



**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM**

**Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique**

**Mémoire De Fin d'Etudes
Pour l'Obtention Du Diplôme de Master En Informatique
Option : Ingénierie des Systèmes d'Information
Thème**

La vérification formelle de diagrammes UML

Etudiante :

KADDOUR SEGHIR Leila

Encadrant:

LAREDJ Mohamed Adnane

Année Universitaire 2015/2016

Résumé

La technologie de workflow, tendant à automatiser les processus métier et à fournir un support pour leur gestion. Cependant l'exactitude de leur modélisation influe sur la performance de l'entreprise. Dans ce mémoire, nous allons utiliser les diagrammes d'activités de langage UML pour modéliser ces processus workflow. Ils sont faciles et compréhensibles par tous les acteurs de l'entreprise impliqués dans cette modélisation, mais souffrent d'un manque de formalisme qui peut produire des incohérences et un dysfonctionnement des processus modélisés, et ne permettent pas d'autre part d'effectuer une analyse et une vérification, pour cela nous proposons une transformation des diagrammes d'activités d'UML vers leurs équivalents dans un modèle formel tel que les réseaux de Pétri, qui représente un outil formel pouvant être utilisé comme une base pour une éventuelle vérification et analyse afin de garantir la cohérence et le bon fonctionnement d'un workflow.

Mots clés : processus métier, workflow, UML, diagrammes d'activités, réseaux de Pétri, transformation de graphe.

À

Mes Parents

Mes Sœurs

Mes Amies

Remerciements

Je tiens à remercier mon encadreur Monsieur LAREDJ Mohamed Adnane pour m'avoir guidé et pour la confiance et les conseils avisés qu'il m'a accordés pendant ce travail.

Je remercie mes enseignants qui ont contribué à mon éducation et mon enseignement, j'espère être à la hauteur de leur espérance.

Je tiens à remercier également tous ceux qui m'ont soutenu et encouragé de près ou de loin tout au long de ce travail.

Sommaire

Introduction générale	1
Chapitre 1 : Définitions et concepts de base	
I. Introduction	3
II. Définitions et concepts de bases	3
II.1. Le Système d'Information (SI)	3
II.2. La collaboration	3
II.3. La coopération	4
II.4. La coordination	4
II.5. La communication	4
II.6. Le Système d'Information collaboratif (SIC)	4
III. Le processus métier	5
IV. Le groupware.....	5
V. Le workflow	5
V.1. Domaines d'application des workflows	6
V.2. Typologies des applications workflow	6
V.2.1. Workflows administratifs	6
V.2.2. Workflows de production	6
V.2.3. Workflows collaboratifs	6
V.2.4. Workflows ad-hoc	6
V.3. Le système de gestion de workflow (SGWf)	6
V.4. Le modèle de référence du workflow	7
VI. La modélisation des Workflows.....	8
VII. Conclusion	9
Chapitre 2 : La modélisation des Diagrammes d'Activité d'UML	
I. Introduction.....	10
II. Le langage UML	10
II.1. Diagrammes structurels ou statiques	11
II.2. Diagrammes comportementaux ou dynamiques.....	11
III. Les diagrammes d'activités d'UML	12
III.1. Intérêts des diagrammes d'activité	12
III.2. Notation	13
III.2.1. Les nœuds.....	13
III.2.1.1. Nœud d'activité.....	13
A. Nœud d'objet	13
B. Nœud de contrôle	14
C. Nœud exécutable	15
III.2.1.2. Une partition d'activité	16

III.2.1.3. Une région d'expansion	16
III.2.2. Les arcs.....	17
III.2.2.1. Flot de contrôle (control flow)	17
III.2.2.2. Flot d'objet (object flow)	17
III.2.2.3. Un Handler d'exception	17
IV. Conclusion.....	18
Chapitre 3 : Les réseaux de Pétri et la transformation des modèles	
I. Introduction.....	20
II. La vérification formelle de diagrammes UML	20
III. Les réseaux de Pétri.....	20
III.1. Le marquage	21
III.2. Représentation matricielle.....	21
III.3. Fonctionnement des réseaux de Pétri	22
III.4. Réseaux de Pétri particuliers	23
III.4.1. Les structures particulières.....	23
III.4.1.1. Graphe d'états	23
III.4.1.2. Graphe d'évènement (Graphe de transitions)	23
III.4.1.3. RdP sans conflit	24
III.4.1.4. RdP à choix libre.....	24
III.4.1.5. RdP simple	24
III.4.1.6. RdP pur	25
III.4.2. Schémas particuliers de RdP	25
III.4.2.1. Parallélisme	25
III.4.2.2. Synchronisation.....	26
III.5. Les propriétés des réseaux de Pétri	26
III.5.1. Caractère Borné.....	27
III.5.2. Pseudo-vivacité	27
III.5.3. Quasi-vivacité	27
III.5.4. Vivacité	27
III.5.5. Absence de blocage.....	27
III.5.6. Ré-initialisabilité	27
IV. Les méthodes d'analyse.....	29
IV.1. Le graphe des marquages accessibles	30
IV.1.1. L'arbre / arborescence & le graphe de couverture.....	31
V. La transformation des diagrammes d'activités d'UML vers les réseaux de Pétri	32
V.1. Transformation de Modèle	32
V.2. Nœud initial.....	32
V.3. Nœud final d'activité.....	32
V.4. Nœud final de flux.....	33
V.5. Nœud de décision	33
V.6. Nœud de fusion.....	33
V.7. Nœud de bifurcation.....	33
V.8. Nœud d'union.....	33

V.9. Nœud d'action	33
V.10. Nœud d'objet	34
V.11. Nœud d'envoi de signal	34
V.12. Nœud de réception de signal et le nœud de temps d'attente	34
V.13. Arc de flux de contrôle	35
V.14. Arc de flux d'objet.....	35
VI. Conclusion.....	35
Chapitre 4 : Implémentation	
I. Introduction.....	37
II. Exemple d'application	37
II.1. Le client	37
II.2. Le service comptable	37
II.3. Le service de livraison	37
III. Les outils de développement.....	41
III.1. Langage de programmation C#	41
III.2. Framework .NET.....	41
III.3 L'environnement de développement	41
IV. Le logiciel « UML2RdP »	41
V. Description des interfaces du logiciel.....	42
V.1. L'accueil	42
V.2.La deuxième interface (fenêtre de modélisation de D.A d'UML)	42
V.2.1. Ruban du logiciel	43
V.2.1.1. Les menus.....	43
V.2.1.2. La barre d'outils	44
V.2.2. La modélisation du dessin	45
V.2.2.1. Paramétrage des objets	46
V.2.3. Le déplacement des objets.....	46
V.3. La troisième interface (fenêtre de RdP)	46
V.3.1. Simulation du réseau	50
Conclusion générale et perspectives.....	52
Bibliographie	55

Liste des Figures

Figure.1.1. Modèle de référence du workflow	8
Figure.2.1. Notation d'un nœud d'objet	13
Figure.2.2. Représentation des pins d'entrée et de sortie sur une activité	13
Figure.2.3. Notations de nœud initial et les deux types de nœuds finaux	14
Figure.2.4. Notation de nœud de décision	14
Figure.2.5. Notation de nœud de fusion	15
Figure.2.6. Notation de nœud de bifurcation	15
Figure.2.7. Notation de nœud d'union	15
Figure.2.8. Notation de nœud d'action	15
Figure.2.9. Notations des nœuds exécutables	16
Figure.2.10. Notation de la partition	16
Figure.2.11. Notation d'une région d'expansion	16
Figure.2.12. Notation d'un flot de contrôle	17
Figure.2.13. Les deux notations d'un flot d'objet	17
Figure.2.14. Les deux notations d'un Handler	17
Figure.2.15. Exemple d'un diagramme d'activité	18
Figure.3.1. Représentation graphique d'un réseau de Pétri	21
Figure.3.2. Exemple d'un réseau de Pétri marqué avec son vecteur de marquage M_0	21
Figure.3.3. Un RdP et ses matrices de Pré-condition, Post-condition et C	22
Figure.3.4. Evolution du marquage	22
Figure.3.4. Exemple de graphe d'état	23
Figure.3.5. Exemple de graphe d'évènement	23
Figure.3.6. Exemple de graphe sans et avec conflit	24
Figure.3.7. Exemple de graphe à choix libre / non pas à choix libre	24
Figure.3.8. RdP simple / non simple	25
Figure.3.9. RdP pur / impur	25
Figure.3.10. Parallélisme	26
Figure.3.11. Synchronisation	26
Figure.3.12. RdP marqué et borné	30
Figure.3.13. Graphe des Marquages accessibles	30
Figure.3.14. Exemple d'un RdP non borné	31
Figure.3.15. Graphe de couverture	31
Figure.3.16. La notation de RdP équivalente au nœud initial	32
Figure.3.17. La notation de RdP équivalente au nœud final d'activité	32
Figure.3.18. La notation de RdP équivalente au nœud final de flux	32
Figure.3.19. La notation de RdP équivalente au nœud de décision	33
Figure.3.20. La notation de RdP équivalente au nœud de fusion	33
Figure.3.21. La notation de RdP équivalente au nœud de bifurcation	33
Figure.3.22. La notation de RdP équivalente au nœud d'union	33
Figure.3.23. La notation de RdP équivalente au nœud d'action	34

Figure.3.24. La notation de RdP équivalente au nœud d'objet	34
Figure.3.25. La notation de RdP équivalente au nœud d'envoi de signal	34
Figure.3.26. La notation de RdP équivalente au nœud de réception de signal et de temps d'attente	35
Figure.3.27. La notation de RdP équivalente à un arc de flux de contrôle.....	35
Figure.3.28. La notation de RdP équivalente à un arc de flux d'objet	35
Figure.4.1. Exemple d'application de gestion d'une commande de produits.....	39
Figure.4.2. Le Réseau de Pétri obtenu après la transformation correspond à l'exemple d'application	40
Figure.4.3. L'interface d'accueil du logiciel	42
Figure.4.4. L'icône du logiciel	42
Figure.4.5. L'interface de modélisation des diagrammes d'activité d'UML du logiciel	43
Figure.4.6. Le menu Fichier	43
Figure.4.7. Le menu Exporter.....	44
Figure.4.8. Le menu Aide	44
Figure.4.9. Le premier et le deuxième groupe du ruban du logiciel.....	44
Figure.4.10. Le troisième groupe du ruban du logiciel	45
Figure.4.11. Le paramétrage des objets	46
Figure.4.12. L'interface du RdP	47
Figure.4.13. Les propriétés du RdP	48
Figure.4.14. La structure du RdP -1-	48
Figure.4.15. La structure du RdP -2-	49
Figure.4.16. La représentation matricielle du RdP	49
Figure.4.17. Le graphe de marquages accessibles du RdP	50
Figure.4.18. La simulation du RdP	50

Liste des Tableaux

Tableau.3.1. Exemples des propriétés des RdP	29
Tableau.1. Ensembles des problèmes rencontrés lors du passage du D.A vers le RdP.....	53

Introduction générale

Depuis son invention, l'informatique a connu une grande évolution dans tous les secteurs. Avec cette évolution, et dans le monde des entreprises où l'information est au cœur de ces performances, l'enjeu, sans nul doute, le plus important est de construire le parfait système d'information, car celui-ci est le centre de fonctionnement de tous types d'entreprises, son efficacité est un élément-clé pour la survie et la réussite de ces entreprises. Pour cela, des transformations majeures aux plans humains, organisationnels et technologiques sont obligatoires, et donc la notion de système d'information collaboratif fait son entrée aux entreprises pour assurer la cohésion et l'interopérabilité entre ses partenaires potentiels.

