directement lié à ce Web service. L'interface du Web service peut changer au fil du temps sans compromettre la capacité du client à interagir avec le service. Un système étroitement couplé implique que la logique client et serveur sont étroitement liées l'une à l'autre, ce qui implique que, si une interface change, l'autre doit être mise à jour. L'adoption d'une architecture faiblement couplée tend à rendre les systèmes logiciels plus faciles à gérer et permet une intégration plus simple entre différents systèmes.
- **Les Web services sont basés sur l'existant :** Les Web services ne constituent pas un changement fondamental pour remplacer l'infrastructure existante. Ils offrent un excellent moyen qui permet aux systèmes existants d'interagir et d'évoluer avec de nouvelles applications. Face aux évolutions du marché, les entreprises sont devenues plus agiles et peuvent s'adapter rapidement aux nouveaux besoins tout en tirant parti des systèmes existants.
- **Les Web services basés sur XML :** XML constitue la technologie de base des architectures, les Web service utilisent XML au niveau des couches de représentation et de transport des données.
- **Accessibilité universelle :** Les Web services peuvent être définis, décrits et découverts sur le Web, ce qui contribue à leur accessibilité. Non seulement les utilisateurs de Web services peuvent trouver des services appropriés, mais les Web services peuvent se décrire et se publier afin qu'ils puissent se lier et interagir les uns avec les autres.

- **Auto-contenu et auto-descriptif** : Un Web service contient dans sa description toutes les informations nécessaires à l'utilisation de ses applications.
- **Gros Grains** : La technologie des Web services offre un moyen naturel de définir des services à granularité grossière qui accèdent à la bonne quantité de logique métier.
- **Possibilité d'être synchrone ou asynchrone** : La synchronicité fait référence à la liaison du client à l'exécution du service. Dans les appels synchrones, le client bloque et attend que le service termine son opération avant de continuer. Les opérations asynchrones permettent à un client d'appeler un service, puis d'exécuter d'autres fonctions.

Les clients asynchrones récupèrent leur résultat ultérieurement, tandis que les clients synchrones reçoivent leur résultat lorsque le service est terminé. La capacité asynchrone est un facteur clé pour permettre des systèmes faiblement couplés.

clients asynchrones récupèrent leur résultat ultérieurement, tandis que les clients synchrones reçoivent leur résultat lorsque le service est terminé. La capacité asynchrone est un facteur clé pour activer les systèmes à couplage lâche.

- **Prend en charge l'échange de documents** : L'un des principaux avantages de *XML* est sa façon générique de représenter non seulement les données, mais aussi les documents complexes. Ces documents peuvent être aussi simples que représenter une adresse actuelle, ou ils peuvent être aussi complexes que représenter un livre entier ou une demande de devis (RFQ). Les Web services prennent en charge l'échange transparent de documents pour faciliter l'intégration de l'entreprise.

1.5 Architecture des Web services

1.5.1 Architecture de référence

Pour faciliter l'interopérabilité et l'extensibilité du paradigme des Web services, une architecture de base (référence) est nécessaire afin de préserver les objectifs initiaux visés par les Web services lors des évolutions technologiques successives.

Fournisseur de service (provider) :

Le fournisseur correspond au propriétaire du service, sa tâche consiste à implémenter les fonctions du Web services, décrire l'interface de ces fonctions en utilisant une manière standard puis publier l'interface dans un annuaire de service, pour permettre aux consommateurs de découvrir les services.

Le Client (requester) :

correspond au demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un Web service.

L'annuaire des services (Service registry) :

correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités la recherche des services.

La FIG 1.1 [49] montre comment Cette architecture fonctionne :

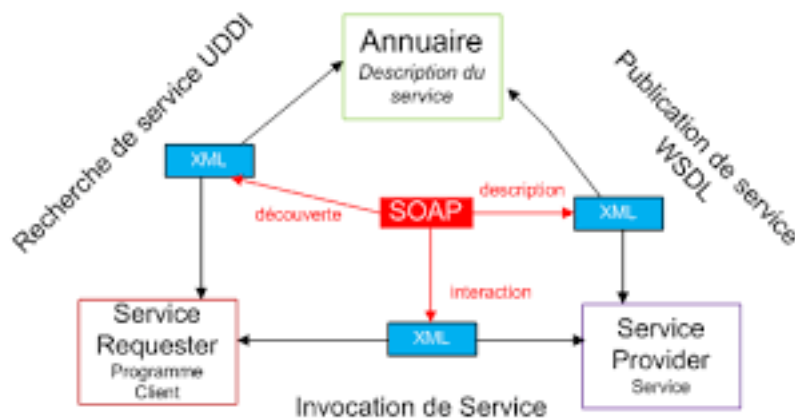


FIGURE 1.1: Scénario d'utilisation des Web services par les acteurs client/fournisseur

La dynamique de l'architecture est en général la suivante :

1. Le fournisseur de service publie ses Web services sur l'annuaire.
2. Le client cherche le Web service dont il a besoin à l'aide de l'annuaire.

3. L'annuaire cherche pour le client, trouve le Web service approprié et renvoie une réponse au client.
4. Le client envoie une deuxième requête au serveur pour obtenir le contrat du service.
5. Le fournisseur envoie sa réponse sous la forme établie par WSDL en langage XML.
6. Le client appelle le Web service selon le contrat en envoyant un document XML représentant sa demande.
7. Le Web service retourne le résultat de l'appel.

1.5.2 Architecture étendue :

Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. Les fonctions de couches supérieures reposent sur celles de couches inférieures.

L'infrastructure de base des Web services seule, ne répond pas à tout les problèmes d'intégration. Elle est complétée par d'autres couches :

La couche Business processus : permet d'utiliser les Web services dans le domaine du e-business d'une manière effective. Elle représente un business processus comme un ensemble de Web services.

les couches transversales : contiennent un ensemble de standards qui rendent viable l'utilisation effective des Web services dans le monde industriel (sécurité, administration, transactions et qualité de services (QoS)).

La couche Transport : Assure la connectivité physique.

La FIG 1.2 [52] contient les trois couches.

1.6 Les principales technologies

des Web services :

Le terme technologies de Web services désigne un ensemble de technologies basées sur des standards ouverts (non propriétaires), essentielles pour le transport et la transformation des données d'un client vers un service d'une façon standard. Les plus courants sont : XML, SOAP, WSDL,

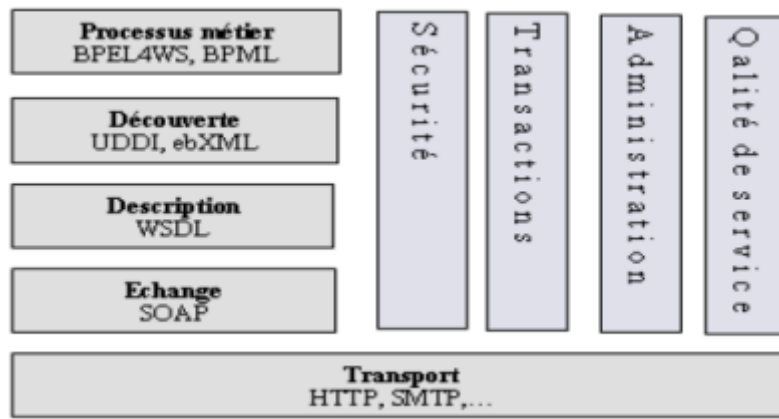


FIGURE 1.2: Architecture en Pile

UDDI.

1.6.1 XML (Extensible Markup Language)

XML (langage de balisage extensible) est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup Language (SGML (ISO 8879)) [35]. est un format texte simple pour représenter des informations structurées : documents, données, configuration, livres, transactions, factures. Sa syntaxe est dite « extensible », car elle permet de définir pour chaque langage son vocabulaire et sa grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte enrichi, etc.) entre systèmes d'informations hétérogènes (interopérabilité). [36]

La norme XML comporte deux parties : XML à proprement parler et les DTD (Document Type Définition) qui définissent les balises qui sont utilisées dans une famille de documents XML.

XML a été conçu pour des documents arbitrairement complexes, tout en s'appuyant sur cinq grands principes : [39]

- La lisibilité à la fois par les machines et par les utilisateurs ;
- La définition sans ambiguïté du contenu d'un document ;
- La définition sans ambiguïté de la structure d'un document ;
- La séparation entre documents et relations entre documents ;
- La séparation entre structure du document et présentation du document.

1.6.2 SOAP (Simple Object Access Protocol)

SOAP est un protocole de la famille XML servant à l'échange d'informations dans un environnement distribué et décentralisé. Il est considéré comme la technologie la plus importante des Web Services. Le standard SOAP a été proposé au W3C par Microsoft, IBM, Lotus, DevelopMentor et Userland, Ce protocole peut appeler des méthodes RPC (Remote Procedure Call) et envoyer des messages à des machines distantes via HTTP, à l'aide de SOAP les applications peuvent converser entre elles et échanger des données sur Internet, quelles que soient les plates-formes sur lesquelles elles s'exécutent.[37]

Un message SOAP est composé de deux parties obligatoires : l'enveloppe SOAP et le corps SOAP; et une partie optionnelle, l'en-tête SOAP :[37, 38]

- **L'enveloppe SOAP (<Envelope>)** : est l'élément racine de chaque message SOAP, il contient deux éléments enfants, un élément facultatif <Header> et un élément obligatoire <Body>.
- **L'en-tête SOAP (l'élément <Header>)** : est un sous-élément facultatif de l'enveloppe SOAP, il est utilisé pour transmettre les informations relatives à l'application qui sont traitées par les nœuds SOAP le long du flux de messages.
- **Le corps SOAP (l'élément <Body>)** : est un sous-élément obligatoire de l'enveloppe SOAP, qui contient des informations destinées au destinataire final du message.

La FIG 1.3[49] décrit la structure d'un message SOAP :

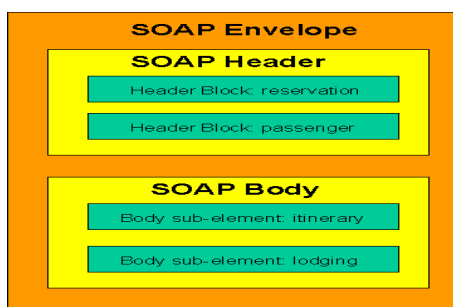


FIGURE 1.3: Structure d'un message SOAP

1.6.3 UDDI (Universal Description Discovery and Integration)

La spécification UDDI (Universal Description, Discovery and Integration) définit un moyen de publier et de découvrir des informations sur les Web services.

UDDI a deux fonction :

- Il s'agit d'un protocole basé sur SOAP qui définit la façon dont les clients communiquent avec les registres UDDI.
- Il s'agit d'un ensemble particulier de registres répliqués mondiaux

Registres UDDI

UDDI gère la découverte des Web services en s'appuyant sur un registre distribué d'entreprises et leurs descriptions de services implémentées dans un format XML commun. se présentent sous deux formes : publique et privée.

Un registre public : Un registre public est un ensemble d'annuaires homologues qui contiennent des informations sur les entreprises et les services.

Un registre privé : Un registre privé vous permet de publier et de tester vos applications internes dans un environnement sécurisé et privé.

Les entreprises peuvent enregistrer trois types d'informations dans un registre UDDI. Ces informations sont contenues dans trois éléments de l'UDDI. Ces trois éléments sont [49] :

- Pages blanches (white pages) : contiennent des informations sur le fournisseur de services, le type d'entreprise, les informations de contact et la catégorisation.
- Pages jaunes (yellow pages) : Sont associées aux Web services fournis par une entreprise, ils contiennent une description indirecte de ce que fait le service.
- Pages vertes (green pages) : Contiennent des informations techniques sur un Web services.

1.6.4 Structure des données UDDI

Le registre UDDI se compose de quatre types de structures de données [49] :

- **BusinessEntity (entité d'affaires)** : Elles décrivent les organisations ayant publié des services dans le répertoire. On y trouve notamment le nom de l'organisation, ses adresses (physiques et Web), des éléments de classification, une liste de contacts ainsi que d'autres informations.
- **BusinessService (service d'affaires)** : Elles décrivent de manière non technique les services proposés par les différentes organisations. On y trouve essentiellement le nom et la descrip-

tion textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs « `bindingTemplate` ».

- **BindingTemplate (modèle de rattachement)** : permet de décrire des Web services utilisant HTTP, mais également des services invoqués par d'autres moyens. Les « `bindingTemplates` » donnent les coordonnées des services. Ils contiennent notamment une description, la définition du point d'accès (une URL) et les éventuels « `tModels` » associés.
- **tModel (index)** : Sont les descriptions techniques des services. UDDI n'impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels (XHTML, par exemple). C'est à ce niveau, que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services.

L'interface UDDI

L'interface UDDI est définie sous forme de documents UDDI et implémentée sous forme de SOAP. Elle est composée des modules suivants [49] :

- **Interrogation inquiry** : permet de rechercher des informations dans un répertoire UDDI et de lire les différents enregistrements suivant le modèle de données UDDI.
- **Publication** : permet de publier des informations dans un répertoire UDDI, conformément à son modèle de données.
- **Sécurité** : elle est utilisée pour obtenir et révoquer les jetons d'authentification nécessaires pour accéder aux enregistrements protégés dans un annuaire UDDI.
- **Contrôle d'accès et propriété** : permet de transférer la propriété d'informations (qui est à l'origine attribuée à l'utilisateur ayant publié ces informations) et de gérer les droits d'accès associés.
- **Abonnement** : permet à un client de s'abonner à un ensemble d'informations et d'être averti lors des modifications de ces informations.

1.6.5 WSDL (Web Service Description Language)

WSDL (Web Service Description Language) est un langage basé sur XML. Les fichiers WSDL contiennent la description de l'accès à des Web services, utilisé pour décrire les fonctionnalités offertes par un Web service. Le WSDL définit les éléments suivants : [40]

- L'emplacement d'accès du Web service;
- Opérations exposées par le Web service;
- Paramètres acceptés par les opérations exposées;
- Tout message de demande ou de réponse associé aux opérations.

La puissance de WSDL découle de deux architectures principales : la capacité de décrire un ensemble d'opérations commerciales et la capacité de séparer la description en deux unités de base. Ces unités sont la description abstraite et la description concrète.