Dans de tel système d'information, les paramètres de collaboration, de coopération et de coordination, en plus de la communication, sont toujours présent.

À partir de toutes de tous ça, nous arrivons à comprendre que le travail collaboratif est l'ensemble des activités réalisés par les acteurs d'une entreprise afin d'aboutir un but métier, sont alors introduits sous le nom d'un processus métier. Certaines activités de ce processus métier peuvent être *automatisées* dans un but de réduire le temps, de répondre mieux aux besoins des clients, de simplifier le travail, d'éviter les erreurs humaines et d'améliorer la productivité de l'entreprise, et par conséquent le gain sera élevé. Et donc de nouvelles technologies sont alors apparu afin d'assurer cette automatisation de ces processus, telle que les workflows qui sont des applications particulières des groupwares. Dans ce contexte, nous avons besoin de la TCAO (*Travail Coopératif Assisté par Ordinateur*) qui nous fournit l'outil informatique qui est considéré comme une matière première dans les entreprises.

La modélisation des processus se fait à l'aide des langages ou des méthodes de modélisations unifiée compréhensible par tous les acteurs de l'équipe. Il existe plusieurs méthodes de modélisation des processus workflows, telle que BPMN (*Business Process Modeling Notation*), OSSAD (*Office Support Systems Analysis and Design*) et le langage UML (*Unified Modeling Language*). C'est sur ce dernier que va porter notre travail de recherche et en particulier les diagrammes d'activités.

Malgré qu'UML est un langage riche, doté d'une notation ouverte et largement utilisée, les diagrammes d'UML ont besoin d'être vérifiés et analysés pour s'assurer que le comportement spécifié dans les diagrammes est correct, et de garantir le bon fonctionnement de l'ensemble des processus de l'entreprise. Ce fait est dû qu'UML est un langage semi-formelle, et donc sa sémantique qui est assez faible et l'ambiguïté persistante sont les grands problèmes que souffre ce genre de langage de modélisation.

Notre démarche aura pour principal objectif l'analyse et la vérification formelle des diagrammes d'activités d'UML en utilisant les Réseaux de Pétri. Cet outil mathématique

nous permet d'assurer la satisfaction de certaines propriétés des processus modélisés, et nous offrir en plus leur simulation. Pour cette démarche, un besoin de transformation des diagrammes d'activités d'UML vers les réseaux de Pétri est obligatoire, cela se fait à l'aide d'un ensemble de règles de transformation prédéfinis.

Pour notre projet, on a travaillé, en premier lieu, sur la modélisation des processus, en utilisons les Diagrammes d'Activité d'UML dans un but de faciliter cette tâche aux utilisateurs, en suite, réaliser une transformation d'une façon automatique afin de passer du premier modèle au modèle cible qui est le Réseau de Pétri, on se basant sur un ensemble de règles de transformation. À la fin, les modèles obtenus de la transformation sont analysés et vérifiés formellement afin de valider leurs bons fonctionnements.

Notre document est organisé en quatre chapitres :

- ❖ Le premier chapitre (Définitions et concepts de base) : Dans ce chapitre, nous allons définir les différentes notions de base concernant notre thème ; comme le système d'information, le travail collaboratif...etc. Nous allons introduire à la fin du chapitre à la modélisation des workflows.
- ❖ Le deuxième chapitre (La modélisation des Diagrammes d'Activité d'UML) : Dans ce chapitre, nous allons entamer la modélisation des Diagrammes d'Activité d'UML, et on terminera le chapitre par un exemple qui représente les principaux nœuds du D.A.
- ❖ Le troisième chapitre (Les Réseaux de Pétri et la transformation des modèles) : Dans ce chapitre, nous allons étudier les Réseaux de Pétri, leurs fonctionnements, leurs schémas particuliers, leurs propriétés et quelques méthodes d'analyse. Et nous allons voir aussi les règles de transformation du Diagramme d'Activité vers le Réseau de Pétri.
- ❖ Le quatrième chapitre (Implémentation) : Dans ce chapitre, nous allons décrire notre application nommé « UML2RdP » que nous avons réalisé.

Chapitre 1

Définitions et concepts de base

I. Introduction :

La place de l'information dans les organisations a peu à peu été reconnue comme centrale. Pour cela, disposer d'un système d'information efficace, constitue un enjeu principal pour ces organisations. Ce système d'information doit garantir le travail collaboratif et la communication entre les participants, on parle donc du système d'information collaboratif. En effet, toute organisation, quels que soient sa vocation et ses objectifs, est appelée à gérer des processus spécifiques à ses fonctions et à ses compétences. Ces processus sont donc appelés les processus métier. Ces derniers sont automatisés grâce aux nouvelles technologies de l'information telles que les groupewares, les outils de la TCAO (*Travail Coopératif Assister par Ordinateur*) ou les workflows.

La modélisation de l'ensemble des processus métier de l'organisation est considéré comme une phase primordiale, car elle permet d'améliorer leurs gestions et d'avoir une meilleure vue de leurs fonctionnements. Dans ce premier chapitre, nous rappelons les principaux concepts de bases et définitions du système d'information, du travail collaboratif et coopératif, ...etc. On exposera ensuite les processus métier, les groupewares et les workflows et d'autres définitions relatives à ce dernier. À la fin du présent chapitre, nous introduirons la notion de la modélisation des workflows.

II. Définitions et concepts de bases :

Afin de définir le *système d'information collaboratif*, il est impératif de définir la notion de *système d'information* qui est le centre nerveux de tout type d'entreprise, et les quatre paramètres essentiels, qui sont : la collaboration, la coopération, la coordination et la communication.

II.1. Le Système d'Information (SI) :

Le Système d'Information (noté SI), est défini selon (Robert Reix), comme : « *Un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures... permettant d'acquérir, de traiter, de stocker des informations (sous formes de données, textes, images, sons, etc.) dans et entre des organisations.* » [PRI 06]

II.2. La collaboration :

Une collaboration « *est un travail en commun ; un travail entre plusieurs personnes qui génère la création d'une œuvre commune* » (le petit Robert) [CRO 06], où chaque intervenant apporte son savoir, son idée en pouvant ou non s'inspirer des apports réalisés précédemment par les autres. [POT 07]

La notion de collaboration quant à elle implique qu'il n'y a aucune répartition des tâches entre les collaborateurs. En effet ces derniers travaillent ensemble à chaque étape du travail. Le travail final réalisé de façon collaborative ne permet pas de distinguer les tâches accomplies par l'un ou l'autre des collaborateurs.

Le travail collaboratif *« permet d'échanger et de partager des compétences pour mieux réussir un projet commun »* (Office Québécois de la Langue Française). [CRO 06] Donc le travail collaboratif nécessite ainsi des membres d'une équipe une plus forte interactivité et requiert plus de motivation et de confiance interpersonnelle. [PIQ 09]

II.3. La coopération :

La coopération est une *« action collective par laquelle des sujets contribuent au même résultat »* (De Terssac et Maggi, 1996). [LAL 10]

« Le travail coopératif est un travail de groupe hiérarchiquement organisé qui fonctionne suivant un planning impliquant des délais et une division des tâches (c'est-à-dire chaque collaborateur réalisant une partie du travail). Chaque intervenant (collaborateur) sait ce qu'il doit faire dès le début et communique, échange ou partage des éléments uniquement pour arriver à son objectif individuel. A la fin, le travail de chacun est relié pour créer un objet unique de travail. » [VIG 10]

II.4. La coordination :

Selon Maggi (1997) la coordination n'est que l'ensemble des règles d'action qui structure la coopération. [VIG 10]

II.5. La communication :

La communication représente un élément essentiel de la coopération, de la collaboration et de la coordination. Cependant, comme le soulignent Gronier, et al. (2004), la particularité de la communication est qu'elle n'est pas une finalité en soi, mais un moyen pour atteindre un but. [VIG 10] Elle consiste en l'échange d'information entre les éléments du groupe ou entre les groupes. [SIN 13]

II.6. Le Système d'information collaboratif (SIC) :

Le *Système d'Information Collaboratif* (noté SIC) constitue un système d'information inter-organisationnel susceptible de supporter les processus collaboratifs des partenaires. Le principe de cette option est de faire jouer au SIC le rôle de système intermédiaire assurant la cohésion entre les partenaires potentiels. Le SIC devient alors le garant de l'*interopérabilité* des SI des partenaires. [TOU 07]

Un système d'information peut être décomposé en plusieurs sous systèmes d'information. Dans ce cas, les systèmes d'information collaboratifs internes à l'entreprise permettent l'interopérabilité des différents systèmes d'information d'une même entreprise. [SOU 13]

Et alors, le travail collaboratif est nécessaire dans les organisations, il a incité les entreprises à repenser et à introduire l'approche des *processus* dans leurs systèmes d'information. Dans la partie qui suit nous allons définir les notions de processus et processus métier.

III. Le processus métier:

Le terme *processus* vient du latin et signifie littéralement « aller de l'avant ». Dans un usage général, il évoque l'idée d'une marche progressive, souvent selon un plan déterminé à l'avance. [SAA 06] Et alors, un *processus métier* (en anglais *Business Process*) est un enchaînement d'activités à exécuter par des acteurs jouant des rôles particuliers, consommant et produisant des ressources pour réaliser un objectif de l'entreprise. Les activités peuvent être déclenchées par des événements et peuvent à leur tour produire des événements. [BEN 10] [ULM 11]

L'automatisation d'une partie ou de la totalité d'un processus métier, résulte en un workflow qui est une application particulière du groupware.

IV. Le groupware :

On distingue par le terme *groupware* (en français Collectique) l'ensemble de méthodes et de techniques de travail en équipe, plus les outils logiciels (appelés collecticiels) conçus pour améliorer le mécanisme de communication, de coopération et de coordination spécifiques aux processus de travail. [SIN 13] Ces outils logiciels sont sur lesquels s'appuie la mise en œuvre du domaine TCAO (*Travail Coopératif Assisté par Ordinateur*) en anglais CSCW (*Computer Supported Cooperative Work*) qui a été inventé par Irene Grief au milieu des années 80 pour un objectif d'étudier le rôle de l'outil informatique dans le travail de groupe, quel que soit le type de travail réalisé, et d'aider les utilisateurs à exploiter au mieux les ordinateurs dans leur travail quotidien. [BOUA 12] Ce système permet à plusieurs utilisateurs, séparés ou réunis par le temps et par l'espace, de travailler ensemble par le biais d'une infrastructure informatique.

Le groupware renvoie à des applications diverses et variées, parmi celles les workflows qui sont l'un des outils de la TCAO, que nous allons entamer par la suite.

V. Le workflow :

Un *workflow* est traduit littéralement en français par « flux de travail ». De nombreuses définitions existent pour le workflow, cependant, la plus répandue est celle donnée par le WfMC (*Workflow Management Coalition*), cette organisation internationale, créée en 1993, regroupe des éditeurs, des utilisateurs et des experts dans le domaine du workflow. [AMA 02]

La définition est : « *Un workflow est l'automatisation partielle ou complète d'un processus, au cours duquel des documents, des informations, des tâches passent d'un participant à un autre, au sein d'un groupe de travail, en conformité avec un ensemble de règles* » (WfMC, 1999). [ZEF 12]

Autrement dit, un workflow est la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et les différents acteurs impliqués dans la réalisation d'un processus métier. [PIQ 09] La mise en place d'un workflow, permet de : [BOUA 12]

- ❖ traiter plus rapidement et à moindre coût les différents processus dans l'entreprise ;
- ❖ décrire les délais à respecter ;
- ❖ déterminer les rôles tenus par chaque acteur et lui fournir les informations nécessaires pour la réalisation de sa tâche.

V.1. Domaines d'application des workflows :

- ❖ Informatique : développement de logiciel;
- ❖ Finance : les systèmes bancaires, les assurances;
- ❖ Médical : suivi du dossier médical d'un patient, planification des opérations chirurgicales.

V.2. Typologies des applications workflows :

Différentes classifications ont été proposées pour les applications workflows. Ces classifications dépendent des objectifs et des besoins de l'entreprise.

V.2.1. Workflows administratifs : ces types de workflows gèrent les processus administratifs répétitifs et prédictibles qui suivent une procédure bien définie, par exemple : demande d'achat, demande de billet pour le voyage, congés,... [AMA 02]

V.2.2. Workflows de production : concernent les processus métiers répétitifs et prédictibles fortement structurés qui implémentent le noyau du métier de l'entreprise et comprennent dans leurs activités, par exemple : instruction de demandes de prêts bancaires, traitement des réclamations déposées par les compagnies d'assurances,... [BEN 10]

V.2.3. Workflows collaboratifs : ces types de workflows se concentrent sur le travail d'équipe en vue d'atteindre des objectifs communs, ils gèrent aussi les processus qui évoluent assez fréquemment. L'utilisation des workflows collaboratifs a pour but de faciliter la communication inter-groupes, par exemple : la préparation d'une documentation de produits, la rédaction collective d'un rapport d'expertise,... [LAR 12]

V.2.4. Workflows ad-hoc : dédiés à la gestion des processus de base dans une entreprise. Le support workflow est, dans ce cas, limité à fournir la communication, à router les données entre les participants, et gérer les accès et états, par exemple : la planification de réunions,... [BOUA 12] [BEN 10]

V.3. Le système de gestion de workflow (SGWf) : [LAR 12]

Un système de gestion de workflow (en anglais *Workflow Management System*, noté WfMS) est un système informatique dédié à la gestion des processus métier. C'est un système qui sert à définir, implémenter et gérer l'exécution d'un ou plusieurs workflows à l'aide d'un environnement logiciel, fonctionnant avec un ou plusieurs moteurs de workflow et capable d'interpréter la définition des processus, d'interagir avec les participants du workflow et d'appeler des outils informatiques et des applications externes.

Remarque : Pour le « moteur de workflow », c'est le cœur du système de gestion de workflow, constituant un outil précieux pour formaliser et automatiser les processus.

V.4. Le modèle de référence du workflow : [LAR 12] [BOUA 12]

Il a pour objectif de résoudre les problèmes d'interopérabilité entre les différents produits de workflow lors de leur communication en apportant un cadre commun sur lequel les éditeurs de solutions peuvent se baser. L'association WfMC (*Workflow Management Coalition*) a développé un *modèle de référence* pour les systèmes de gestion de workflow. La Figure.1 illustre son architecture qui est constituée de cinq interfaces.

- ❖ **Outils de définition des processus (interface 1) :** c'est l'outil permettant la description et la modélisation graphique sous une notation existante (voir le paragraphe VI) ou propriétaire du processus à automatiser.
- ❖ **Applications clientes workflow (interface 2) :** ces applications permettent la communication entre le système de workflow et l'utilisateur.
- ❖ **Applications invoquées ou appelées (interface 3) :** cette interface permet aux systèmes de gestion workflows de communiquer avec toutes les applications externes nécessaires à l'accomplissement de tâches.
- ❖ **Autres services d'exécution workflow (interface 4) :** cette interface permet de communiquer et d'échanger les informations avec d'autres systèmes de gestion de workflow.
- ❖ **Outils de contrôle et d'administration (interface 5) :** les outils de cette interface donnent accès aux tâches d'administration, telles que la suspension d'une tâche ou l'arrêt d'un processus, l'enregistrement de l'avancement des cas, l'allocation des ressources. En outre, ces outils sont utilisés pour surveiller et contrôler le workflow, par exemple la détection des exceptions et la gestion des anomalies,...etc.
- ❖ **Moteur de workflow :** c'est le centre nerveux d'un système de gestion de workflow (SGWf). Il correspond à un environnement d'exécution d'un ou plusieurs workflows. Ce moteur de workflow, c'est grâce aux cinq interfaces du modèle de référence de SGWf qu'il permet de communiquer avec les autres composants de SGWf et avec les autres SGWfs.
- ❖ La partie « API de workflow et format d'échange » s'occupe de la gestion des connexions externes avec d'autres applications ou d'autres services. [TOU 07]
- ❖ La partie « services d'exécution workflow » assure la gestion des modèles workflows réalisés. [TOU 07]

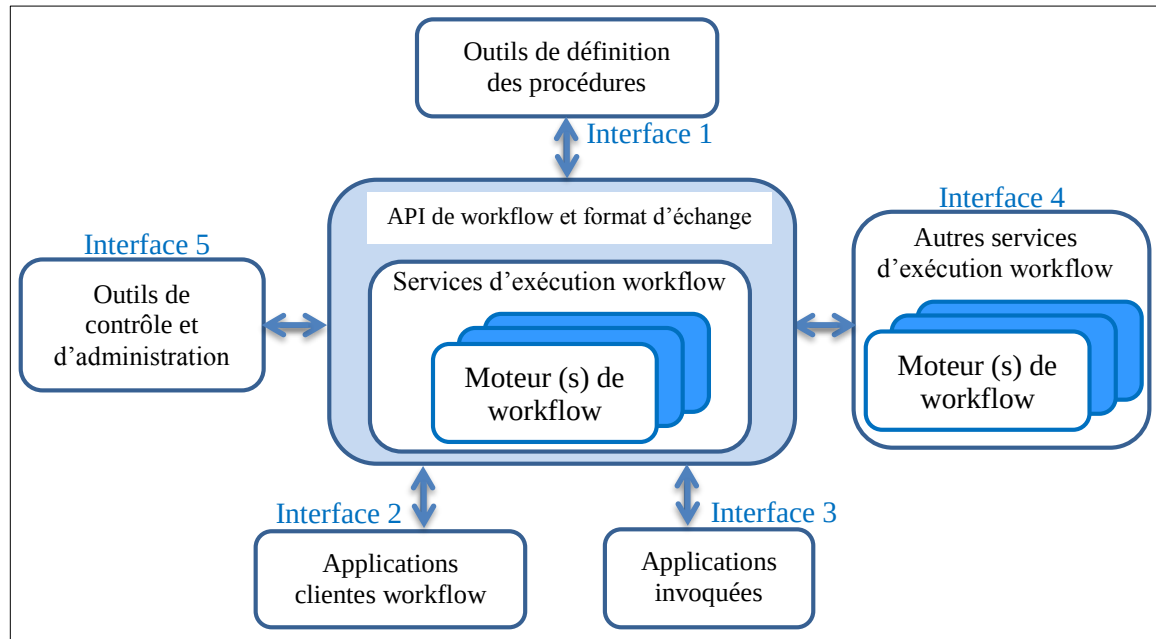


Figure.1.1.Modèle de référence du workflow. [ZEF 12]

Ce modèle de référence offre un élément essentiel pour les workflows qui la modélisation, c'est grâce aux outils de leur première interface comme nous avons mentionné auparavant, et donc les questions qui se posent : pourquoi on a besoin de cette modélisation ? Et comment modéliser les processus workflows ?

VI. La modélisation des Workflows :

En général, modéliser un système avant sa réalisation, permet de mieux comprendre leur fonctionnement. C'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence.