La figure 1.5 illustre la structure d'un fichier WSDL

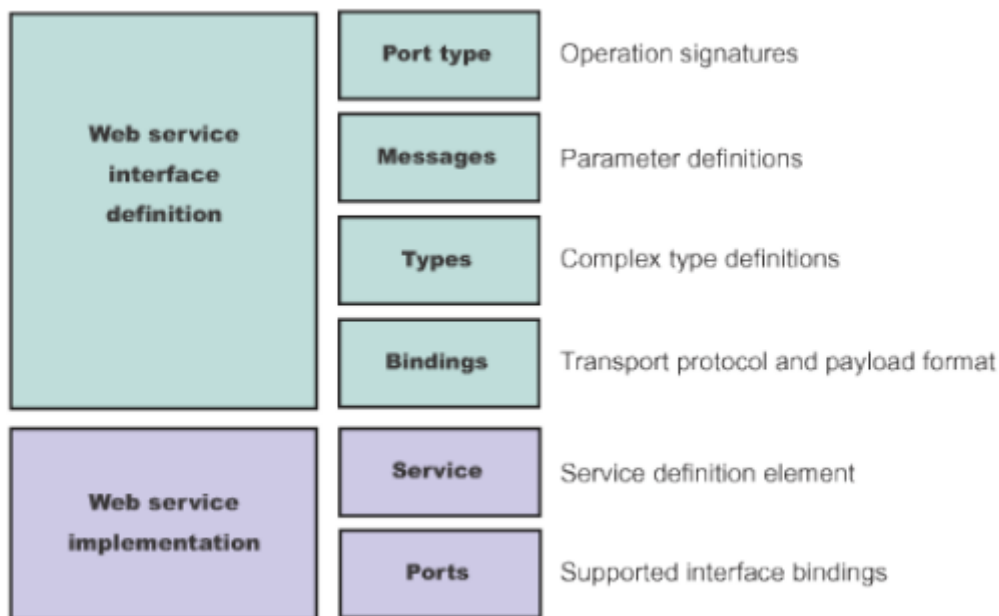


FIGURE 1.4: structure des informations dans un fichier WSDL

Description abstraite[40]

définir les éléments de l'interface de service tel que les types de données, les messages, les types de ports. Ces parties décrivent des informations abstraites indépendantes au contexte de mise en œuvre.

- **Type** : L'élément types d'un fichier WSDL contient les définitions de type de données utilisées par les messages traités par le Web service. Ces définitions utilisent XML pour organiser les informations pertinentes pour l'élément type en cours de définition.

- **Message** : Les éléments de message d'un fichier WSDL définissent les paramètres d'entrée ou de sortie pour les opérations disponibles dans le Web service. Chaque message peut être constitué d'une ou plusieurs parties.
- **PortType** : L'élément portType d'un fichier WSDL définit l'interface réelle du Web service. Un type de port est simplement un groupe d'opérations connexes et est comparable à une bibliothèque de fonctions, un module ou une classe dans un langage de programmation traditionnel. Chaque opération a un nom et contient au plus un message d'entrée et un message de sortie qui sont défini par les éléments "INPUT" (resp "OUTPUT").
Le message "FAULT" est présent et définit un message d'erreur retourné.
- **Import** : Utilisé pour importer d'autres fichiers WSDL.
- **Document** : Permet de décrire des commentaires.

Description concrète

Consiste à définir les informations relatives à l'endroit où est publié le service (description de l'implémentation du service) en utilisant les éléments Liaison, Port et Service.

- **Liaison (Binding)** : L'élément de liaison d'un fichier WSDL lie l'interface définie par le type de port aux protocoles de transport et de messagerie. ‘
- **Port** : Représente un point d'accès de services défini par une adresse réseau par un fournisseur et une ou plusieurs liaisons(Binding) aux protocoles de transport pour un Port Type donné.
- **Service** : Regroupe un ensemble de ports relié entre eux.

1.7 Composition des Web services

Dans les dernières années, l'utilisation de Web service a connu une popularité grandissante. Ces services sont très utilisés notamment par les entreprises pour rendre accessible leurs métiers ou leurs données via le Web. Les Web services, tels qu'ils sont présentés, sont conceptuellement limités à des fonctionnalités relativement simples qui sont modélisées par une collection d'opérations.[41]

1.7.1 Définition composition des Web services

Des différentes définitions ont été proposées pour le concept de composition, nous citons dans ce qui suit les plus communes :

La composition des Web services c'est la capacité d'offrir des services à valeur ajoutée en combinant des services existants probablement offerts par différentes organisations.[42]

La composition des Web services peut être définie comme le processus de sélection, de combinaison et d'exécution de services en vue d'accomplir un objectif donné .

Georges Gardarin : a définit la composition des Web services comme une technique qui permet d'assembler des Web services afin d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôle (boucle, test, traitement d'exception, etc.) et d'échange.[43]

1.7.2 Cycle de vie d'une composition d'un Web service

La composition des Web services peut être vue comme étant un moyen efficace pour créer, exécuter, et maintenir des services qui dépendent d'autres. **Benatallah et al** dans [45] Ont défini un cycle de vie d'une composition de Web services englobant six activités :

1. **Encapsulation de services natifs (Wrapping services)** : Elle garantit que tout service peut être appelé lors d'une composition, (que soit le modèle de ses données, le protocole d'interaction,ect.).
2. **Établissement d'accord d'externalisation (Setting outsourcing agreements)** : Consiste à négocier et établir les obligations contractuelles entre les services partenaires.
3. **Assemblage de services composants (Assembling composite services)** : Cette tâche permet de décrire, les services à composer de manière abstrait par l'identification des services réalisant une composition donnée, tout en spécifiant leurs interactions à un haut niveau d'abstraction.
4. **Exécution de services composants(Executing services)** : Permet d'exécuter les spécifications de la composition.
5. **Le contrôle de l'exécution de services composites (Monitoring services)** : Permet la vérification de plusieurs aspects et propriétés, par exemple : elle contrôle l'accès aux services, vérifie les changements de statut et les échanges de messages. Ceci permet la détection des

erreurs (violations de contrats), de mesurer les performances des services appelés et de prédire des exceptions.

6. **Évolutivité des services (Evolving service)** : Assure l'évolution de la composition, (c.à.d. la modification des services invoqués, utilisation de nouveaux services, prise en compte des retours de l'activité de contrôle).

1.7.3 Types de composition

La composition peut être classifiée en cinq catégories :

1. **Composition manuelle** : C'est à l'utilisateur de programmer et d'implémenter la composition.
2. **Composition semi-automatique** : Elle fournit des suggestions sémantiques pour aider à la sélection des Web services dans le processus de composition.
3. **Composition automatique** : prend en charge tout le processus de composition sans qu'aucune intervention de l'utilisateur ne soit requise. Cependant, la composition automatique a été toujours une tâche difficile à réaliser.
4. **Composition statique (offline)** : Une telle composition est adaptée pour les environnements fermés où les composants n'évoluent pas souvent. Ce type de composition engendre des applications peu flexibles, parfois inappropriées avec les exigences des clients. Microsoft Biztalk est l'un des moteurs de composition statiques de Web services.
5. **Composition dynamique (on-line)** : Elle permet de créer de manière autonome des services complexes en combinant des composants à la volée en fonction des demandes de l'utilisateur et du contexte.[44] Elle évolue dans des environnements flexibles et ouverts, où la sélection et la combinaison des composants sont effectuées à la demande.

1.8 Avantages et Inconvénients des Web services

1.8.1 Avantages des Web services

- Les Web services permettent des programmes écrits dans différentes langues et sur des plateformes différentes de communiquer entre elles-mêmes
- Les Web services utilisent des normes et des protocoles ouverts tel que SOAP et HTTP.
- Les outils de développement, basés sur ces standards, permettent la création automatique de programmes utilisant les Web services existants.
- Les services sont conçus de façon à ce qu'ils puissent être réutilisés ultérieurement pour minimiser la redondance et rendre le service réutilisable par les différentes applications du système d'information.
- Les coûts réduits grâce aux Web services.

1.8.2 Inconvénients des Web services

- Le manque de la sémantique.
- Le langage de la description (WSDL) est classique.

1.9 Conclusion

Un service Web met à disposition un service via Internet. Il constitue ainsi une interface permettant à deux applications de communiquer. Pour y parvenir, la technologie doit disposer de deux propriétés essentielles être multiplateforme et être partagée. Dans ce chapitre nous avons établi une étude de l'état de l'art des Web services. Dans le chapitre qui suit nous allons définir les méthodes de l'évaluation de performances qui assure un bon fonctionnement de la technologie.

2

Évaluation de performances

2.1 Introduction

L'évolution permanente des systèmes et réseaux informatiques et des télécommunications souligne le besoin croissant d'outils facilitant l'étude de leur comportement. Il est nécessaire d'avoir une assistance dans les phases de conception (comparaison des solutions), de développement (dimensionnement) et d'utilisation (gestion des flux et de la QoS : Quality of Service) des systèmes et réseaux [20]. En effet, le développement d'un système complexe demande non seulement une modélisation qualitative pour vérifier sa correction logique, mais aussi une validation a priori des performances du système lors de la phase de conception. En plus, lorsque ces systèmes possèdent des contraintes temporelles (applications temps réel), nous devons inclure parmi les éléments à retenir des considérations de performances qui ne peuvent être abordées avec rigueur que grâce à l'utilisation de techniques quantitatives pour l'évaluation de performances.

2.2 Évaluation qualitative / quantitative

Il existe deux approches d'évaluation pour un système : [19]

1. L'approche qualitative appelé l'évaluation qualitative s'intéresse à définir des propriétés structurelles et comportementales :
 - Absence de blocage;
 - Disponibilité;
 - Gestion de la concurrence.
2. L'approche quantitative appelé l'évaluation quantitative consiste à calculer les critères de performances du système :
 - Débit : exprime le rapport entre la quantité d'information (par second) et la quantité totale d'information (données utilisateur, données de contrôle et données retransmises en cas d'erreurs) véhiculée par réseau.
 - Temps de réponse : est la durée qui s'écoule après une variation, permettant de contrôler la réaction d'un système.
 - Le nombre moyen de clients dans le système : est égale au taux d'arrivées des clients multiplié par le temps moyen d'attente d'un client.

- La durée moyen de séjour dans le système qui est composée de la durée moyen d'attente et la durée moyen de service.
- La durée moyen d'une période d'activité : c'est-à-dire l'intervalle de temps pendant lequel il y a toujours au moins un client dans le système.

2.2.1 Formalisme qualitatif/ quantitatif

- L'analyse quantitative est essentiellement réalisée à l'aide de files d'attente.
- L'analyse qualitative fait appel aux chaîne de markov et Réseaux de Pétri.[19]

2.3 Évaluation de performances

Définition 2.1 (évaluation de performances)

L'évaluation de performance s'intéresse au calcul des indices de performances d'un système[46]. Ces derniers sont représentés sous forme de valeurs quantitatives, comme le débit, le temps d'attente, le temps de réponse, ...[47]

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

- **Un système** : C'est l'entité dont on évalue les performances. Un système est un ensemble de ressources partagés entre les différentes tâches.
- **Une Charge** : Représente généralement le trafic en entrée, qui pourrait être l'ensemble des messages servis par un dispositif du réseau, le nombre de tâches à exécuter par un processeur.

2.3.1 Étapes d'évaluation de performances

La démarche à suivre pour l'évaluation de performances peut être schématisée par la FIG 2.1 :

- Définir clairement le but de l'étude et délimiter le système à étudier.
- Lister tous les services offerts par le système et les issues possibles par service.
- Sélectionner les critères de performances.
- Lister les paramètres du système et de la charge.
- Sélectionner les techniques d'évaluation.

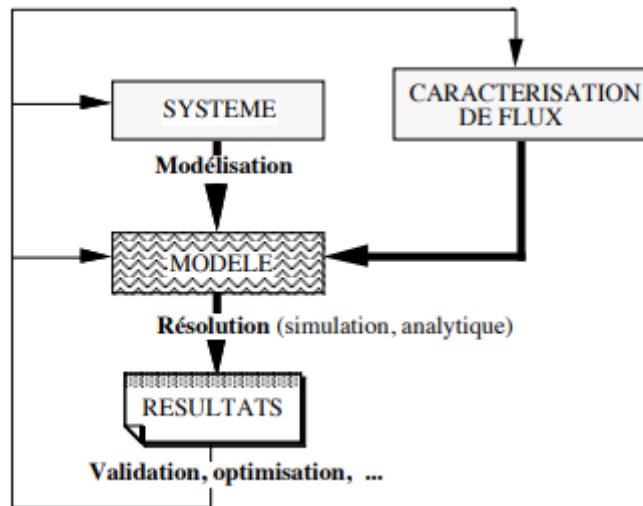


FIGURE 2.1: Étapes d'évaluation de performance

- Sélectionner les scénarios de charge.
- Définir les plans d'expérience.
- Analyser et interpréter les résultats.
- Présenter les résultats.

2.4 Modélisation

2.4.1 Notations de Modélisation

Modélisation, Modéliser et modèle Est un processus qui permet de traduire un phénomène réel en un modèle afin de lui appliquer des outils mathématique, dans le but de décrire et prévoir le comportement de ce phénomène. Modéliser un phénomène réels revient à représenter en un problème mathématique ce qui était exprimé en langage courant auparavant, en utilisant des outils mathématiques. le modèle est la mise du système en expressions mathématiques, logique ou symboliques entre les entités. Ces modèles sont dits analytiques.[18]

on cite quelque différent modèle :

- **Modèle Statique** : les modèles statique sont à l'équilibre ou en équilibre en fonction du temps.
- **Modèle dynamique** : changent constamment en fonction du temps.

- **Déterministes** : traduisant des relations de causes à effets entre les variables caractérisant un phénomène donné (dépendent de facteurs connus).
- **Stochastiques** : représente une évolution discrète ou à temps continu, d'une variable aléatoire .
- **Modèle Discrets** : le modèle discret peut être utilisé pour décrire l'évolution d'un système continu.
- **Modèle Continus** : le modèle continu peut être utilisé pour décrire l'évolution d'un système continu.