Un modèle est un langage commun, précis, qui est connu par tous les membres de l'équipe. Un modèle est donc une représentation simplifiée consensuelle d'une partie du monde réel, exprimée dans un langage de représentation dans un but défini. [BEN 10]

Il existe plusieurs méthodes de modélisation des workflows, parmi lesquelles on peut citer : BPMN, OSSAD et le langage UML et aussi les réseaux de Pétri.

La méthode BPMN (*Business Process Modeling Notation*) est une notation graphique standardisée pour modéliser des processus métier. Cette norme ne fait pas de distinction entre des processus humains et informatiques, son but est de fournir une notation qui soit réellement compréhensible par tous les utilisateurs de l'entreprise. [HAMA 13]

La méthode OSSAD (*Office Support Systems Analysis and Design*) est une méthode de modélisation du système d'informations et d'organisation ouverte et standard, cette méthode a pour but de gérer les problèmes organisationnels amenés par l'arrivée massive de la technologie dans les bureaux (bureautique). [GLA 02]

La méthode qui se base sur le langage UML (*Unified Modeling Language*) est une méthode généralement la plus utilisée pour modéliser les processus de workflow, elle présente beaucoup d'avantages par rapport aux autres méthodes en intégrant plusieurs types de diagrammes. Dans notre travail nous avons choisi d'adopter ce langage, et bien précisément leurs diagrammes d'activités, car celui-ci est un langage à une notation claire et facile à comprendre et à représenter par tous les membres de l'équipe de travail.

On trouve plusieurs outils de modélisation des workflows qui utilisent ces notations standards, nous citons :

- ❖ Modelio : un outil open source de modélisation des processus, il support les standards UML2 (les diagrammes d'activités, et d'autres diagrammes) et BPMN2.
- ❖ Bonita BPM : un outil open source de modélisation des processus métier par l'utilisation de la notation BPMN2.
- ❖ Workey Designer : outil de modélisation des processus qui utilise la notation OSSAD.

VII. Conclusion :

Nous avons présenté dans ce chapitre les différents concepts qui sont utilisés au sein de notre travail : la définition de la modélisation et les différentes méthodes de modélisation de workflows. Parmi elles le langage UML, et plus précisément leur diagramme d'activité que nous allons définir dans le chapitre suivant.

Chapitre 2

La modélisation des Diagrammes d'Activité d'UML

I. Introduction :

Les méthodes graphiques offrent un moyen prometteur pour maîtriser la complexité des systèmes modélisés en fournit une représentation complète et une meilleure vue. Le langage UML est l'un des langages les plus utilisés dans la modélisation informelle qui permet une représentation avec une syntaxe très riche. En effet UML est adopté comme un standard industriel incontournable.

Dans ce chapitre nous allons nous intéresser au diagramme d'activité d'UML comme un diagramme comportemental pour la modélisation des processus métier et les systèmes workflows.

II. Le langage UML :

UML est un langage semi-formel de modélisation graphique et de notation orientée objet, qui a été développé et standardisé par Rational Software et Object Management Group (OMG). Il a vu le jour en 1997 et est très rapidement devenu un standard de l'industrie pour spécifier, visualiser, développer et documenter des logiciels.

UML est un langage qui permet de modéliser non seulement des applications informatiques ou des structures de données, mais également les activités d'un domaine : mécanique, biologie, processus métier...et de taille quelconque.

Le langage de modélisation unifié UML (en anglais : *Unified Modeling Language*) est né, comme son nom l'indique, du regroupement de trois techniques de modélisation objet, la méthode Booch, OMT (*Object Modeling Technique*) et OOSE (*Object Oriented Software Engineering*), respectivement développées par GradyBooch, James Rumbaugh et Ivar Jacobson. **[GLA 02]**

Au niveau de *Unified Modeling Language*, deux éléments importants sont à noter. Le terme *unified* et le terme *langage*. Le premier terme signifie que les auteurs ont essayé de regrouper les éléments importants des concepts objets, alors que le deuxième montre qu'il s'agit d'un langage de modélisation et non d'une méthode, car une méthode de conception est définie comme une démarche d'organisation qui a pour objectif de résoudre un problème spécifique. La méthode de conception utilise un formalisme ou un langage pour exprimer le résultat. **[HAM 15]**

L'usage d'une représentation graphique est un atout, car les diagrammes effacent les ambiguïtés dans les modèles. Un dessin exprime de manière plus naturelle et plus lisible ce qu'un texte peine à réaliser même lorsqu'il est bien commenté. **[BOU 13]**

UML dans sa version 2.0, comporte treize types de diagrammes. Ils sont répartis en deux grands groupes : **[AOU 14]**

1. Les diagrammes structurels ;
2. Les diagrammes comportementaux ou dynamiques.

II.1. Diagrammes structurels ou statiques :

Ces diagrammes permettent de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système informatisé.

- ❖ **Le diagramme de classes** : fournit une représentation sous forme de classes et de relations. Il ne contient pas d'informations temporelles, une classe est une représentation abstraite d'un ensemble d'éléments similaires. [GLA 02]
- ❖ **Le diagramme d'objets** : représente les objets et leurs relations, un objet étant un élément particulier d'une classe. [GLA 02]
- ❖ **Le diagramme de composants** : montre l'implémentation physique d'un système, en termes de composants logiciels. [GLA 02]
- ❖ **Le diagramme de déploiement** : décrit la configuration des éléments de traitement à l'exécution et les composants qui leur sont rattachés. [GLA 02]
- ❖ **Le diagramme de paquetage** : c'est une représentation simplifiée des diagrammes de classes complexes. Il peut regrouper les classes en paquets. Un paquet est une collection d'éléments UML logiquement liés. [AOU 14]
- ❖ **Le diagramme de structure composite** : permet de présenter la décomposition hiérarchique d'un objet, un cas d'utilisation, une collaboration, une activité ou une classe, en un ensemble de structures internes. [AOU 14]

II.2. Diagrammes comportementaux ou dynamiques :

Les diagrammes comportementaux modélisent les aspects dynamiques du système. Ces aspects incluent les interactions entre le système et ses différents acteurs, ainsi que la façon dont les différents objets contenus dans le système communiquent entre eux.

- ❖ **Le diagramme de cas d'utilisation** : représente les comportements d'un système du point de vue de l'utilisateur. [GLA 02]
- ❖ **Le diagramme d'activités** : décrit les flux entre activités au sein d'un système. Cela permet de représenter le déroulement d'une procédure ou d'une fonction. [GLA 02]
- ❖ **Le diagramme d'états-transitions** : exprime le comportement dynamique d'un objet en termes d'états, d'activités, de transitions et d'événements. [GLA 02]
- ❖ **Le diagramme de séquence** : représente les objets et leurs interactions selon une ligne temporelle. [GLA 02]

- ❖ **Le diagramme de collaboration ou communication** : est une combinaison des diagrammes classes, séquences, cas de utilisation et objet, Il est composé d'un ensemble d'objets qui communiquent entre eux par des messages. [AOU 14]
- ❖ **Le diagramme de temps** : montre le comportement des objets dans une période de temps donnée. [AOU 14]
- ❖ **Le diagramme vu d'ensemble des interactions** : met l'accent sur la vue d'ensemble du flux de contrôle des interactions. [AOU 14]

Dans notre recherche nous intéressons aux diagrammes d'activités d'UML comme des diagrammes comportementaux, que nous allons les bien définir par la suite.

III. Les diagrammes d'activités d'UML :

Les diagrammes d'activité permettent de mettre l'accent sur les traitements. Ils sont utilisés pour représenter le comportement d'un système (ou d'une partie d'un système) ou le déroulement d'un cas d'utilisation, ils montrent les flux de contrôle et les flux d'information qui circulent au sein d'un système, ils permettent aussi de représenter le séquençage exact des activités et de définir des conditions d'exécution.

Les diagrammes d'activité d'UML sont donc utilisés pour modéliser les systèmes workflows. [HAM 15]

III.1. Intérêts des diagrammes d'activité : [HAM 15]

- ❖ Représenter graphiquement le comportement interne d'une opération ou d'un cas d'utilisation sous forme d'une suite d'actions ;
- ❖ Utiliser le mécanisme de synchronisation pour représenter les successions d'états synchrones ;
- ❖ Utiliser des transitions automatiques évite la nécessité d'existence d'évènement de transition pour avoir un changement d'états ;
- ❖ Modéliser un workflow dans un cas d'utilisation, ou entre plusieurs cas d'utilisations ;
- ❖ Définir avec précision les traitements qui ont cours au sein du système ;
- ❖ Le diagramme d'activité est le plus approprié pour modéliser la dynamique d'une tâche ou d'un cas d'utilisation.

III.2. Notation: [GLA 02]

Nous présentons dans cette partie quelques éléments qui peuvent être contenus dans un diagramme d'activité :

III.2.1. Les nœuds :

III.2.1.1. Nœud d'activité :

Une activité est une série d'actions, elle cause un changement de l'état du système. Un nœud d'activité est un type d'élément abstrait permettant de représenter les étapes le long du flot d'une activité. Il existe trois types de nœuds d'activités : [HAM 15]

A. Nœud d'objet :

Un nœud d'objet permet de définir un flot d'objets dans un diagramme d'activités. Un flot d'objets permet de passer des données d'une activité à une autre. [HAM 15]

UML 2 propose deux notations équivalentes pour les flots d'objets, la première utilise la notation d'un nœud objet, alors que la deuxième utilise la notation des pins.

- ❖ Pour la première notation, un nœud d'objets est représenté par un rectangle contenant le nom du nœud. Il est de plus possible d'écrire l'état de l'objet entre crochets en dessous de son nom. [HAM 15]

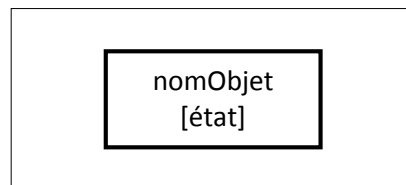


Figure.2.1. Notation d'un nœud d'objet.

- ❖ L'autre notation possible consiste à indiquer un petit carré attaché à la bordure d'une activité appelé *broche* (ou *pins* en anglais). [HAM 15]

Un pin représente un point de connexion pour une activité, il est connecté en entrée ou en sortie d'une activité. L'activité ne peut débuter qu'après l'affectation d'une valeur à chacun de ses pins d'entrée. Quand l'activité se termine, une valeur doit être affectée à chacun de ses pins de sortie.

Graphiquement, un pin est typé et éventuellement nommé. Il peut contenir des flèches indiquant sa direction (entrée ou sortie). [HAM 15]

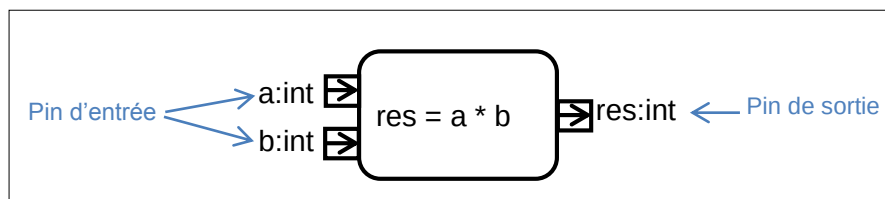


Figure.2.2. Représentation des pins d'entrée et de sortie sur une activité. [GER 15]

- ❖ Le nœud d'expansion est un nœud d'objet qui peut être utilisé pour indiquer un flux à travers les limites d'une région d'expansion (pour cette dernière voir le paragraphe III.1.1.3). [HAM 15]

B. Nœud de contrôle :

Un nœud de contrôle est un nœud d'activités abstrait utilisé pour coordonner les flux entre les nœuds d'une activité. [HAM 15]

❖ Nœud initial : [MAL 15]

Un nœud initial est un nœud de contrôle qu'il marque le début d'un diagramme d'activités. Il possède un arc sortant et pas d'arc entrant, on peut avoir plusieurs nœuds initiaux dans une activité. Il est représenté graphiquement par un point noir.

❖ Nœud final : [HAM 15]

Un nœud final est un nœud de contrôle dans lequel le flux d'activité s'arrête. Il peut avoir un ou plusieurs arcs entrants et aucun arc sortant. Il existe deux types de nœuds finaux :

1. Les nœuds finaux d'activité:

Dans un nœud final d'activité, lorsque l'un de ses arcs entrants est activé, l'exécution de l'activité en cours s'achève, et tout nœud ou flux actif au sein de cette activité est abandonné. Il est représenté par un cercle avec un point noir à l'intérieur.

2. Les nœuds finaux de flux :

L'arrivée du flux d'exécution à un nœud final de flux, implique la terminaison du flux de ce dernier. Mais cette fin n'a aucun effet sur les autres flux actifs de l'activité. Il est représenté par un cercle avec une croix.



Figure.2.3. Notations de nœud initial et les deux types de nœuds finaux.

❖ Nœud de décision :

Le nœud de décision permet de faire un choix entre plusieurs flux sortants conditionnés par une expression booléenne. Il est représenté par une forme de diamant et il peut être nommé aussi. [HAM 15]

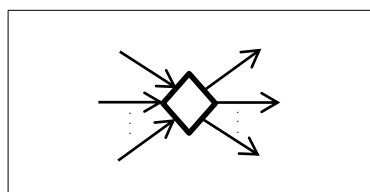


Figure.2.4. Notation de nœud de décision.

❖ Nœud de fusion ou interclassement :

Un nœud de fusion est un nœud de contrôle, il rassemble plusieurs flots alternatifs entrants en un seul flot sortant. Il ne s'agit pas de synchronisation d'actions. (Même représentation que le nœud de fusion). [MAL 15]

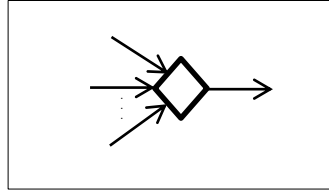


Figure.2.5. Notation de nœud de fusion.

❖ Nœud de bifurcation :

Le nœud de bifurcation est un point de divergence qui sépare un flux d'entrée en plusieurs flots *concurrents* en sortie, c'est à dire toutes les activités sont enclenchées en parallèle, c'est un branchement multiple. Il est représenté par une barre horizontale ou verticale. [GUE 11]

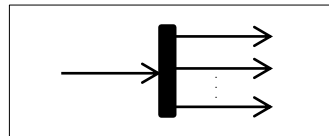


Figure.2.6. Notation de nœud de bifurcation.

❖ Nœud d'union :

Un nœud d'union (nœud de jointure) est un point de convergence qui *synchronise* des flots multiples. Il possède plusieurs arcs entrants et un seul arc sortant. Le flux de sortie ne peut exécuter jusqu'à ce que tous les flux d'entrées aient été reçus. (Même représentation que le nœud de bifurcation). [MAL 15]

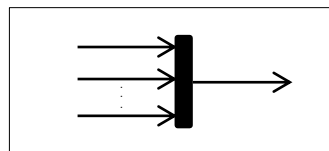


Figure.2.7. Notation de nœud d'union.

C. Nœud exécutable :

Une action est un nœud d'activité exécutable, c'est le plus petit traitement qui puisse être exprimé en UML. Elle prend un ensemble d'entrées et les convertit en un ensemble de résultats. [HAM 15] Graphiquement, une action est représentée par un rectangle avec des coins arrondis.

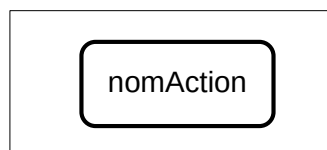


Figure.2.8. Notation de nœud d'action.

SendSignal (envoyer signal), AcceptEvent (attente de réception d'un signal), AcceptTimeEvent (attente d'un événement temporel) sont des types d'action prédéfinis dans la notation UML. [ROD 09]

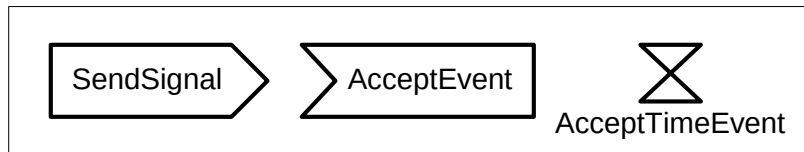


Figure.2.9. Notations des nœuds exécutable. [ROD 09]

III.2.1.2. Une partition d'activité : [HAM 15]

Les partitions d'activité appelées aussi couloirs ou lignes d'eau (en anglais *swimlane*), divisent l'espace des nœuds et des arcs pour qu'ils permettent d'indiquer qui (de l'utilisateur ou du système) réalise les actions.

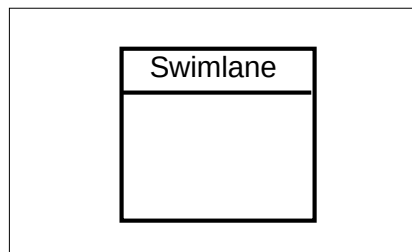


Figure.2.10. Notation de la partition.

III.2.1.3. Une région d'expansion:[HAM 15]

Une région d'expansion est une zone d'activités à des entrées et des sorties sous forme de nœuds d'expansion. Ces derniers sont matérialisés sur la frontière de la boîte (la région d'expansion) par des petits rectangles divisés en plusieurs compartiments. Elle s'exécute une fois pour chaque élément dans une collection d'entrée. Le mot-clé "itérative", "parallèle" ou "flux" est affiché dans le coin en haut à gauche de la région pour indiquer le mode de répétition.

La région d'expansion assure la gestion d'itérations. Elle est représentée par un cadre arrondi pointillés.

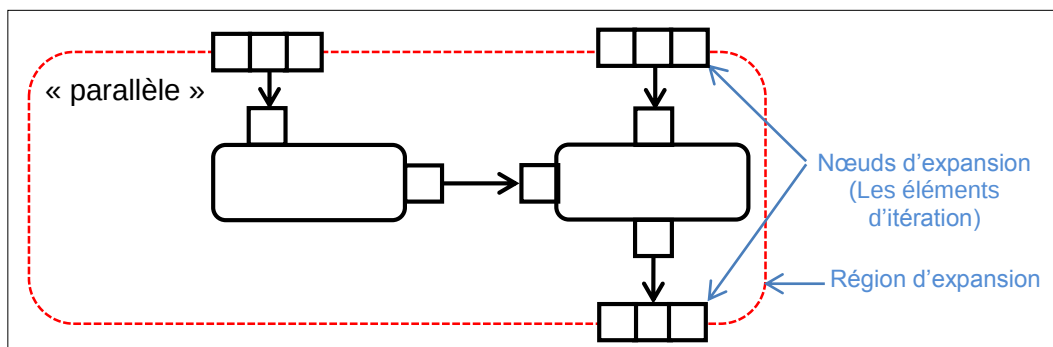


Figure.2.11. Notation d'une région d'expansion. [spa 15]

III.2.2. Les arcs:

III.2.2.1. Flot de contrôle (control flow) : [HAM 15]

Un arc de flux de contrôle est un arc qui permet de relier et de montrer le chemin d'exécution de deux nœuds d'activité. Il ne transmet pas des données.

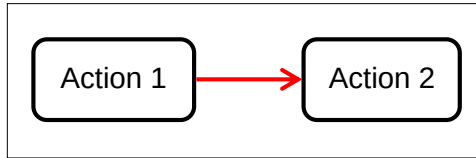


Figure.2.12. Notation d'un flot de contrôle. [HAM 15]

III.2.2.2. Flot d'objet (object flow) : [HAM 15]

Un flux d'objets est un arc qui connecte des nœuds d'objet pour fournir des entrées aux actions.

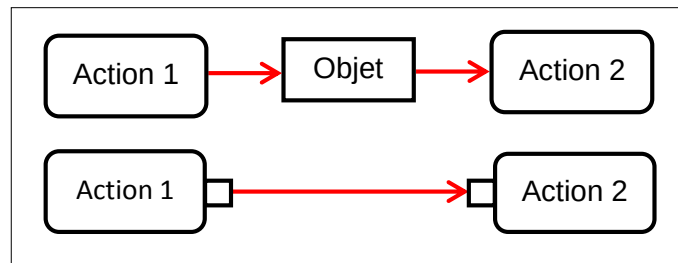


Figure.2.13. Les deux notations d'un flot d'objet. [HAM 15]

III.2.2.3. Un Handler d'exception :

Un Handler (ou un gestionnaire d'exception) est un élément qui spécifie un corps à exécuter dans le cas où une exception serait levée au cours de l'exécution d'un nœud protégé. Il est représenté par une flèche « éclair ».