2.5 Les méthodes d'évaluation des performances

2.5.1 Méthodes Analytiques

- Modéliser le système grâce à un formalisme mathématique (file d'attente, réseaux de petri, ...).
- Calculer les indices de performances par des formules analytiques (sont peu coûteuses en temps) ou des méthodes numériques (ont une complexité numérique élevée).
- Peut être utilisée pour valider les résultats des simulation.
- Peut donner des résultats exacts rapides.

1. Files d'attente

Définition 2.2 : Basé sur deux entités : client et serveur (un client arrive dans une file, attend un service, reçoit le service, puis se dirige vers une autre file ou quitte le système), ce formalisme est largement utilisé pour modéliser des systèmes dans les domaines d'activité les plus divers :

- * la vie quotidienne (acheter un ticket de cinéma au guichet, passer la caisse de supermarché, etc.)
- * l'informatique (une tâche attend dans la RAM pour être traitée par la CPU),
- * dans les télécommunications (un message dans la mémoire tampon de la carte réseau en attendant que le médium de transmission soit libre, dans un commutateur en attendant d'être acheminé vers une bonne ligne de sortie).

- * le transport (voiture devant un péage, un feu tricolore)
- * la gestion de production dans une usine, etc... .

Caractéristiques d'un système de files d'attente

La théorie des systèmes d'attente a comme objectif d'en étudier les structures et de calculer les valeurs caractéristiques permettant de décrire les performances d'un tel système à partir de la distribution stationnaire du processus $\{X(t); t \geq 0\}$;

Notation de kendall

Une file d'attente est notée $A/B/m/K/N/Z$ selon Kendall avec : [20]

- A : Processus d' arrivée des clients (distribution d'inter arrivée).
- B : Schéma de service (distribution de durée de service de clients).
- m : nombre de serveurs.
- K : capacité maximale de la file d'attente.
- N : Capacité du système.
- Z : discipline du système qui décrit la façon dont les clients sont ordonnancés.

Pour les arrivées et les services on utilise les symboles suivants :

- M : loi exponentielle.
- D : loi déterministe.
- G : loi générale.
- Hk : loi hyper exponentielle d'ordre k .
- Ek : loi d'Erlang d'ordre k .

Le nombre de serveurs peut varier de 1 à l'infini (noté ∞), de même pour K et N .

En ce qui concerne Z , les ordonnancements les plus utilisés sont :

- $FIFO$ (First In, First Out).
- $LIFO$ (Last In, First Out).
- $RANDOM$: le prochain client à servir est choisi aléatoirement.
- $Round - Robin$ (cyclique avec un quantum Q).

- *PS* (Processor Sharing) : Round-Robin avec Q tend vers 0. Tous les n clients servis en même temps avec un taux μ/n .
- *PRIOR* (Avec priorité, avec préemption ou sans préemption)

Modélisation par les système de fils d'attente

Les systèmes des files d'attente servent à analyser les performances d'un système réel modélisé. Cette analyse peut être menée en étudiant le comportement du système selon deux axes :

- Étude en régime transitoire : l'étude du régime transitoire permet de répondre à des questions de performances qui sont liées à des instants données ou sur des périodes de courts terme.
- Étude en régime stationnaire (permanent) : consiste à vérifier si le système tend vers un équilibre (en terme de probabilité) lorsque le temps croît (à long terme).

Analyse mathématique d'un système de files d'attente

L'étude mathématique d'un système de files d'attente se fait généralement par l'introduction d'un processus stochastique, défini de façon appropriée. On s'intéresse principalement au nombre de clients $X(t)$, se trouvant dans le système à l'instant $t(t \geq 0)$. En fonction des quantités qui définissent le système, on cherche à déterminer :

- Les probabilités d'état $P_n(t) = P(X(t) = n)$, qui définissent le régime transitoire du processus stochastique $\{X(t), t \geq 0\}$. Il est évident que les fonctions $P_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.
- Le régime stationnaire du processus stochastique est défini par :

$$\pi_n(t) = \lim_{t \rightarrow +\infty} P_n(t) = P(X(+\infty) = n) = P(X = n) \quad (n = 0, 1, 2, \dots)$$

où $\pi_n \geq 0$ est appelée distribution stationnaire du processus $X(t), t \geq 0$.

2. Réseaux de petri

Réseaux de petri sont utilisés pour modéliser le comportement dynamique des systèmes à événements discrets. Ils permettent de modéliser les relations de précédentes et les interactions structurelles d'événements stochastiques, concurrents et asynchrones. De plus, leur

nature graphique permet une meilleure visualisation des systèmes complexes. Ils sont de nos jours largement utilisés dans la modélisation, l'évaluation des performances. Ils représentent un outil de modélisation hiérarchisé, avec un support mathématique bien développé, ils permettent d'entreprendre l'analyse et la conception de système complexe. l'existence de plusieurs extensions de réseaux de petri tel que : réseaux de petri temporisé, réseaux de petri colorée,... permet une analyse quantitative et qualitative du système étudié.

3. Chaîne de Markov

Sont utilisées pour modéliser des systèmes comme un ensemble d'états où les taux de transition entre les états sont connus.

2.5.2 Simulation

La simulation est l'un des outils d'aide à la décision les plus efficaces à la disposition des concepteurs et des gestionnaires des systèmes complexes, une technique de modélisation largement utilisée dans l'évaluation de performances des systèmes informatiques et réseaux de communications. Elles consistent à construire un modèle d'un système réel et à conduire des expériences sur ce modèle afin de comprendre le comportement de ce système et de calculer les performances. Cela permet de modéliser des situations très complexes que l'on ne peut résoudre analytiquement.

Les étapes de développement d'un modèle de simulation [56, 20]

- (a) **L'identification du problème :** identifier et analyser le problème, spécification des objectifs.
- (b) **la simulation est-elle appropriée à la résolution du problème :** il faut poser certaines questions :
Comment le problème pourrait être résolu sans la simulation?
comment d'autres ont-ils déjà résolu ce type de problème?
- (c) **La formulation du problème :** ceci a pour but d'identifier toutes les entrées dont on a besoin pour le modèle et d'y associer les sorties pour pouvoir collecter les données.

- (d) **Collecte et entrée des données :** La collecte des données se base sur les entrées et les sorties.
- (e) **Formulation du modèle mathématique :**
On se concentre sur la procédure qui relie l'entrée à la sortie.
On identifie ensuite les principales sous-routines et leurs relations.
On détaille chaque sous-routine.
- (f) **Estimation des paramètres :** Certains paramètres sont déterministes alors que d'autres sont stochastiques.
- (g) **Évaluation de l'état actuel du modèle :** Cette étape consiste à évaluer les performances du modèle puis de les comparer à celles du système réel.
- (h) **Analyse et interprétation des résultats :** Le concepteur passe à l'analyse et à l'interprétation de ces résultats pour donner des recommandations et des propositions.

Les types de simulation

Pour évaluer un système réel, on propose deux principales méthodes de simulation qui peuvent être utilisées en fonction de la nature du système cible.

- **La simulation de monte-carlo :** Elle se base sur la génération de nombres aléatoires, afin de reproduire les résultats d'un calcul pour lequel les données sont incertaines.
- **La simulation continue :** Elle permet d'analyser d'une manière continue le comportement du système, représenté sous la forme d'équations différentielles, au cours du temps.

Avantages de la simulation [57, 20]

- Observation des états du système.
- Études des points de fonctionnement d'un système. ‘
- Études de l'impact des variables sur les performances du système.
- Étude d'un système sans les contraintes matérielle.
- peut réaliser des procédures de file d'attente interactives.
- La demande peut varier à travers le temps et l'espace.

Inconvénients de la simulation

- La conception de modèles peut nécessiter des compétences.

- Résultats pas forcément généralisable
- Elle n'est possible que si le développeur comprend parfaitement le système.
- Certains utilisateurs peuvent appliquer le modèle sans connaître et apprécier les limites.

2.5.3 Prototype

- Elles sont exécutées sur le système réel ou sur le prototypes.
- Le coût d'étude est souvent très élevé et la durée d'étude longue.
- Elle est une phase de pré-implémentassions.

2.6 Conclusion

Dans ce Chapitre nous avons présenté quelques notions de base de l'évaluation de performances et définir ces méthodes (méthodes analytiques, simulation, prototype). Dans la chapitre suivant nous allons définir un outil de modélisation qui sera évalué par l'une des méthodes de l'évaluation de performances.

3

Réseaux de Petri

3.1 Introduction

La théorie des réseaux de Petri (RdP) a été développée à partir de la thèse de doctorat de Carl Adam Petri en 1962. Un réseau de Petri est un modèle abstrait et formel de flux d'informations. Les propriétés, les concepts et les techniques des RdP sont développés dans une recherche de ressources naturelles, des méthodes simples et puissantes pour décrire et analyser le flux d'informations et de contrôle dans les systèmes, en particulier les systèmes qui peuvent présenter des activités asynchrones et simultanées. La principale utilisation des réseaux de Petri a été la modélisation de systèmes d'événements dans lesquels il est possible que certains événements se produisent simultanément, c'est un outil de modélisation très puissant qui propose une modélisation statique et dynamique.

- Modélisation statique, qui décrit l'architecture du système.
- Modélisation dynamique, qui décrit les comportements possibles du système.

3.2 Notion de base

Définition informelle

- Un réseau de Petri est un graphe biparti, dont on particularise les deux familles de sommets : les places, les transitions et les arcs qui les relient. La façon dont ces transitions sont reliées définit le comportement d'un RdP.
 - Des places représentées par des cercles et peuvent être marquées par une ou plusieurs marques appelées jetons.
 - Des transitions représentées par des rectangles ou des traits, sous certaines conditions sur le marquage du réseau.
 - Les arcs sont représentés par des flèches qui lient une place à une transition ou inversement une transition à une place. ×

Définition formelle [1] D'une manière formelle un RdP non marqué est un quadruplet $R = (P, T, Pre, Post)$ tel que :

$P = \{p_1, p_2, \dots, p_n\}$: est un ensemble fini de places.

$T = \{t_1, t_2, \dots, t_m\}$: est un ensemble fini de transitions.

$Pre : P \times T \longrightarrow \mathbb{N}$: est l'application d'incidence avant (places précédents) ;

$Post : P \times T \rightarrow \mathbb{N}$: est l'application d'incidence arrière (places suivantes) ;

\mathbb{N} : est l'ensemble des entiers naturels.

$P \cap T = \emptyset, P \cup T \neq \emptyset$.

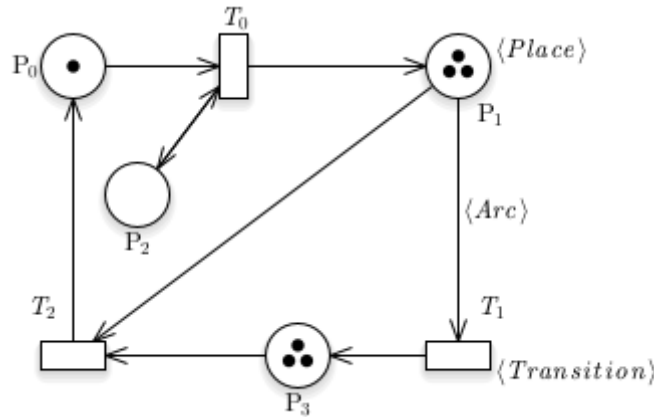


FIGURE 3.1: Exemple d'un réseau de Petri

Remarque 3.1 :

- Comme dans tout graphe biparti, un arc ne relie jamais deux sommets de la même famille, comme le montre la FIGURE 3.1. C'est aussi pareil pour les transitions.

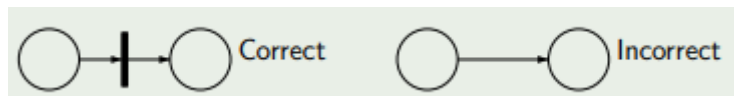


FIGURE 3.2: La structure d'un RdP

- Si un arc n'a pas de poids associé (appelé aussi valuation), c'est que ce poids vaut 1, sinon l'arc est biffé et une valuation lui est attribuée .

3.2.1 Matrice d'incidence

La matrice d'un RdP est la matrice entière C , elle traduit le coût global du franchissement d'une transition pour chaque place (la différence entre ce qui est produit et ce qui est consommé) [4].

La matrice d'incidence du réseau est [15] :

$$C = Post - Pre.$$

Matrice d'incidence avant : $Pre(p_i, p_j)$ indique le nombre de marquages qui doit contenir la place p_i , pour que la transition t_j soit franchissable.

$$C^- = [C_{ij}^-] \text{ où } C_{ij}^- = Pre(p_i, t_j)$$

Matrice d'incidence arrière : $Post(i, j)$ indique le nombre de marquages déposés dans la place p_i à la suite du franchissement de la transition t_j .

$$C^+ = [C_{ij}^+] \text{ où } C_{ij}^+ = Post(p_i, t_j)$$

Ainsi la matrice C^- fait le bilan des liaisons amont aux transitions, et la matrice C^+ fait le bilan des liaisons avant les transitions.

3.2.2 Notations matricielles :

- Un arc relie une place p à une transition t , si et seulement si $Pre(p, t) \neq 0$. Un arc relie une transition t à une place P , si et seulement si $Post(p, t) \neq 0$. Les valeurs non nulles des matrices Pre et $Post$ sont associées aux arcs comme étiquettes [48].
- Le réseau de Petri est un objet initialement graphique, peu exploitable sous cet aspect malgré la convivialité qu'il apporte, il peut être transcrit algébriquement. C matrice d'incidence qui va être décrite, fait la synthèse de tous les liens entre places et transitions du RdP.
- Cette matrice en générale est rectangulaire, possède un nombre de colonne P égal au nombre de transitions du réseau, ainsi qu'un nombre de lignes égal au nombre de places du réseau, chaque élément de la matrice témoigne de la présence ou de l'absence de lien, entre chaque place et chaque transition, ainsi que du poids attaché à l'arc en question. La direction de cet arc est transcrite par le signe de l'élément en question.

Exemple 3.1 : Soit le RdP dans la FIGURE 3.3 suivante :

Où tous les arcs sont de poids 1, on a :

$$C^- = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}; \quad C^+ = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

La matrice d'incidence :

$$C = \begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}$$

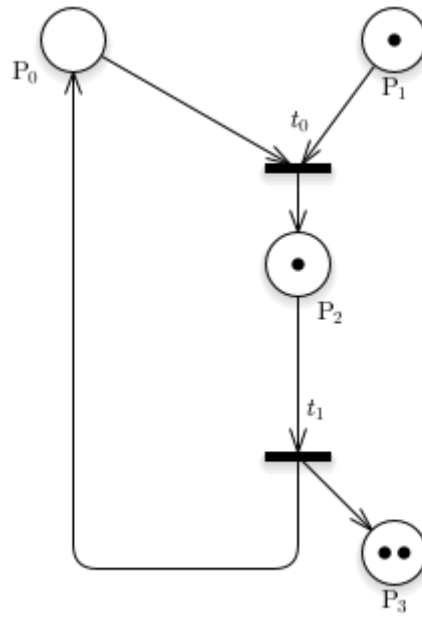


FIGURE 3.3: Exemple d'un RdP

3.3 Graphe associé

Un RdP peut se présenter par un graphe biparti (c'est un graphe dont les sommets se répartissent en deux ensembles disjoints P et T).

Définition 3.1 (Graphe associé) [(11)] : Le graphe $G = \langle P, T, L, V \rangle$ biparti associé au RdP $R = \langle P, T, Pre, Post \rangle$ est défini comme suit :

$$\begin{cases} L(p) = \{t \in T \mid Pre(p, t) > 0\} \\ L(t) = \{p \in P \mid Post(p, t) > 0\} \end{cases}$$

Où L permet d'obtenir les successeurs d'une place ou d'une transition.

3.4 Rdp marqué

Le marquage d'un RdP est un vecteur de dimension n (n : nombre de places du réseau) à composantes entières positives ou nulles[(11)].

Un Réseau Marqué R est un couple (R, M) constitué d'un réseau de Petri R et d'une application de marquage définie sur P et à valeurs dans \mathbb{N} (ie le marquage du réseau à un instant donné) Un RdP marqué et caractérisé par un couple $A = (R, M_0)$ où [(11)] :

- R : est un RdP.
- M_0 : est le marquage initial qui est une application définie par :

$$\begin{cases} M_0 : P \longrightarrow \mathbb{N} \\ P \longrightarrow M_0(p) \end{cases}$$

Exemple 3.2 :

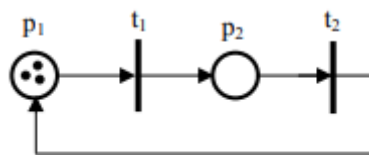


FIGURE 3.4: Un RdP marqué

Le marquage initial : $M_0 = [M_0(p_1); M_0(p_2)] = [3; 0]$

Remarque 3.2 Un marquage peut être représenté sous forme d'un vecteur colonne

$$M_0 = ((M_0(p_1); M_0(p_2)))^T = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Le marquage initial M_0 d'un RdP correspond à la distribution initiale des jetons dans chacune de places du RdP qui précise l'état initial du système dans ce cas, on parle du RdP marqué par opposition à un RdP non marqué, c'est à dire pour lequel le marquage initial n'est pas précisé.

Définition 3.2 (Transition source) Une transition source est une transition qui ne comporte aucune place d'entrée.

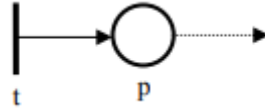


FIGURE 3.5: Transition source

Définition 3.3 (Transition puits) : Une transition puits est une transition qui ne comporte aucune place de sortie.

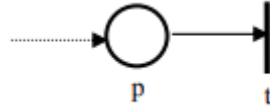


FIGURE 3.6: Transition puits

Définition 3.4 (Tir de transition) : Une transition t est tirable (ou franchissable, ou validée) lorsque [5] :

$$\forall p \in P, M(p) \geq \text{Pre}(p, t)$$

est une transition validée dans le marquage M_0 .

3.5 Évolution d'un RdP

L'évolution d'état du réseau de Petri correspond à une évolution du marquage, les jetons qui indiquent l'état du réseau à un instant donné, peuvent passer d'une place à l'autre par franchissement, ou par un tir d'une transition. Dans le cas des réseaux dites à arcs simples ou de poids égale à 1, le franchissement d'une transition consiste à retirer 1 jeton dans chacune des places d'entrée de la transition, et à en ajouter un, dans chacune des places de sorties.

En générale l'évolution des états d'un réseau de Petri marqué simple obéit aux règles suivantes [10] :

- Une transition est franchissable ou sensibilisée ou encore tirable lorsque chacune des places d'entrées, possède au moins le nombre de jetons correspondant au poids de l'arc reliant à la transition.
- Le réseau ne peut évoluer que par franchissement d'une seule transition à la fois transition sélectionnée parmi toutes celles validées au moment du choix.

- Le franchissement d'une transition est indivisible et de durée nulle.

Définition 3.5 (Évaluation de marquage)[5] Soit $A = (R, M_0)$ un réseau de Petri marqué, de transitions T et de places P . Le franchissement d'une transition t de T validée dans le marquage M_0 conduit au marquage M_1 :

- $\forall p \in P, \forall t \in T, M_1(p) = M(p) + C(p, t)$
- $\forall p \in P, \forall t \in T, M_1(p) = M(p) + Post(p, t) - Pre(p, t)$

On note alors $M[t > M_1$

Exemple 3.3 :



FIGURE 3.7: Avant Franchissement

Après Franchissement

Les marquages sont les suivants :

Avant franchissement : $M(P_1) = 3; M(P_2) = 4; M(P_3) = 1; M(P_4) = 0$

Après franchissement : $M(P_1) = 1; M(P_2) = 3; M(P_3) = 2; M(P_4) = 3$

Remarque 3.3

Le franchissement d'une transition ne garantit pas la conservation de la quantité des marques globales. Dans l'exemple précédent, on a globalement un jeton de plus après franchissement de la transition. Selon les poids attribués aux arcs liés à une transition donnée, les transitions sont :

"Consommatrice", "Génératrice" ou "Conservatrice" de marques.

Définition 3.6[29] : Soit $Post(j)$ le poids d'un arc aval j de la transition T

- ★ si $\sum_n Pre(i) < \sum_m Post(j)$, alors T est génératrice.
- ★ si $\sum_n Pre(i) > \sum_m Post(j)$, alors T est consommatrice.
- ★ si $\sum_n Pre(i) = \sum_m Post(j)$, alors T est conservatrice.

3.5.1 Franchissement d'une transition

Si t est franchissable pour le marquage M , le franchissement (tir, firing) de t donne le nouveau marquage M' tel que [(6)] :

$$(\forall p \in P) M'(p) = M(p) - Pre(p, t) + Post(p, t)$$

Définition 3.7 (Franchissement d'une transition source) Une transition source est une transition qui ne comporte aucune place d'entrée, c'est une transition toujours franchissable et le franchissement a lieu lorsque l'événement associé se produit.

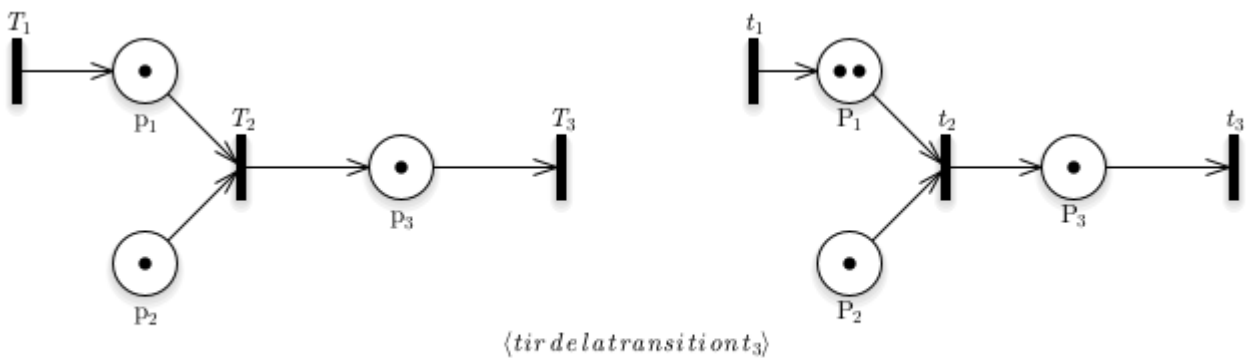


FIGURE 3.8: Franchissement d'une transition source

Définition 3.8 (Franchissement d'une Transition puits) Une transition puits est une transition qui ne comporte aucune place de sortie, le franchissement d'une transition puits enlève des jetons de toutes les places d'entrée de la transition.

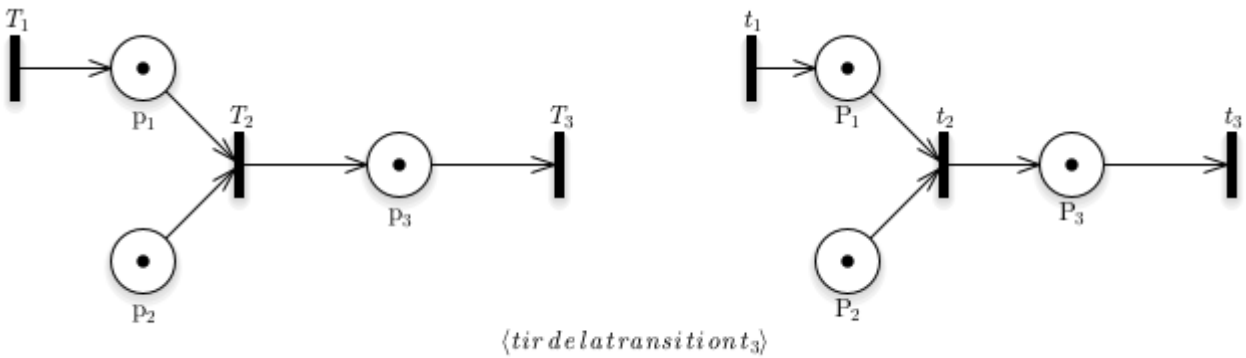


FIGURE 3.9: Franchissement d'une transition puits

3.5.2 Conflit et parallélisme

L'avantage des Rdp réside dans leur capacité à modéliser un grand nombre de comportements dans les systèmes complexes. Parmi ces comportements, nous trouvons le parallélisme, la synchronisation, le partage de ressources, les conflits, etc.

Définition 3.9 (Le parallélisme) Le parallélisme est défini comme l'évolution simultanée de plusieurs processus dans un même système. Dans un Rdp, le parallélisme est déclenché avec une transition ayant plusieurs places de sortie[(11)].

1/Parallélisme structurel

Deux transitions t_1 et t_2 sont parallèles structurellement si :

$$Pre(\bullet, t_1)^T \times Pre(\bullet, t_2) = 0$$

Elles n'ont donc aucune place d'entrée commune (le produit scalaire de leurs vecteurs Pre sont nul).

2/Parallélisme effectif

Deux transition t_1 et t_2 sont parallèles pour un marquage donné M si est seulement si elles sont parallèles structurellement et [(9)] :

$$M(p) \geq Pre(p, t_1)$$

$$M(p) \geq Pre(p, t_2)$$

Définition 3.10 (Conflit)[9, 55] Le conflit est l'existence d'une place qui a au moins deux transitions de sortie. La notion de conflit modélise un choix ou une décision. Deux sortes de conflit existent

1/Conflit structurel

Une place $p_i \in P$ constitue un conflit structurel si elle est une place d'entrée pour au moins deux transitions.

$$\exists p \quad Pre(p, t_1) \cdot Pre(p, t_2) \neq 0$$

2/Conflit effectif

On parle d'un conflit effectif entre deux transitions en conflit structurel s'il existe un marquage qui sensibilise les deux transitions et que le franchissement d'une transition empêche le franchissement de l'autre une seule transition sera franchie mais rien dans le réseau ne permet de prévoir laquelle [(9)].

$$M \geq Pre(p, t_1)$$

$$M \geq Pre(p, t_2)$$

3.6 Séquence de franchissement

Définition informelle [5] Le franchissement successif de transitions (sensibilisées) dans un ordre donné, à partir d'un marquage donné constitue une séquence de franchissements.

- On s'intéresse à l'évolution du réseau, lors du tir successif de plusieurs transitions.
- Lorsque $M[t_1 t_2 > M_2]$, on dit que la séquence de transitions $t_1 t_2$ est franchissable depuis le marquage M .
- On note $M[t_1 t_2 > M_2]$.

Une séquence de franchissement est un mot construit sur l'alphabet T^* des transitions de T . On note s une séquence de franchissement.

Définition Formelle [33]

- Soit (R, M_0) : un RdP marqué.
- $s = t_1, t_2, \dots, t_n \in T^*$: une séquence de transitions.
- La séquence s est franchissable depuis M , si et seulement s'il existe des marquages M_1, M_2, \dots, M_n , tels que
:

$$M_1[t_1 > M_2[t_2 > \dots M_{n-1}[t_{n-1} > M_n.$$

Avec T^* est un sous ensemble de T constitué des transitions qui forment la séquence de franchissement. Dans ce cas le tir ou le franchissement conduit au marquage M_n , on note alors $M(s > M_n)$ [33].

Définition 3.11 (Équation d'état) : Un RdP possède un état d'accueil M_a pour un marquage initial M_0 , si pour tout marquage accessible M_i , il existe une séquence de tirs s telle que $M_i[s > M_a$, il s'en suit qu'un RdP est réinitialisable pour un marquage initial M_0 si M_0 est un état d'accueil.

Remarque 3.4

- Il s'agit d'une condition nécessaire mais pas suffisante : il se pourrait que s ne soit pas franchissable depuis M .
- Deux étapes pour calculer un marquage :
 1. Démontrer que le marquage valide la séquence;
 2. Calculer le marquage résultat.