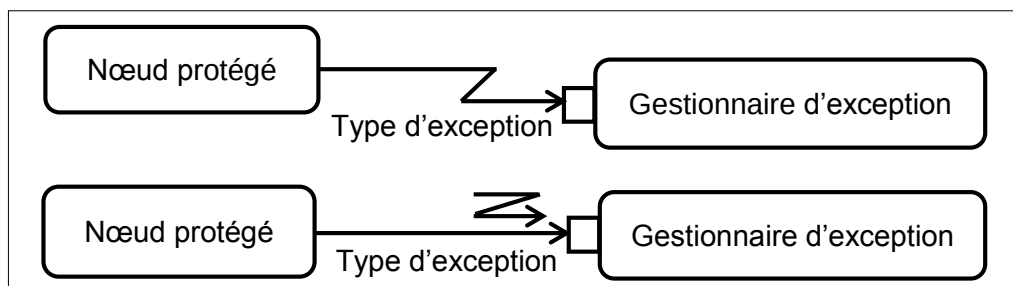


Figure.2.14. Les deux notations d'un Handler. [HAM 15]

Voici un exemple d'un diagramme d'activité UML:

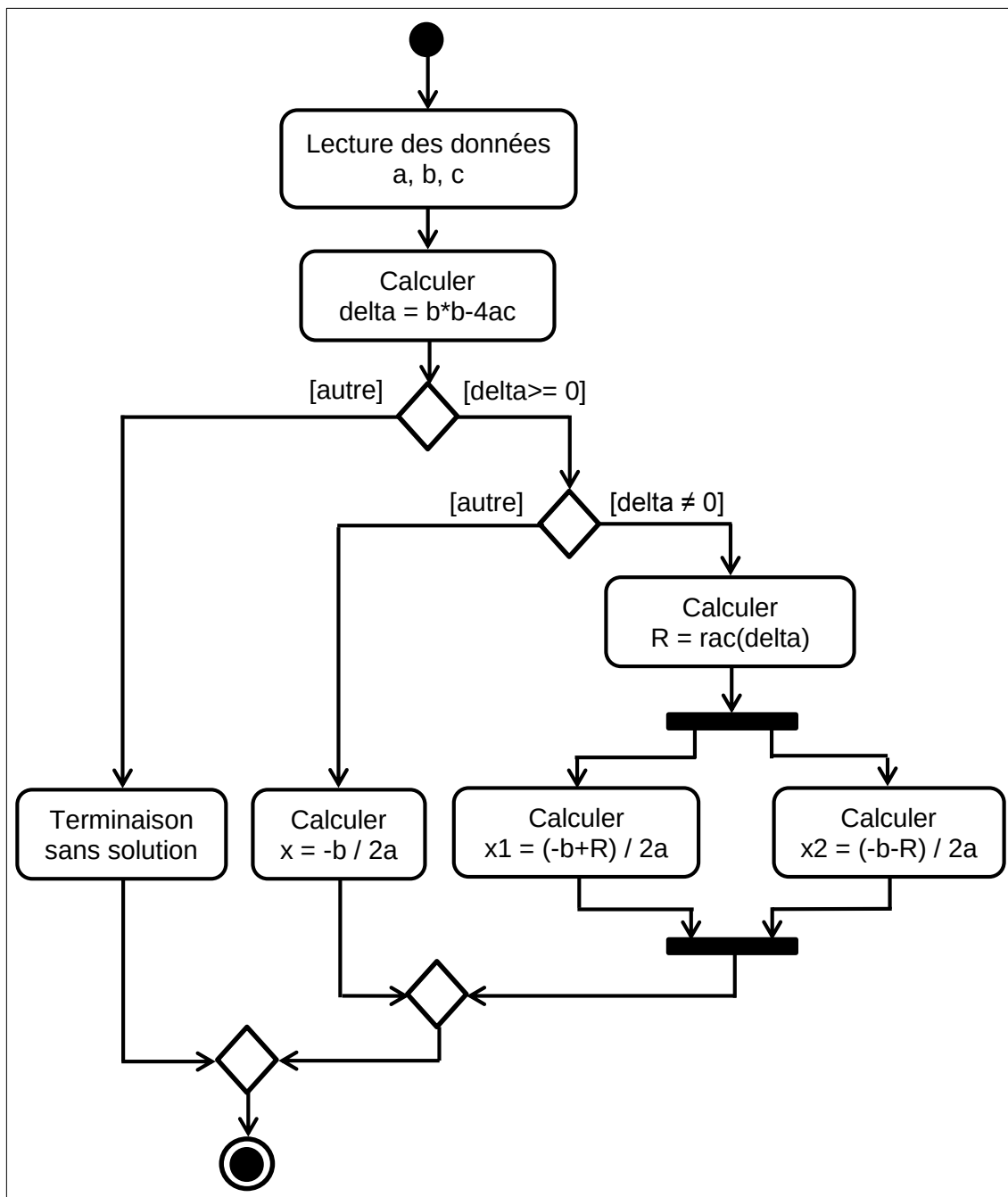


Figure.2.15.Exemple d'un diagramme d'activité. [HAM 15]

IV. Conclusion :

Dans ce chapitre, nous avons présenté le langage de modélisation UML, et ses diagrammes. Ensuite, nous avons donné une description détaillée des diagrammes d'activité d'UML 2.0 qui constituent un outil de modélisation des processus métiers et des

systèmes workflows. Ils sont adaptés à la modélisation du cheminement de flux de données et de flux de contrôle.

Les diagrammes d'activité ont des mécanismes d'expressions graphiques et textuelles très riches pour exprimer les évènements, le séquençement, le choix et le parallélisme. Mais ils souffrent de l'inconvénient du manque des outils de vérification et d'analyse de comportement, qui sont disponible dans les réseaux de Pétri. Le passage des diagrammes d'activité vers les réseaux de Pétri se fait grâce à un ensemble de règles de transformation. Ceci constitue l'objet principal de notre travail.

Chapitre 3

Les Réseaux de Pétri et la transformation des modèles

I. Introduction :

Après avoir vu dans le chapitre précédent la modélisation d'un processus métier en diagramme d'activité d'UML, on procède à son interprétation et son exécution. Mais malheureusement, le langage semi-formel UML ne permet pas de le vérifier.

Dans ce chapitre, nous allons voir les réseaux de Pétri comme des outils formels puissants pour la modélisation, l'analyse et la vérification des diagrammes d'activité d'UML, afin de démontrer mathématiquement leur bon fonctionnement.

Pour pouvoir effectuer cette analyse et vérification formelle du diagramme d'activité, on doit, avant tout, procéder à sa conversion en un réseau de Pétri, en respectant son comportement (sémantique) préconise par l'OMG selon un ensemble de règles.

II. La vérification formelle de diagrammes UML :

UML est un langage à caractère plutôt visuel, il souffre d'un manque de sémantique formelle, car il ne fournit pas une notation mathématique pour décrire précisément le comportement attendu des processus métier de l'entreprise. En effet, les modèles d'UML développés pourront donc contenir des incohérences ou des inconsistances difficiles à détecter manuellement, et dans le cas où les processus modélisés sont complexes, cela devient impossible.

Les méthodes formelles représentent une solution intéressante à ce problème. Elles auront pour effet d'éliminer les ambiguïtés et les erreurs au niveau de l'interprétation des modèles. Les réseaux de pétri sont l'un de ces méthodes les plus intéressants et largement utilisés.

III. Les réseaux de Pétri :

Les réseaux de Pétri (noté RdP) ont été introduits par Carl Adam Pétri, en 1962. Un RdP est un outil graphique et mathématique pour modéliser et analyser (ou vérifier) les comportements de plusieurs types de systèmes, parmi lesquels les processus métier, grâce aux modèles de graphes, aux équations algébriques,...etc. En plus, il nous permet de simuler les activités dynamiques et concurrentes. Un réseau de Pétri est un graphe orienté et biparti, il est donné par quatre types de nœuds : **[DRI 14]**

- ❖ Une place « P », qui correspond à l'état d'une ressource du système ou une condition, elle est représentée par un cercle ou un rond.
- ❖ Une transition « T », qui correspond à un événement ou une action se déroulant au sein du système, elle est représentée par rectangle ou un trait.
- ❖ Un marquage d'une place (ou jeton), qui définit le nombre de ressources présentes dans la place, on le représente par un point noir.

- ❖ Un arc qui représente les conditions nécessaires pour déclencher une action, il ne relie jamais deux sommets (ou nœuds) de la même famille.

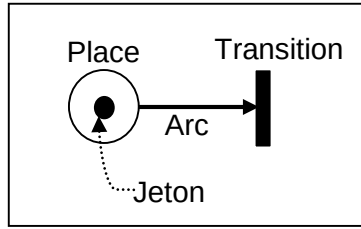


Figure.3.1. Représentation graphique d'un réseau de Pétri.

On dit qu'une transition est « source » : si elle n'a pas de place en entrée. Et on dit qu'une transition est « puits » : si elle n'a pas de place en sortie.

III.1. Le marquage :

L'état du système modélisé par un réseau de Pétri est représenté par un vecteur appelé marquage (noté M), on dispose un nombre entier (positif ou nul) et fini de marques ou jetons dans chaque place du réseau. Le marquage du réseau est constitué de toutes les marques présentées dans le réseau à un instant donné, par exemple le marquage qui décrit l'état initial du système à l'instant $t = 0$ est noté par M_0 . [DRI 14]

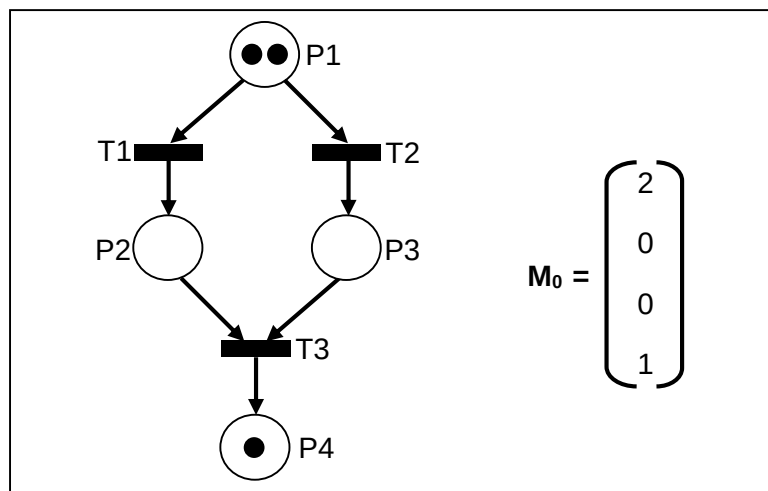


Figure.3.2. Exemple d'un réseau de Pétri marqué avec son vecteur de marquage M_0 .