Définition 3.12 (Marquage accessible) [6] : Franchissement d'une transition validée dans un RdP apporte une modification au marquage initial M_0 . Un marquage M_n est dit accessible à partir du marquage initial M_0 , s'il existe une séquence de franchissement $s = T_1, T_2, \dots, T_n$ qui transforme M_0 en M_n . Dans ce cas on écrit :

$$M_0[s > M_n)$$

: qui signifie que M_n est accessible à partir M_0 par s .

- Ensemble de toutes les marques accessible à partir de M_0 dans un RdP noté $R(M_0)$.
- Ensemble de toutes les séquences de franchissement possibles à partir de M_0 est noté $L(M_0)$.

$$M_i[s > M_k$$

Définition 3.13 (Ensemble d'accessibilité) [55] : Soit (R, M_0) un RdP. L'ensemble des marquages accessibles ou ensemble d'accessibilité de ce réseau est noté $A(R, M_0)$ ou A , est l'ensemble des marquages atteints par une séquence de franchissement :

$$A(R, M_0) = \{M \in NP / \exists s \in T^*; \text{et } M_0(s > M)\}.$$

Définition 3.14 (Graphe des marquages accessibles) L'ensemble des marquages accessibles $A(R, M_0)$ est défini par [5] :

$$\begin{cases} M_0 \in A(R, M_0) \\ \text{si } M \in A(R, M_0) \text{ et } M[t > M' \text{ alors } M' \in A(R, M_0). \end{cases}$$

Cette définition signifie que le marquage initial est un marquage accessible et tout marquage successeur d'un marquage accessible est un marquage accessible.

Exemple Graphe de marquage :

Soit le réseau Rdp marqué suivant :

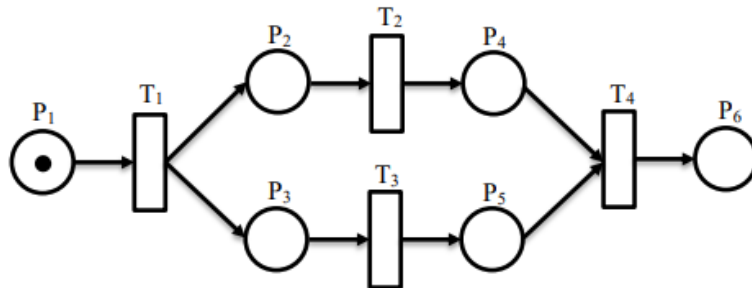


FIGURE 3.10: Exemple d'un réseau de Petri marqué

Prenant le réseau de Petri de la figure 3.10, l'ensemble des marquages accessibles est $M_0, M_1, M_2, M_3, M_4, M_5$. Le graphe de marquages est représenté par la figure 3.11 :

3.7 Propriétés d'un réseau de Petri

Le modèle de RdP nous donne des techniques pour analyser les propriétés qui sont les caractéristiques qui permettent d'évaluer la qualité d'un système donné. Elle peuvent être classées en deux groupes.

3.7.1 Les propriétés dynamiques

Ces propriétés dépendent à la fois du marquage initial M_0 et de la structure du réseau [7, 8, 6].

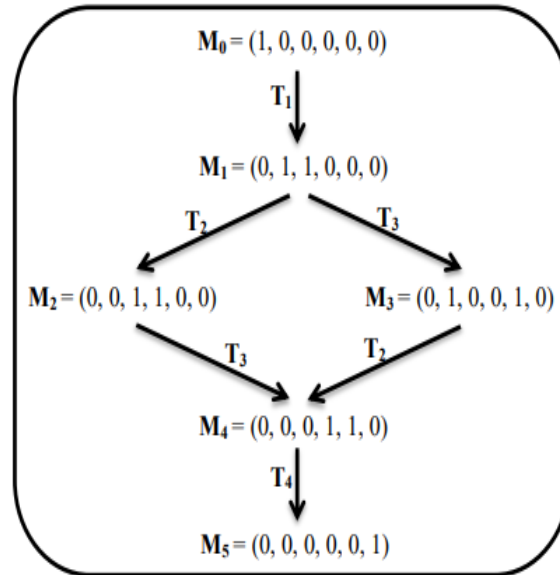


FIGURE 3.11: Graphe de marquage

1. **Réseau borné** La bornitude d'un RdP exprime le fait que le nombre d'états que peut prendre le système modélisé par ce RdP est fini, autrement dit, le nombre de marquages accessibles est fini.

Dans le cas contraire, où le RdP est non borné, le nombre d'états est infini et ceci est dû, au fait que certains paramètres de ce système sont non bornés.

Un RdP marqué est k -borné si toutes ses places sont k -bornées, c'est-à-dire, quel que soit le marquage accessible M à partir du marquage initial M_0 , et quel que soit la place P , considérée le nombre de jetons contenus dans cette place est inférieur ou égale à une borne k .

Définition 3.15 Soit un RdP $R = (P, T, Pre, Post)$. une place $p \in P$ est dite K -bornée pour un marquage initial M_0 si et seulement si :

$$A(R, M_0) \text{ est } k\text{-borné} \leftrightarrow \forall M \in R(M_0), \forall p \in P : M(p) \leq k$$

. Un RdP marqué est sauf ou binaire pour un marquage initial M_0 s'il est 1-borné.

2. **Vivacité [8]** : Le réseau de Petri R est dit vivant si, quel que soit le marquage atteint à partir de M_0 , il est possible de franchir toute transition du réseau en progressant à travers une certaine séquence de franchissement . Un Rdp vivant est un réseau de Petri sans blocage. Un blocage est un marquage tel qu'aucune transition n'est validée.

Définition 3.16 : Soit un RdP $R = (P, T, Pre, Post)$.

$$RdP \text{ vivant} \Leftrightarrow \forall T_j \in T, \forall M \in R(M_0), \exists S/T_j \in S, \exists M' \in R(M_0) / M[S > M'.$$

3. **État d'accueil et réseau de Petri réinitialisable :** Un réseau de Petri possède un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage accessible $M \in R(m_0)$, il existe une séquence de franchissement s tel que, $M[s > M_a$. Un réseau de Petri est réinitialisable pour un marquage initial M_0 si M_0 est un état d'accueil.

4. **Accessibilité :** Le franchissement d'une transition validée dans un réseau de Petri apporte une modification au marquage initial du réseau.

Un marquage M_k est dit accessible à partir du marquage M_0 , s'il existe une séquence de franchissement $s = T_1 T_2 \dots T_k$ qui transforme M_0 en M_k .

Dans ce cas, on écrit $M_0[s > M_k$ ce qui signifie que M_k est accessible à partir de M_0 par s . L'ensemble de tous les marquages accessibles à partir de M_0 dans un réseau de Petri R est noté $R(M_0)$. L'ensemble de toutes les séquences de franchissement possibles à partir de M_0 dans un réseau de Petri est noté $R(M_0)$

5. **Blocage :** Un blocage est un marquage tel qu'aucune transition n'est validée.

3.7.2 Les propriétés structurelles

Les propriétés structurelles dépendent uniquement de la topologie du réseau. Il s'agit de faire ressortir les propriétés statique du système étudié. Ces différentes propriétés sont indépendantes du marquage [12, 13].

1. **P-invariant :** les invariants de marquage appelés p-invariant ou encore p-semi flots, illustrent la conservation du nombre de jetons sans un sous ensemble de places du RdP. Un vecteur noté Y de dimension égale au nombre de places du RdP est un p-invariant, si et seulement s'il vérifié l'équation suivante :

$$Y^t \times C = \vec{0}$$

L'ensemble des places pour lesquelles la composante associée dans le p-invariante est non

nulle est appelée la composante conservatrice du RdP, où C correspond à la matrice d'incidence du RdP.

2. **T-invariant** : un vecteur non nul d'entiers X est un T-invariant, ou encore T-semi flots du RdP, si et seulement il vérifie l'équation suivante :

$$C \times X = 0$$

Un T-invariant correspondant à une séquence de franchissement réalisable appelé composante répétitive.

3.8 Réseaux de Petri particuliers

Quelques définitions des RdP particuliers [6, 7, 14, 15] :

1. **Graphe d'état** : Un graphe d'état est un RdP, où chaque transition possède exactement une place d'entrée et une place de sortie.
2. **Graphe d'événement (GEV)** : Un graphe d'événement est un RdP, où chaque place possède exactement une transition d'entrée et une transition de sortie. Et parfois appelé graphe de transition.
3. **RdP sans conflit** Un RdP est un réseau sans conflit, si et seulement si chaque place a au plus une transition de sortie. Un RdP avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties.
4. **RdP Simple** : Un RdP simple est un RdP, ordinaire tels que chaque transition a au plus une place d'entrée qui peut être reliée à d'autres transitions.
5. **RdP autonome** : Un RdP autonome décrit le fonctionnement d'un système, dont les instants de franchissement ne sont pas connus ou indiqués.
6. **RdP non autonome** : Un RdP non autonome décrit le fonctionnement d'un système; dont l'évolution est conditionnée par les événements externes ou par le temps. On peut dire que un RdP non autonome est synchronisé et/ou temporisé.
7. **RdP Pur** Un RdP pur est un réseau dans lequel, il n'existe pas de transition ayant une place d'entrée qui soit à la fois place de sortie de cette transition.

8. **RdP Impure** Un RdP impure est un réseau dans lequel, il existe une transition ayant une place qui soit à la fois place d'entrée et place de sortie.
9. **RdP généralisé** Un RdP généralisé est un RdP dans lequel les poids sont associés aux arcs.
10. **RdP a capacité** Un RdP a capacité est un RdP dans lequel des capacités (nombre entiers strictement positifs) sont associés aux places. Le franchissement d'une transition d'entrée d'une place P_i dont la capacité est $cap(p_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i qui dépasse cette capacité.
11. **RdP priorité** Dans un tel réseau, si on atteint un marquage tel que, plusieurs transitions sont franchissables on doit franchir la transition qui a la plus grande priorité.

3.9 Extensions des RdP

Le concept RdP classique a été largement développé par de nombreux auteurs dans les années 70, dans le monde entier, en intégrant particulièrement l'aspect temporel et stochastique dans le modèle initial. Ci-dessous, nous introduisons quelques extensions.

3.9.1 Réseaux de Petri à temps discret

Les réseaux de Petri à temps discret peuvent être classés en deux familles :

- Les RdP temporisés qui permettent une modélisation du temps sur un espace entier.
- Les RdP stochastiques qui ont un comportement qui n'est pas déterministe, mais stochastique, ils permettent de modéliser les processus aléatoires.

Réseaux de Petri stochastiques (RdPS)

Les réseaux de Petri stochastiques (RdPS) occupent une place prépondérante en tant qu'approche d'évaluation des performances des systèmes informatiques ou industriels [58]. Les problèmes d'évaluation liés à la sûreté faisant intervenir des phénomènes aléatoires, les transitions du réseau de Petri ont comporté des temps de franchissement aléatoires, distribués par une loi exponentielle. Cette distribution exponentielle permet d'exploiter les propriétés mathématiques d'un processus de Markov.

Définition 3.18 : Un réseau de Petri stochastique est le couple $(R; \wedge)$ avec [55] :

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

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

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

La fonction $P_\theta(x)$ décrit la probabilité que la durée de sensibilisation soit inférieure ou égale à x . Avec les RdPS, on pourra par exemple, calculer le temps de bon fonctionnement entre deux défaillances, le temps de réparation ou dans certains cas la durée opérationnelle d'une machine, les taux de production, l'évolution des stocks, etc.

Réseaux de Petri temporisés

L'introduction des temporisations dans un réseau de Petri permet de décrire un système dont le fonctionnement dépend du temps et permet ainsi d'envisager une analyse quantitative du système étudié. Dans un réseau de Petri temporisé, les temporisations sont associées aux places (RdP P-temporisé) ou aux transitions (RdP T-temporisé)[30]. Ces deux façons d'envisager les temporisations conduisent à des outils équivalents, même si la mise en oeuvre est différente. Ils sont utiles pour l'évaluation des performances des systèmes modélisés.

Définition 3.19 : Un RdP temporisé est défini par le couple (R, d) avec [55] :

- R est un réseaux de Petri $R = (P, T, pre, Post, M_0)$;
- $d : T \Rightarrow Q^+$ est la fonction de temporisation.

3.9.2 Les RdP étendus

Réseaux de Petri à arcs inhibiteurs :

Un arc inhibiteur est un arc orienté qui part d'une place (P) pour aboutir à une transition[15]. la transition (t) n'est validée que si la place (P) ne contient aucune marque.

Le franchissement de t consiste à retirer une marque dans chaque place d'entrée de t à l'exception de P , et à ajouter une marque dans chaque place de sortie de t . On utilise aussi les expressions "test à zéro" et "RdP étendus".



FIGURE 3.12: Représentation d'un arc inhibiteur.

Définition 3.20 [4] :

- Un RdP à arcs inhibiteur est défini par un 5-uplet $R = (P; T; Pe; Post; Inh)$, où :
- P : est un ensemble fini de places et T un ensemble fini de transitions ;
- $Pre : P \times T$ et $Post : P \times T$ sont les fonctions d'incidence avant et d'incidence arrière respectivement ;
- $Inh : P \times T \longrightarrow N/\{0\}$ est la fonction d'inhibition.

Les réseaux de Petri à priorité :

Un tel réseau est utilisé lorsque l'on veut imposer un choix entre plusieurs transitions validées. [15]

Définition 3.21 : Un RdP étendu est un tuple $RdPE = (R, Inh, >, M_0)$, où :

- R est un RdP ;
- $Inh : P \times T \longrightarrow \mathbb{N}^*$
est l'application d'inhibition qui associe à tout couple (p_i, t_j) le poids de l'arc inhibiteur reliant la place p_i à la transition t_j .
- $>$ est une relation de priorité entre transitions ;
- M_0 le marquage initial.

3.9.3 Réseaux de Petri hiérarchiques

Un réseau de Petri hiérarchique se compose de plusieurs modules ‘pages’, où chaque module représente un sous réseau de Petri. Le référencement d’un module se fait avec une transition, ou une place de substitution portant le nom du module en question. L’utilisation des RDP hiérarchiques est préconisée pour la modélisation des problèmes de grande taille[32], construire un modèle en combinant un certain nombre des réseaux en un seul réseau. L’objectif est de réduire la complexité d’un modèle en permettant la modélisation modulaire [31].

3.9.4 Réseau de Petri coloré

Définition 3.22

Les réseaux de Petri ordinaire sont généralement insuffisant et doivent être étendus pour qu’il serait applicable à des problèmes réels. Les RdP colorée (Jensen and Rozenberg) offrent une possibilité d’expression plus compacte que les réseaux Petri ordinaires. Parmi les caractéristique d’un RdP coloré, nous citons[22] :

- Chaque place contient des jetons typés (colorés). Nous associons à chaque place et transition des domaines de couleurs.
- Un arc liant une place et une transition est étiqueté par une application linéaire appelée fonction de couleur.
- Les transitions du réseau peuvent être munies d’une garde (une condition booléenne sur la couleur de la transition qui limite le tir de la transition aux couleurs pour lesquelles la garde est vraie).

Les RdP colorés ou dérivés des RdP ordinaires sont des RdP dans lesquels les jetons portent des couleurs Chaque couleur correspond à une information bien définie associée au jeton. Deux principales raisons justifie l’appel aux RdP colorés : [21]

- Possibilité de transporter une information structurée car les places ne contiennent pas que des jetons uniformes.
- Les RdPC s’utilise pour éviter le problème de taille importante des RdP ordinaires dans le cas ou des entités différentes présentent des comportements similaires ce qui condence le modèle.

Définition 3.23,[21] :

Un réseau de petri coloré non hiérarchique est un produit cartésien $R = (\Sigma, P, T, A, N, C, G, E, I)$ défini par :

- Σ : l'ensemble fini de types non vides, appelés ensembles de couleurs, P : l'ensemble fini des places,
- T : l'ensemble fini des transitions,
- A : l'ensemble fini d'arcs tel que $P \cap T = P \cap A = T \cap A = \emptyset$
- N : la fonction des nœuds. Elle est définie de A vers $P \times T \cup T \times P$
- C : la fonction de couleurs de P vers Σ
- G : la fonction des gardes

Définition 3.24 (Marquage coloré),[22] :

Soit R un RdP coloré. Un marquage de R est une fonction qui à toute place $p \in P$ associe un élément de $\text{Bag}(C(p))$ (ensemble des multi-ensemble). L'ensemble des marquages de R est noté M_R .

Définition 3.25 (Règle de franchissement), [22] :

Une transition T est franchissable pour une instance $c_T \in C_T$ et un marquage M si et seulement si :

- Soit T est non gardée, soit la garde vaut 1 (=true) pour c_T .
- $\forall p \in P, M(p) \geq C^-(P, T)(c_T)$

3.10 Outils de modélisation des réseaux de Petri

Avec l'évolution, plusieurs chercheurs ont développé des outils de simulation et de vérification des RdP tel que (CPNTools, Greatspn, CPNAMI, PROD, JARP, MARIA, LOLA, Petri Net Kernel, INA, OPMSE, Artifex, ExSpect, FLOWer,...). Ces outils présentent un environnement graphique d'édition des RdP avec la possibilité de simuler le modèle et d'analyser des propriétés génériques des RdP.

3.11 Conclusion

L'avantage des réseaux de Petri est d'une part leurs fondation mathématique forte et d'autre part leur capacité à prendre en compte la concurrence.

Nous avons présenté dans ce chapitre le formalisme d'un RdP, quelques définitions et propriétés, nous avons cité quelques unes des extensions apportées à ces RdP.

Dans les chapitres suivante nous allons proposé une modélisation des web services à l'aide des réseaux de Petri colorés et évaluer ses performances en utilisant un simulateur de RdPC.

4

Étude de l'existant sur l'évaluation des performances des Web services

4.1 Introduction

Dans ce chapitre, nous allons présenter quelques travaux sur l'évaluation de performances des Web services, ou ils ont utilisé des méthodes de performance pour évaluer les performances des différents systèmes et nous allons faire une comparaison entre ces travaux.

4.2 Travaux sur l'évaluation de performances des Web services

Nous allons présenter 4 travaux d'évaluations de performances des Web services, décrivons le modèle, la méthode utilisée et les indices de performances calculés

Dans [25], ils ont proposé un modèle analytique basé sur les réseaux de Petri stochastique tenant en compte l'arrivée des clients et l'arrivée des Web services, avec deux stations; la première représente une demande de client et la deuxième représente une demande des Web services. Permettent d'évaluer les performances d'un système de Web services, ils ont calculé le temps moyen de réponse et le nombre moyen de client dans le système.

Dans [26], il a développé une application prototype pour mesurer le temps d'échange de messages de Web services RESTful et de Web service SOAP/WSDL avec le même client, et les comparés pour évaluer leur performances. De plus il a calculé le temps moyen de réponse et la taille de message.

- Il a développé une application prototype en utilisant le langage de programmation Java.
- Il a mesuré le temps total d'échange de messages des services Web RESTful et des services Web conventionnels SOAP / WSDL et je les ai comparés pour évaluer les performances.
- Il a calculé le temps moyen d'échange des messages, le temps moyen de réponse et la taille de message.

Dans [27], ils ont proposé un modèle de réseau de Petri coloré avec synchronisation de deux files d'attente, où la première présente les Web services et la deuxième présente les demandes des clients. Ils ont utilisé la méthode analytique pour calculer le nombre moyen de clients dans le système et le temps moyen de séjour dans le système.

Dans [28] ils ont proposé des solutions pour la découverte, la sélection des Web services et leur composition de telle façon à optimiser la recherche en terme de temps de réponse. L'évaluation

des performances des solutions proposées sont basées sur la simulation :

- ★ Sélection de Web service basé sur les contraintes QoS, ils ont calculé le temps de réponse et de débit.
- ★ Sélection de Web services composites pour un groupe de clients, où ils ont calculé le temps de réponse et le coût.

Dans [15] ils ont essayé d'évaluer et de comparer les performances des quatre protocoles de routage Ad hoc DSR, AODV, OLSR et DSDV en utilisant le simulateur réseau NS-2 . Les métriques d'évaluation de performances considérées dans cette étude sont le coût de routage, le taux de paquets délivrés, le délai, la gigue et la concentration de l'activité. Ils ont réalisé des expérience avec le simulateur a fin d'analysé les métriques de performances des protocoles de routage.

Dans [59] ils proposé un système Web services simples est composites, modélisé par un réseau de file d'attente. le modèle obtenu est markovien avec deux quantités stochastique principale (les temps des inter arrivée et la durée de service). La propriété sans mémoire de la loi exponentielle facilite l'analyse de ce modèle. Ils ont calculé le taux d'arrivée et la durée moyenne de séjour dans le système.

Dans [60] ils ont proposé un système Web services simple, modélisé par un réseau de Petri colorée simple de type demande/ réponse qui se compose d'un client et d'un Web services. Les clients sont les même et les clients aussi sont les même. Le client envoie au Web service une demande de liste d'enregistrements de client, le Web service reçoit la demande et répond en renvoyant un nombre spécifier d'enregistrements client/sans application de Web services en fonctions des informations contenues dans la demande. Ils ont calculé la taille de message et le coût .

Dans [61] ils proposé un système Web services simples est composites, modélisé par un réseau de file d'attente. le modèle obtenu est markovien avec deux quantités stochastique principale (les temps des inter arrivée et la durée de service). La propriété sans mémoire de la loi exponentielle facilite l'analyse de ce modèle, a l'aide d'un simulateur implémenté sur Matlab ils ont calculé la durée moyenne et le temps moyenne .

Dans [62] une approche orientée d'un modèle de Web services proposé pour la vérification et l'évaluation des performances de Web service proposé ou ils ont pris en considération que les web services sont différents et la demande des clients sont différents aussi, modélisé par un réseau de

petri temporisé à l'aide de l'algèbre(max,+). Où départ ils ont proposé des modèles mathématiques pour décrire le comportements analytique des différents patterns participant à la composition de service proposé. puis a l'aide d'un prototype ils ont calculé le temps moyen de réponse et le délai. Dans [63] Ils ont proposé un Web services composites, modélisé par les files d'attente. On obtient un modèle s'intéresse a la demande de client et le Web service, à l'aide de la méthode analytique ils ont calculé le temps de réponse et la durée moyenne de séjour.

4.3 Étude comparative des travaux

La TAB 4.1 montre une étude comparative pour les travaux présenté précédemment sur l'évaluation de la performance des Web services.

Modèle : Identifier les méthodes utilisé.

Solution : méthode analytique, simulation et un prototype.

Évaluation de performance : temps de réponse, coût, etc.

Web service : nous avons deux types, Web service simple, Web service composites.

Mark ✓ signifier que les auteurs pris en compte la caractéristique.

		TR1 [25]	TR2 [26]	TR3 [27]	TR4 [28]	TR5 [15]	TR6 [59]	TR7 [60]	TR8 [61]	TR9 [62]	TR10 [63]
Modèle	-RdPC - RdPS - fille d'attente -RdT	✓		✓			✓	✓		✓	✓
Solution	-solution analy- tique -Simulation -prototype	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Évaluation de performance	-disponibilité -temps de réponse -coût -nombre moyen -débit -requête de client - temps d'arrivée - TMDS - durée moyenne	✓	✓	✓	✓	✓		✓	✓	✓	✓
Web service	-simple -composite	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 4.1: Étude comparative des travaux de l'existence de l'évaluation de performances des Web services

À travers la TAB 4.1 nous avons constaté que les auteurs dans les recherches s'intéressant à ces états :

- L'arrivée des clients et les Web services simultanément ([25, 27, 15, 61]).
- L'arrivée des clients et les Web services simultanément, et de prendre en considération le faite que les clients sont différents et les Web services sont différent. ([27, 62]).
- L'arrivée des Web services ([26, 28, 63]).
- l'arrivée des demande des clients ([60],

4.4 Conclusion

Dans ce chapitre nous avons présenté quelques travaux d'évaluations de performances des Web services, ainsi qu'une étude comparative pour ces travaux. À travers cette comparaison nous nous somme concentrés sur le modèle utilisé, la solution et l'évaluation de performance par rapport à certains critères (Temps de réponse, débit,...).

Dans le chapitre qui suit nous allons proposé un modèle basé sur les réseaux de petri colorés, nous prenons en considération les arrivées des clients et des Web services, et on prendre en considération le faite que les clients sont différents et les Web service sont différents et qu'un client veut un Web service précis, à l'aide d'un simulateur CPN tools nous allons évaluer les performances du modèle.

Modélisation d'un système des Web services

5.1 Introduction

Dans ce chapitre nous allons modéliser et évaluer les performances d'un système de Web services. Nous avons construit un modèle avec les réseaux de petri colorés. Cette approche nous a permis d'avoir un modèle du système de Web services, qui prend en compte deux types différents d'arrivées : les clients et les Web services, sachant que les clients sont différents et les Web services sont différents, de plus un clients cherche un Web service, bien précis. Les performances de ce modèle sont calculées à l'aide d'un simulateur.

5.2 Modélisation des demandes des clients dans un système de Web services.

Le modèle que nous avons proposé décrit le système de Web services, où l'arrivée des demandes clients et des Web services sont prises en compte pour évaluer ses performances, avec un serveur exponentiel. Dans ce modèle, nous avons supposé qu'il y'a deux demandes différentes des web services noté w_1 w_2 , qui sont définit comme suit :

- Une seule arrivé de Web service de type w_1 .
- Quatre arrivées de Web services de type w_2 .

On a deux demandes différentes des clients noté c_1 et c_2 , qui sont définit comme suit :

- Deux requêtes des clients de type c_1 .
- Trois requêtes des clients de type c_2

Notons que les arrivée sur le réseau suivent la distribution de Poisson.

La FIG 5.1 présente de réseau de Petri coloré d'un système de Web services.

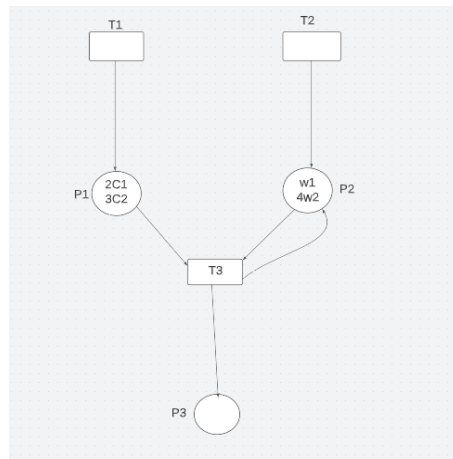


FIGURE 5.1: Modèle de réseau de Petri coloré d'un Web services

5.3 Étude du modèle du réseau de Petri coloré

Nous avons implémenté le modèle sur le simulateur CPN tools, la FIG 5.2 montre-nous comment le modèle apparaît sur l'interface du simulateur :

dans la partie index nous avons déclaré trois différent places le première elle définit les client(colset CL) contient deux différent variable (Var in1,in2), la deuxième définit les Web services (colset WS) contient deux différent variable (var s1, s2), et la dernière présente les résultats (Colset OUT) avec une variable (var b1).