III.2. Représentation matricielle : [DRI 14]

Il est possible de représenter les réseaux de Pétri par des matrices.

- ❖ La matrice $\text{Pré}(p_i, t_j)$: c'est le poids k de l'arc reliant une place à une transition.

$$\text{Pré}(P_i, T_j) = \begin{cases} k, & \text{Si l'arc } W(p_i, t_j) \text{ existe} \\ 0, & \text{sinon} \end{cases}$$

- ❖ La matrice $\text{Post}(t_j, p_i)$: c'est le poids k de l'arc reliant une transition à une place.

$$\text{Post}(T_j, P_i) = \begin{cases} k, & \text{Si l'arc } W(t_j, p_i) \text{ existe} \\ 0, & \text{sinon} \end{cases}$$

- ❖ La matrice $C = \text{Post} - \text{Pré}$ est appelée matrice d'incidence du réseau.

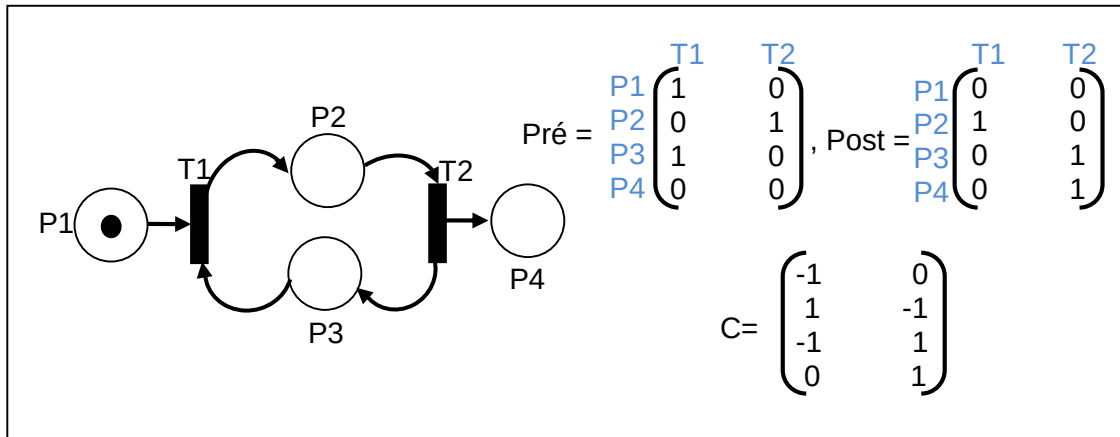


Figure.3.3. Un RdP et ses matrices de Pré-condition, Post-condition et C. [HAMA 13]

III.3. Fonctionnement des réseaux de Pétri :

Le fonctionnement d'un RdP correspond à l'évolution de son marquage au cours du temps. Il se traduit par un déplacement de marques ce qui s'interprète comme la consommation / production de ressources déclenchée par des événements ou des actions. Pour simuler le comportement dynamique d'un système, l'état ou le marquage dans le réseau, est changé selon la règle de transition suivante : « Une transition est validée (on dit aussi sensibilisée ou franchissable ou encore tirable) si toutes ses places en amont contiennent au moins un jeton (ou une marque) ». Et donc, un réseau ne peut évoluer que par franchissement d'une seule transition à la fois, une transition choisie parmi toutes celles qui sont validées à cet instant. [DRI 14]

Le franchissement (ou le tir) d'une transition, consiste à enlever un jeton dans chacune des places d'entrée de la transition et à ajouter un jeton dans chacune des places de sortie de la même transition. [DRI 14]

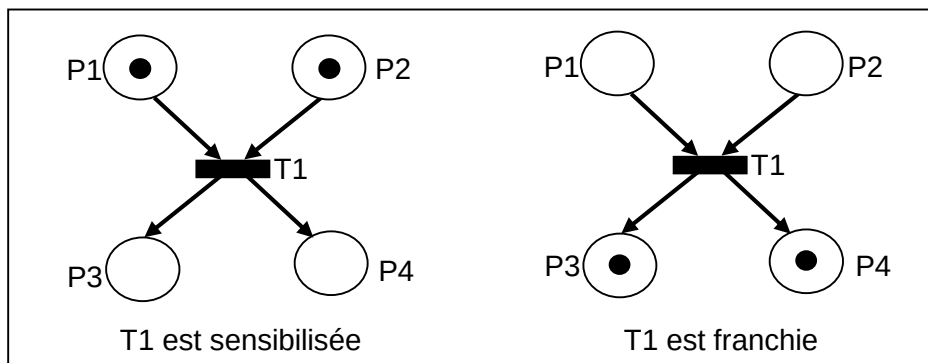


Figure.3.4. Evolution du marquage.

Remarque :

- ❖ La transition *source* est toujours franchissable ;
- ❖ Lorsqu'une transition est validée, cela n'implique pas qu'elle sera immédiatement franchie; cela ne représente qu'une possibilité de franchissement ou d'évolution du RdP ;
- ❖ Le réseau ne peut évoluer que par franchissement d'une *seule transition à la fois*, une transition choisie parmi toutes celles qui sont validées à cet instant.

III.4. Réseaux de Pétri particuliers :

III.4.1. Les structures particulières :

III.4.1.1. Graphe d'états : [DRI 14]

Un RdP est un graphe d'état si et seulement si toute transition a exactement une place en entrée et une place en sortie.

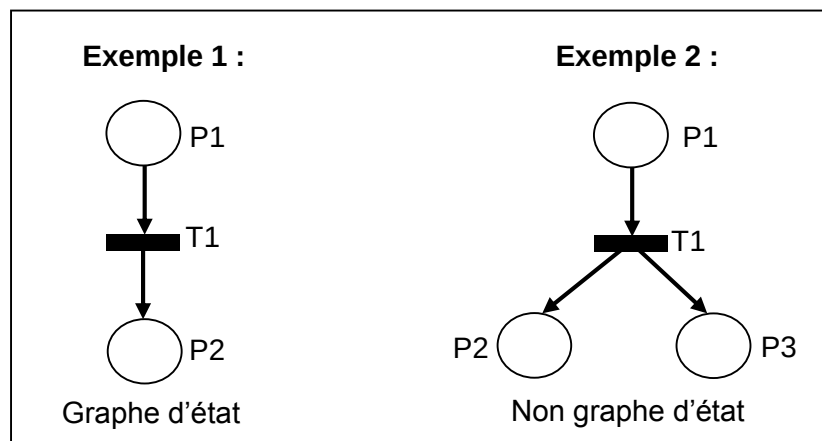


Figure.3.4. Exemple de graphe d'état.

III.4.1.2. Graphe d'évènement (Graphe de transitions) : [DRI 14]

Un RdP est un graphe d'évènement (GE) si et seulement si toute place a exactement une transition en entrée et une transition en sortie.

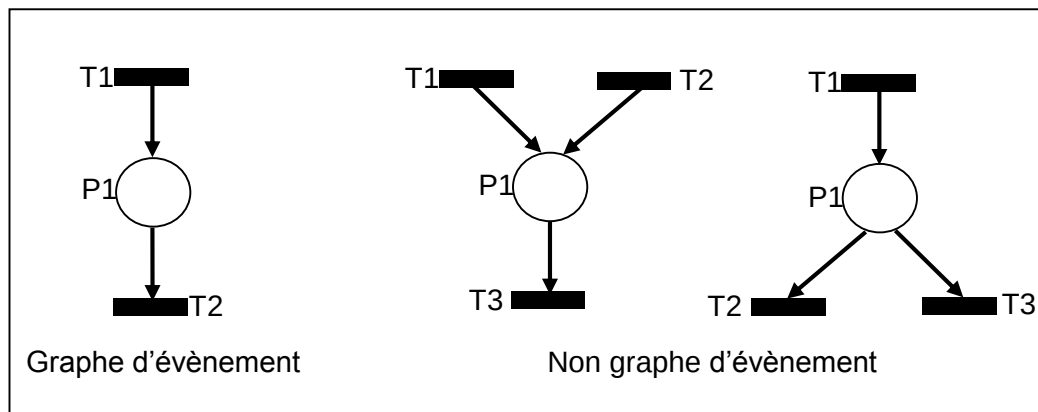


Figure.3.5. Exemple de graphe d'évènement.

III.4.1.3. RdP sans conflit : [DRI 14]

C'est un RdP dans lequel toute place a *au plus* une transition de sortie. Un conflit (structurel) correspond à l'existence d'une place P_i qui a *au moins deux* transitions de sortie $T_j, T_k, \dots$ etc. Notation $\langle P_i, \{T_j, T_k, \dots\} \rangle$.

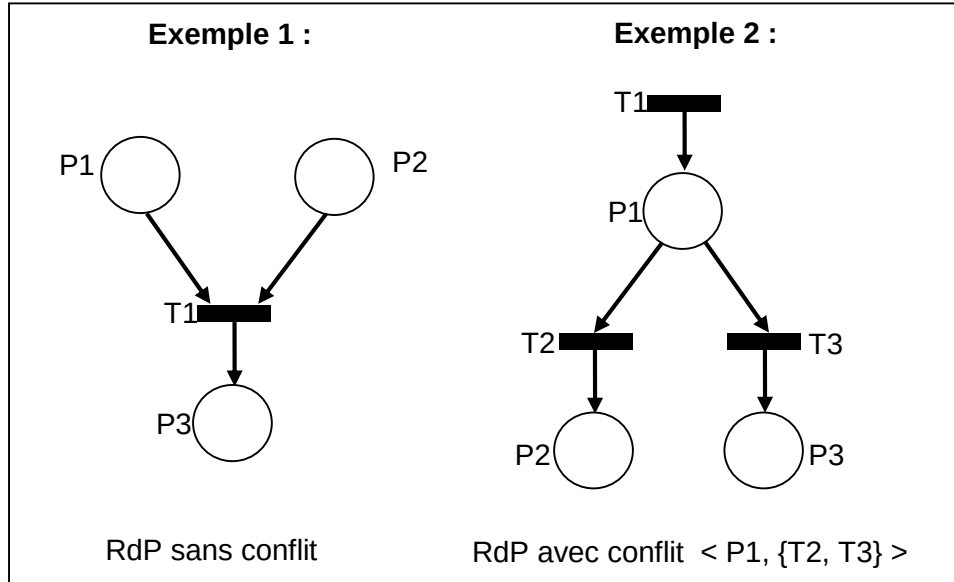


Figure.3.6. Exemple de graphe sans et avec conflit.