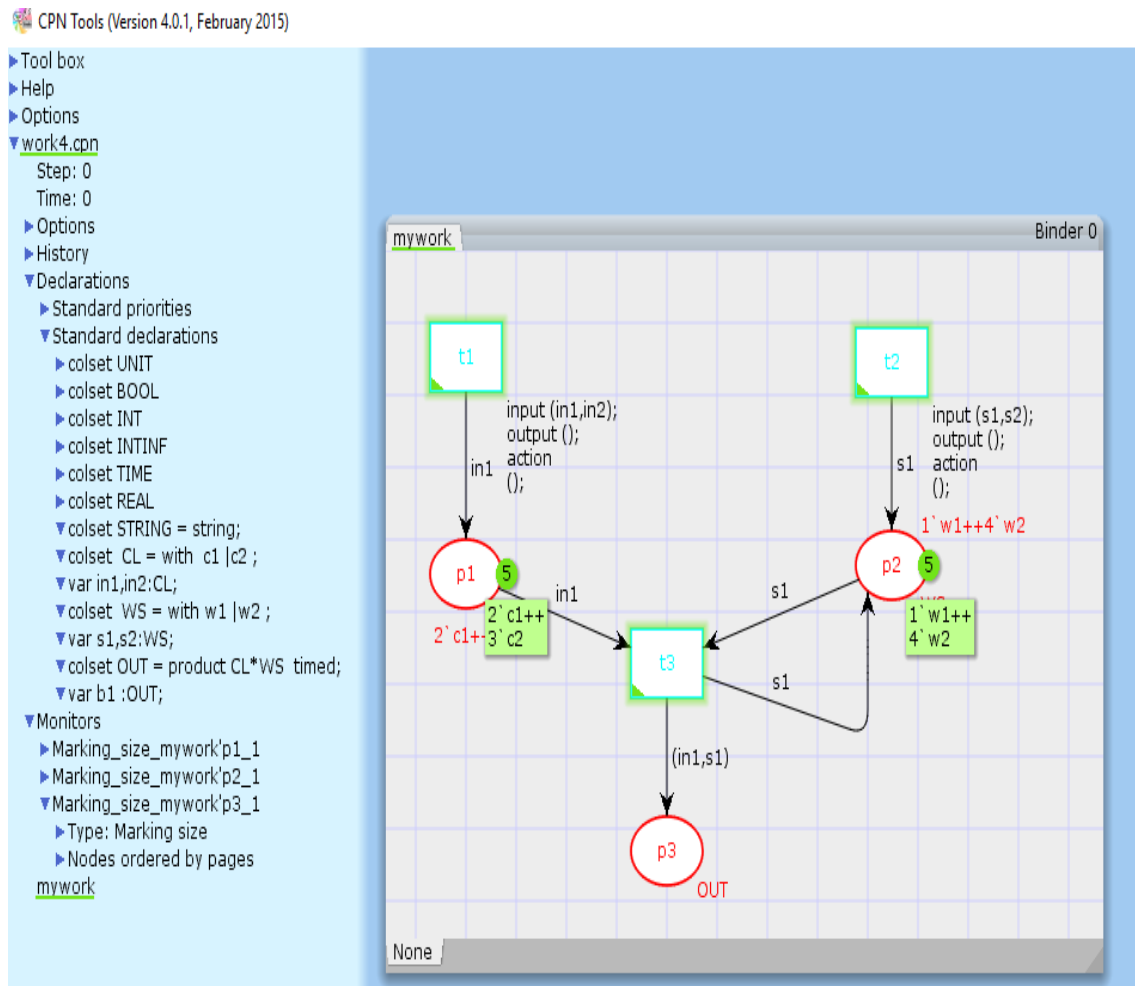


FIGURE 5.2: Le modèle proposé sur l'interface de simulateur

CPN Tools state space report for:
/cygdrive/C/Users/mis/Desktop/projet fin d/work4.cpn
Report generated: Wed Jun 22 13:40:12 2022

Statistics

State Space

Nodes: 13701
Arcs: 147144
Secs: 300
Status: Partial

Scc Graph

Nodes: 13701
Arcs: 147144
Secs: 5

FIGURE 5.3: Rapport 1

Home Markings

None

Liveness Properties

Dead Markings

6630 [9999,9998,9997,9996,9995,...]

Dead Transition Instances

None

Live Transition Instances

None

Fairness Properties

No infinite occurrence sequences.

FIGURE 5.4: Rapport 2

D'après la FIG 5.3 obtenue lors de la vérification avec CPN tools, on constate que le nombre de noeuds dans SCC Graph est le même dans le modèle.

La FIG 5.4 elle montre que le système reçoit toujours des arrivées de l'extérieur, qui arrivent vers les files d'attente. Le modèle est Vivant et par conséquent, n'a pas de transitions mortes. Chaque marquage accessible est une état d'accueil, donc le modèle est réversible donc réinitialisable. Alors on peut dire que le modèle est validé car il vérifie les propriétés suivantes (**Vivacité, Bornitude, Réversibilité**).

5.4 Résultats

de la simulation

5.4.1 Nombre moyen des client, des Web services dans le système et nombre moyen des clients servis

Nous avons fait une simulation sur le simulateur CPN tools pour notre modèle qui commence de 100 step -10000 step, et nous avons résumé les résultats dans les tableaux ci-dessous. La TAB

N^i simulation	Nombre moyen
100	70
1000	686
10000	6606
100000	66487

TABLE 5.1: Nombre moyen des client dans le système

5.1 nous montre le nombre moyen des clients dans le système qui attendent pour être servis par le Web services dont il a besoin. Nous remarquons qu'à chaque fois que le nombre de simulation augment le nombre moyen des client dans le système augment a cause des arrivées des clients de l'extérieur.

N^i simulation	Nombre moyen
100	68
1000	652
10000	6616
100000	66733

TABLE 5.2: Nombre moyen des Web services

A partir des TAB 5.2, nous remarquons qu'à caque fois le nombre de simulation augment le nombre moyen des Web services dans la place p2 augment (les Web services attend dans la file d'attente jusqu'à un client demande ce service).

$N^{\text{simulation}}$	Nombre moyen
100	36
1000	336
10000	3220
100000	33218

TABLE 5.3: Nombre moyen des client servis

La TAB 5.3 nous montre le nombre moyen des clients servis par les Web service. À partir de la TAB 5.3, nous remarquons qu'à chaque fois le nombre de simulation augment le nombre moyen des clients servis augment ce qui signifie que certaines clients on trouver le Web service dont il sont besoin.

Lors de la simulation la place p_3 avec la colset OUT elle ne permettre de récupérer le nombre moyen des clients qui seront servis par les Web services w_1, w_2 . le TAB 5.4 résume les résultats :

$N^{\text{simulation}}$	(c_1, w_1)	(c_1, w_2)	(c_2, w_1)	(c_2, w_2)
100	5	11	8	12
1000	69	107	69	91
10000	815	778	814	813
100000	8333	8314	8192	8379

TABLE 5.4: Nombre des clients servis par les Web services.

La TAB 5.4 nous nous montre le nombre moyen des clients servis par le système qui ont trouvé le Web service qui cherchent-ils. De la TAB 5.1 et 5.3, nous remarquons que il y a encore un certain nombre de clients qui n'ont pas servis, parce qu'ils attendent des autres Web services dont ils ont besoin.

5.5 Position de notre travail par rapport à l'existant

Dans notre travail, nous avons adopté un modèle de réseau de petri colorée pour le système de Web service, après nous l'avons implémenté sur un simulateur et nous avons calculé le nombre moyen des clients dans le service, le nombre moyen des Web services, le nombre moyen des client servis. dans la TAB 5.5 nous illustrons notre modèle dans l'étude comparative des travaux sur l'évaluation de la performance des Web services. Notre modèle prend en compte deux types d'arrivées, les Web services et les requêtes des clients, et de prendre en considération le faite que les clients sont différents et les Web services sont différent.

		TR1 [25]	TR2 [26]	TR3 [27]	TR4 [28]	TR5 [15]	TR6 [59]	TR7 [60]	TR8 [61]	TR9 [62]	TR10 [63]	Notre mo- dèle
Modèle	-RdPC - RdPS - fille d'attente -RdT	✓		✓			✓	✓	✓	✓	✓	✓
Solution	-solution analy- tique -Simulation -prototype	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
Évaluation de performance	-disponibilité -temps de réponse -coût -nombre moyen -débit -requête de client - temps d'arrivée - TMDS - durée moyenne	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Web service	-simple -composite	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE 5.5: Étude comparative des travaux de l'existence de l'évaluation de performances des Web services

5.6 Conclusion

Notre travail consiste à modéliser un système de Web services avec un réseau de petri coloré. Le modèle proposé prend en compte deux types différents d'arrivées : les Web services et les clients, de la découverte et et la composition des Web services. Nous avons évalué ses performances à l'aide d'un simulateur CPN tools, à partir des résultats obtenues (nombre moyen dans le système, nombre moyen des Web services et nombre moyen des client servis), on constate qu'un système de Web services peut être modéliser par un réseau de petri coloré, et les résultats montrent que les clients il choisissent le Web services dont ils ont besoin.

Conclusion et perspectives

Conclusion générale

Dans ce travail on a traité le problème qualité de services dans la découverte et la composition des Web services. Notre objectif consiste à évaluer les performances d'un système des Web services. Pour répondre à l'objectif, nous avons commencé notre travail par un état de l'art sur les Web services, après on a parlé sur les méthodes d'évaluation de performances qui nous aide à évaluer les performances de système et déterminer sa qualité a partir des résultats obtenus, puis on a donné les notions de base des réseaux de petri, quelques définitions, propriétés et les extensions des RdP, parmi les extensions nous avons choisi les réseaux de petri colorés pour modéliser le système. pour ce faire nous avons proposé un modèle de réseau de petri coloré avec deux stations; la première représente une demande des clients et la seconde une demande des Web services, avec un serveur exponentiel. les arrivées sur le réseau suivent la distribution de poisson, à l'aide de simulateur colored petri net tools(CPN tools) nous avons obtenu le nombre moyen des Web services, le nombre moyen des clients dans les systèmes et le nombre moyen des clients servis, et on à montré qu'un client peut être servi par un Web service dont il est besoin.

Perspectives

Comme perspective nous envisageons un ensemble de travaux futurs :

- Résoudre le modèle de réseau de petri avec une autre méthode.
- Faire l'étude du système de Web services dans le cas de pannes de serveur.

Bibliographie

- [1] **A CHOQUET-GENIET** Les réseaux de Petri un outil de modélisation. Springer-Verlag Berlin Edition 2, 2006
- [2] **M. DIAZ** Les Réseaux de Petri modèles fondamentaux. Hermès science publications paris, 1 édition, 2001.
- [3] **A. BOURJIJ** Contribution à la source de fonctionnement des processus industriels par les réseaux de petri. Thèse doctorat université Nancy 1, 1994.
- [4] **A.IDRISSI** How to minimise the energy consumption immobile ad hoc networks (Université Mohammed V, Maroc 2012) .
- [5] **VINCENT AUGUSTE** support de cours Réseaux de Petri Ecole Nationale des Mines de Saint-Etienne 2012.
- [6] **R.Kara**, " Production " cours Master II Académique, Université de Mouloud Mammeri, Tizi-Ouzou 2011.
- [7] **G.SCORLETTI, G.DINET** et **E.MAGAROTTO** Réseaux de Petri "Cours Master I Académique Université de Caen 2006.
- [8] **R.DAVID, H.ALLA** Grafset et Réseaux de pétri, Hermès 2nd édition 1992.
- [9] **Yann Morère** support de Cours Réseaux de petri. 2002.
- [10] *M^r* **Mohamed Rédha Bahri** Une approche intégrée mobile UML/ Réseaux de petri pour l'analyse des systèmes distribués à base d'agents mobiles. Thèse doctorat université Constantine.

- [11] **Sadok Rezig** Approche canoniques pour les synthèse des contrôleurs réseaux de petri. Thèse doctorat unoversité Lorraine, 2016.
- [12] **René David et Hassane Alla** Du grafcet aux réseaux de petri optimisation des temps d'attente des systèmes flexibles de production basée sur les réseaux de petri.
- [13] **Saggadi Samira** Optimisation des temps d'attente des systèmes flexibles de production basée sur les réseaux de petri. université Boumerdès. Magister 2007.
- [14] **S.Hamaci** Etude des graphes d'évènements temporisés avec multiplicateurs l'algebre (min,+). Thèse doctorat université d'angers, 2005.
- [15] **A. Boushaba, O. Mohammed, R. Benabbou**, Evaluation des performances des protocoles de routage Ad hoc.
- [16] **Alain Pavé** Modélisation des systèmes vivants 200 édition Lavoisier 2012, (64-80)
- [17] **Mohamed S.Obaidat, Nouridine A.Boudriga** Fundamentals of performance evaluation of computer and telecommunication systèmes.
- [18] **A.pavé** Modélisation des systèmes vivants 200 édition Lavoisier 2012, (64-80)
- [19] **Z.Bsri** Introduction au cours outils de modélisation et d'évaluation de performances école des sciences appliquées de Tétouin.
- [20] **Y. son** support cour evaluation de performances Master Informatique université de lorraine.
- [21] **M. Bouali** Contribution à l'analyse formelle et au diagnosyic à partir de réseaux de petri colorés avec l'accessibilité arrière 2010.
- [22] **Y. Mahjoub** Etude des système de transport public et réseaux logistique par les réseaux de petri colorés et l'algebre (max,+) : modélisation, évaluation de performances et optimisation 2019.
- [23] <https://cpntools.org/><https://cpntools.org/> dernière consultation 01/07/2022(site officiel de simulateur)
- [24] site www.daimi.au.dk/cpnets/cpn2000 29/06/2022(site de simulateur)
- [25] **D.Aissani, H.Nacer, N.Bernine**Modélisation d'un système des web services avec les réseaux de petrin et évaluation de ses performances.

- [26] **D.Rathod** Évaluation des performances des Services web restful et services web SOAP / WSDL, Journal international de recherche avancée en informatique Journal international de recherche avancée en informatique, 2017.
- [27] **N.Bernine** Qualité de services dans la découverte et la composition des web services. Thèse de doctorat université Bejaia 2018.
- [28] **W.S. Serrai** Evaluation de performances de solution pour la découverte et la composition des services Web. Thèse doctorat université Paris Est, 2020.
- [29] Support de cours sur les Notions essentielles chapitre 1 .
- [30] **Zhang and Z. Mengchu** A Stochastic Petri Net Approach to Modeling and Analysis of Ad Hoc Network. University Heights.
- [31] **P. Hubert, K. Jensen, R.M. Shapiro**, "Hierarchies in coloured Petri Nets", Advances in Petri Nets, Lecture Notes in Computer Science, vol 483, Springer Verlag, Berlin Heidelberg, New York, 1990.
- [32] **H.HAIOUNI et N.Zaghib** Approche mixte de modélisation. Magister école doctorat en informatique de l'est pôle Constantine., 2010.
- [33] **A. C. Geniet**. les réseaux de Petri : Un outil de modélisation". Spriger-Verlag Berlin Edition 2, 2006.
- [34] <https://www.tutorialspoint.com/webservices/web-services-characteristics.htm> dernière consultation 25/05/2022(chapitre 1)
- [35] <https://fr.wikipedia.org/wiki/Extensible-Markup-Language>. dernière consultation 25/05/2022(chapitre 1)
- [36] <https://www.w3.org/XML/> dernière consultation 25/05/2022(chapitre 1)
- [37] <https://www.ibm.com/docs/en/itcam-app-mgr/7.2.1?topic=s-simple-object-access-protocol-soap>. dernière consultation 25/05/2022 (chapitre 1)
- [38] <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/-Toc478383503>. dernière consultation 25/05/2022 (chapitre 1)
- [39] <https://www.w3.org/standards/xml/core>. dernière consultation 25/05/2022 (chapitre 1).

- [40] <https://www.ibm.com/docs/en/sc-and-ds/stack-web-service-description-language>. dernière consultation 25/05/2022(chapitre 1).
- [41] **S. Dustdar and W. Schreiner**, “A survey on web services composition”, International Journal of Web and Grid Services, vol. 1, number. 1, pp.1 -30, Aug 2005.
- [42] **F. Casati, and M.-C Shan**, “Event-Based Interaction Management for Composite Eservices in eFlow”, Information SystemsFrontiers, 4(1), pp. 19-31, 2002.
- [43] **G. Gardarin**, “XML, des bases de données aux services web (in french)”, Dunod, 28 Nov 2002.
- [44] **K. Fujii and T. Suda**, “Dynamic service composition usingsemantic information”, in Proc. The 2nd international conference on Service orientedcomputing, ACM New York, NY, USA, pp.39-48, 2004.
- [45] **B. Benatallah, M. Dumas, M. C. Fauvet, F. A. Rabhi and Q.Z. Sheng**, “Overview of Some Patterns for Architecting and Managing Composite Web Services”, ACM SIGecom Exchanges, vol.3, number. 3, pp.9-16, 2002.
- [46] **B.Bayant** la théorie des files d’attente : des chaines de Markov aux réseaux à forme produit. Number 22. 2000.
- [47] **P.Erard and P.Déguénon** Simulation par événements discrets. PPUr presses polytechniques; 1996.
- [48] **P.Hastir et B.Jodocy** Les réseaux de petri et leurs applications, mémoire master Université de Namur.
- [49] Support de cours sur les Web services Open classrooms 2016.
- [50] <http://www.dicodunet.com/definitions/normes/service-web.htm> dernière consultation 04/06/2022 (chapitre 1)
- [51] <https://www.ibm.com/docs/fr/radfws/applications-web-services-overview> dernière consultation le 4/06/2022 (chapitre 1).
- [52] **H.kreger et al** Web services conceptual architecture IBM software group,g 5(1) 2007.
- [53] <https://www.oracle.com/fr/cloud/definition-web-service/> dernière consultation 04/06/2022 (chapitre 1).

- [54] <https://www.techno-science.net/glossaire-definition/Service-Web.html>. dernière consultation 04/06/2022 (chapitre 1)
- [55] **S. Hakmi**. Evaluation des Performances des Systèmes Prioritaires à l'aide des Réseaux de Petri Stochastiques Généralisés. Mémoire de Magistère, Université de Béjaia, 2011
- [56] **D. Aissani** "Cours de simulation", Département d'informatique et de recherche opérationnelle, Université de Béjaia, 2001
- [57] **F. Valois** "Introduction à la simulation de réseaux", Technical report, fvalois.insalyon.fr/courses/IRZsimulationv3.pdf, Université de Lyon, 2006
- [58] **M.Ioualalen et A.Aissani** Les symétries dans les réseaux de petri stochastiques (RdPS) construction du graphes symbolique. Rairo Recherche opérationnelle; tome 34, n°2 (2000) p.237-249
- [59] **N. Bernine, H. Nacer, K. Adel, D. Aissani** Simulation et analyse d'un Web service, séminaire mathématique de bejaia (LaMos) volume 13, 2014, pp, 17-20.
- [60] **J. Zick, D. Levy, S. Chen, K. Tang** une évaluation des performances de la sécurité des Web services. Université de Sydney 2016, conférence sur l'information distribuée d'entreprise.
- [61] **N. Bernine, H. Nacer, K. Adel, D. Aissani** Évaluation des Performances d'une Architecture pour la Découverte et la Composition des Web Services, séminaire mathématique de bejaia (LaMos) volume 12, 2013, pp, 131-138.
- [62] **W.Ait Cheik Bihi**, approche orientée modèles pour la vérification et l'évaluation de performances de l'interopérabilité et l'interaction des services Web, thèse doctorat 2012. Université Belfort .
- [63] **N. Bernine, D. Aissani**, Un Modèle pour l'évaluation des Performances d'un Système de Web Services, séminaire mathématique de bejaia (LaMos) volume 14, 2015, pp, 3-7.

Annexes

A.1 CPN TOOLS

A.1.1 Histoire

CPN Tools est destiné à remplacer Design/CPN qui est un progiciel répandu pour les CP-nets, Design /CPN a été publié pour la première fois en 1989 avec un support pour l'édition et la simulation de réseaux CP. CPN Tools est le résultat d'un projet de recherche, le projet CPN a été développé , à l'Université d'Aarhus de 2000 à 2010[24]. Les principaux architectes derrière l'outil sont *Kurt Jensen, Søren Christensen, Lars M. Kristensen et Michael Westergaard*. À partir de l'automne 2010, CPN Tools est transféré au groupe AIS, Université de technologie d'Eindhoven, Pays-Bas. L'objectif du projet CPN était de profiter de l'évolution de l'interaction homme-machine, et d'expérimenter ces techniques dans le cadre d'une refonte complète de l'IHM pour Design/CPN. La dernière version est apparue en 2015.

A.1.2 Définition (CPN TOOLS)

CPN Tools est un outil d'édition, de simulation et d'analyse de réseaux de Petri colorés. L'outil propose une vérification incrémentielle de la syntaxe et la génération de code, qui ont lieu pendant la construction d'un réseau. Un simulateur rapide gère efficacement les filets non chronométrés et chronométrés[23]. Des espaces d'état complets et partiels peuvent être générés et analysés, et un rapport d'espace d'état standard contient des informations, telles que les propriétés de limite et les propriétés de vivacité.

A.1.3 L'interface de CPN tools.

La colonne de gauche s'appelle l'index et le reste de l'interface c'est l'espace travail. Comme le montre l'image ci-dessous.

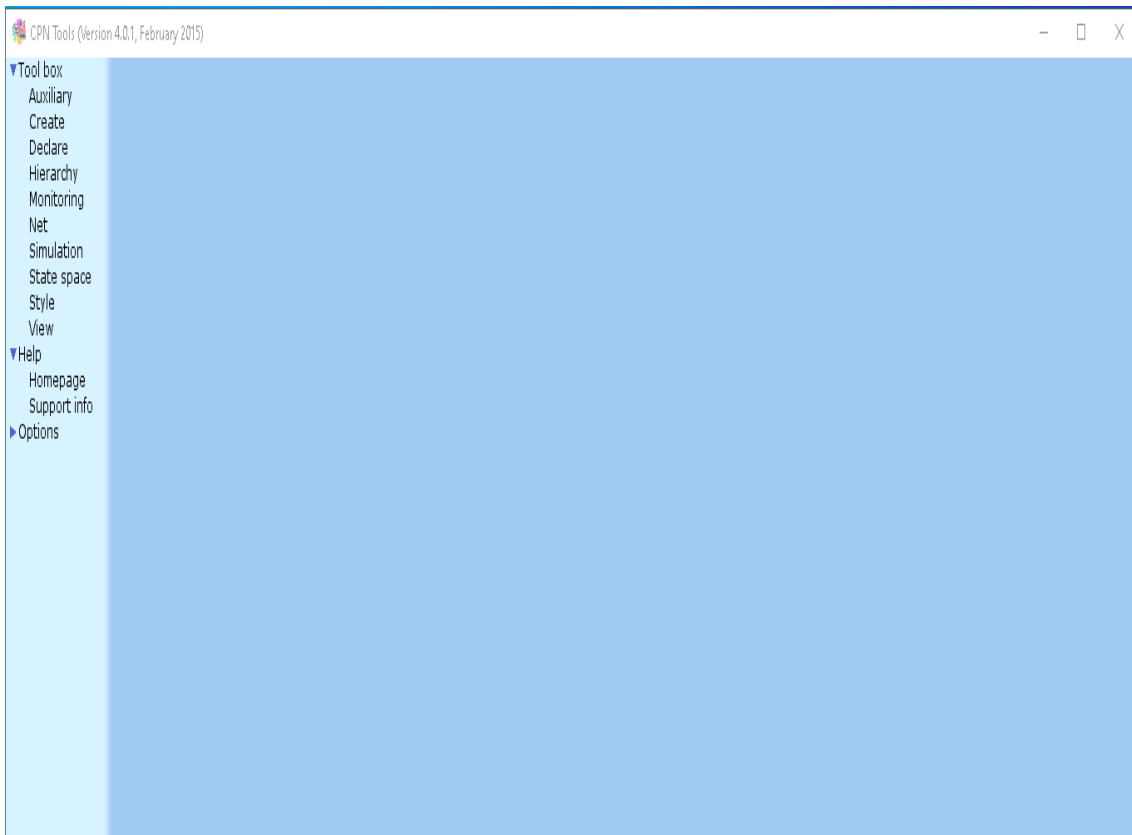


FIGURE A.1: interface de CPN TOOLS

La colonne de gauche (index) contient une zone qui s'appelle **outil** .



FIGURE A.2: la boîte à outils d'index

Elle même contient une liste des **Palettes** disponibles create(crée), simulation,... . Pour accéder à une palette d'outils dans l'index faire glisser une palette vers l'espace travail comme la montre la FIG A.3.

Dans l'image FIG A.4 on vas voir tous les palette qui sont disponible dans Outil dans l'espace travail.

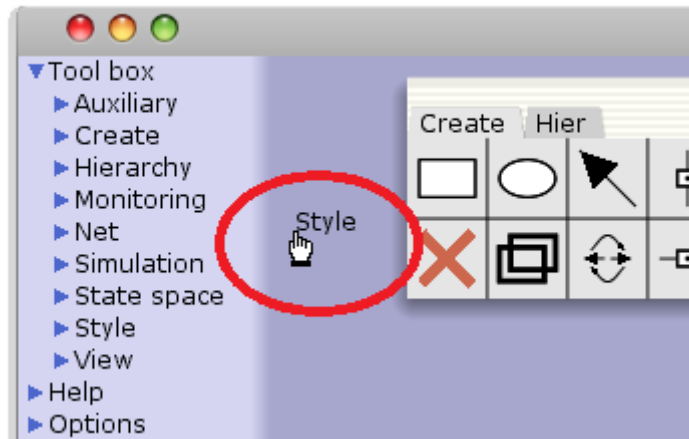


FIGURE A.3: Palettes d'outils dans l'espace travail

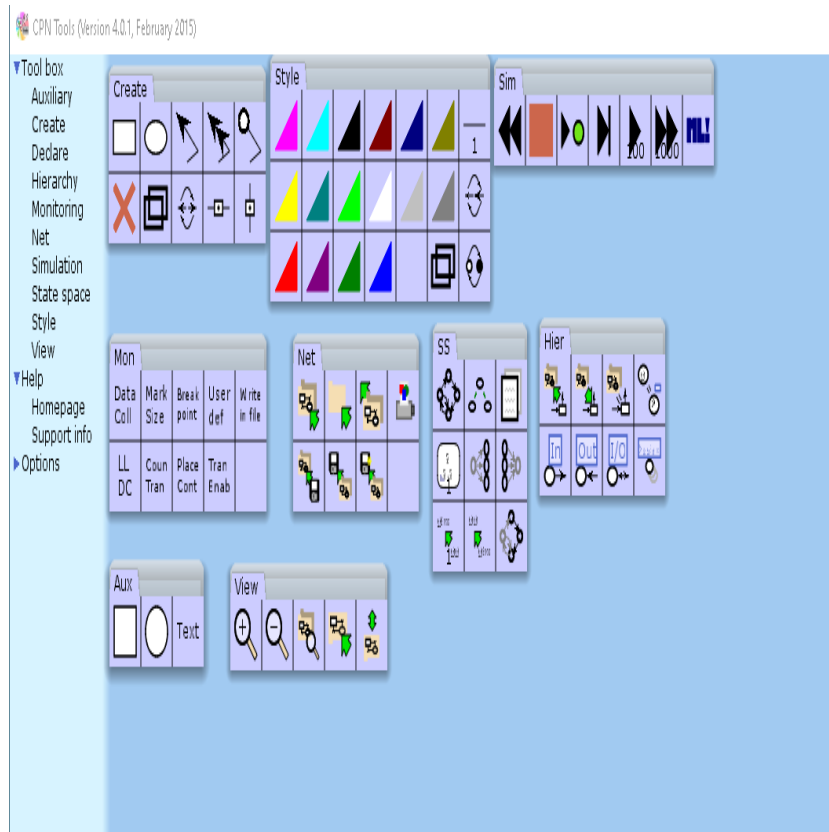


FIGURE A.4: Toutes les palettes dans l'espace travail

les palettes

- Outils auxiliaires : sont utilisés lors de la création d'éléments auxiliaires.



FIGURE A.5: Outils auxiliaires

- Outils de création : sont utilisés lors de la création de la structure de réseau.



FIGURE A.6: Outils de création

- Outils de hiérarchie sont utilisés pour modifier la structure hiérarchique de réseau.

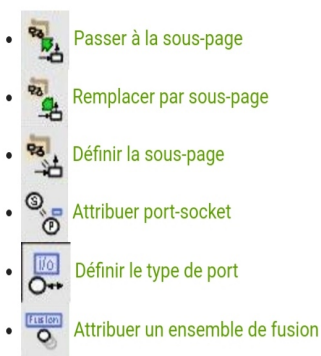


FIGURE A.7: Outils hiérarchie

- Outils net sont utilisés pour charger et enregistrer des réseaux.



FIGURE A.8: Outils net

- Outils de simulation sont utilisés pour simuler le réseau.



FIGURE A.9: Outils de simulation

- Outils d'espace d'état sont utilisés par exemple pour calculer des espace d'état.



FIGURE A.10: Outils d'espace d'état

..