III.4.1.4. RdP à choix libre : [DRI 14]

Un RdP à choix libre est un RdP dans le quel pour tout conflit $\langle P1, \{T1, T2, \dots\} \rangle$ aucune des transitions $T1, T2, \dots$ ne possède pas une autre place d'entrée que $P1$.

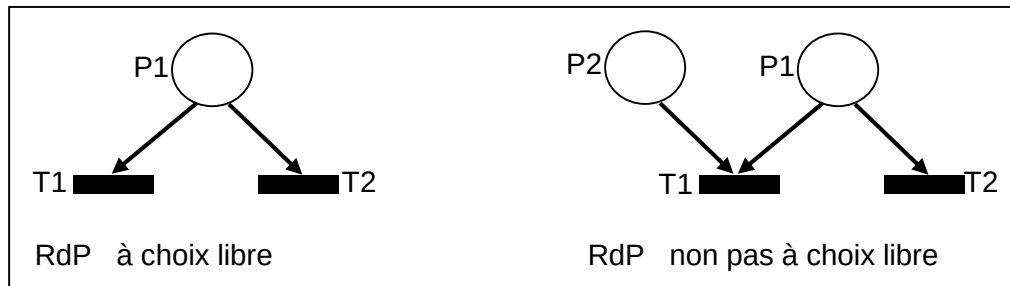


Figure.3.7. Exemple de graphe à choix libre / non pas à choix libre.

III.4.1.5. RdP simple : [DRI 14]

C'est un RdP dans lequel chaque transition ne peut être concernée que par un conflit au plus.

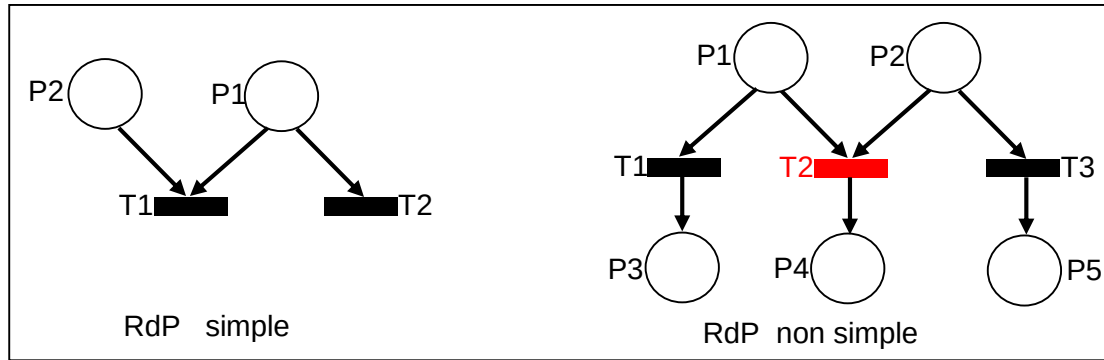


Figure.3.8. RdP simple / non simple.

III.4.1.6. RdP pur : [DRI 14]

C'est un RdP dans lequel, il n'existe pas de transition ayant une place d'entrée qui soit également une place de sortie de cette transition.

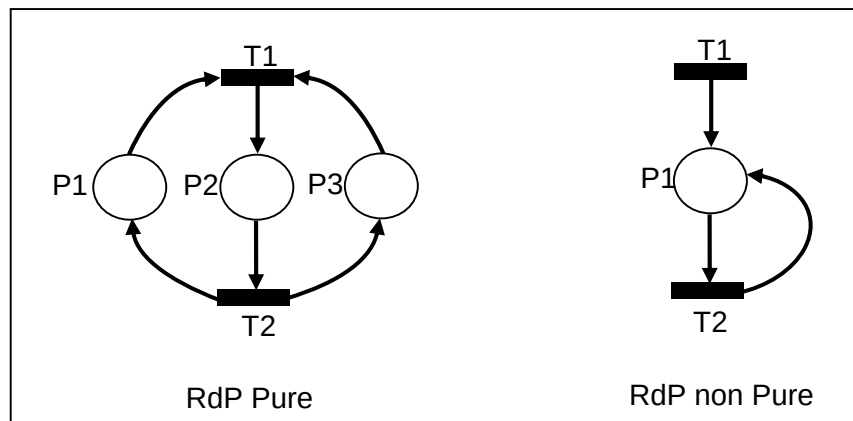


Figure.3.9. RdP pur / impur.

III.4.2. Schémas particuliers de RdP :

III.4.2.1. Parallélisme : [DRI 14]

Le parallélisme (ou concurrence) est facilement modélisé par les RdP. Il représente la possibilité que plusieurs processus évoluent simultanément au sein du même système de façon indépendante l'une de l'autre. On peut provoquer le départ simultané de l'évolution de deux processus à l'aide d'une transition ayant plusieurs places de sortie.

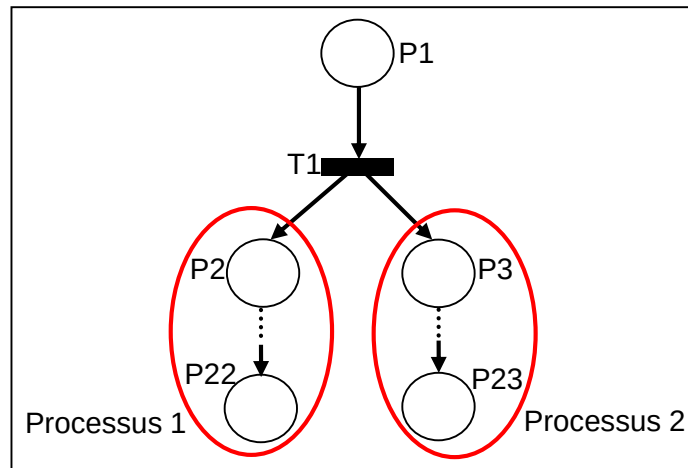


Figure.3.10. Parallélisme.

III.4.2.2. Synchronisation : [DRI 14]

Pour que deux programmes puissent coopérer, il faut que les résultats de chacun soient présents au bon moment.

Ces réseaux de Pétri permettent de faire communiquer et interagir les systèmes avec l'environnement ou un système extérieur. Ils permettent donc de modéliser des systèmes soumis à des contraintes externes.

Exemple : Rendez-vous / Sémaphore

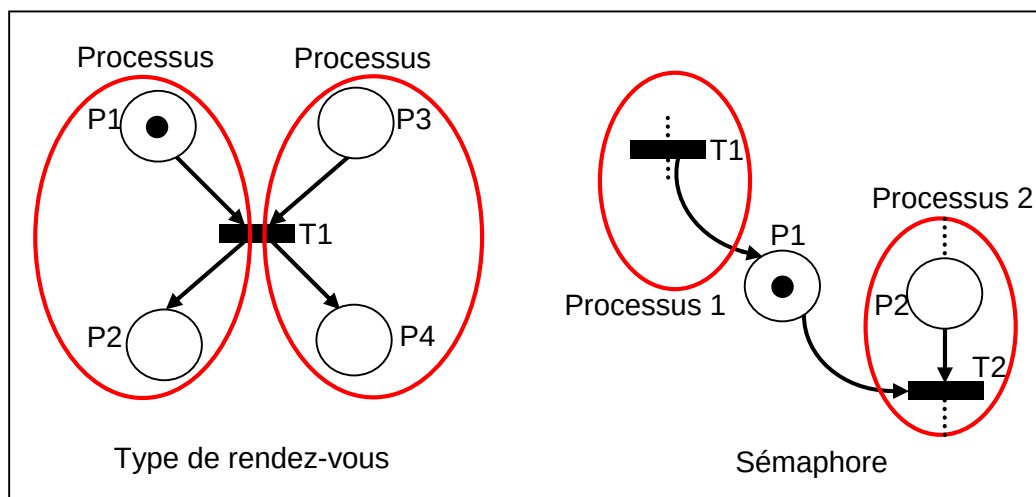


Figure.3.11.Synchronisation.

III.5. Les propriétés des réseaux de Pétri :

Le plus grand intérêt d'un réseau de Pétri est qu'il ne se limite pas à sa capacité de modéliser une grande variété de systèmes. En effet, un RdP offre aussi un large éventail de propriétés mathématiques qui permettent le bon fonctionnement d'un système. Dans ce qui suit nous allons présenter certaines de ces propriétés.

III.5.1. Caractère Borné : [DRI 14]

Une place « p » du réseau marqué (R, M_0) est k -bornée ($k \in \mathbb{N}$, avec $k > 0$) si pour tout marquage M accessible depuis M_0 , $M(p) \leq k$ (le nombre de marques dans P_i est fini). Dans le cas contraire la place « p » est dite non-bornée.

Un RdP est borné pour un marquage initial M_0 si toutes ses places sont bornées.

Remarque : Un RdP s'il est 1- borné est dit « sauf ».

III.5.2. Pseudo-vivacité : [DRI 14]

Un réseau de Pétri (R, M_0) est dit pseudo-vivant si pour tout marquage accessible depuis le marquage initial M_0 , il existe toujours une transition « t » qui puisse être franchie.

III.5.3. Quasi-vivacité : [DRI 14]

La quasi-vivacité d'une transition signifie que depuis le marquage initial M_0 , cette transition peut être franchie au moins une fois, sinon elle est morte. Conséquemment (inutile). Le réseau (R, M_0) est dit quasi-vivant si toutes les transitions sont quasi-vivantes.

III.5.4. Vivacité : [DRI 14]

La propriété de vivacité est une propriété forte, souvent difficile à vérifier. La vivacité d'une transition exprime le fait que quel que soit l'évolution du réseau à partir du marquage initial, le franchissement à terme de cette transition est toujours possible.

Un réseau est vivant si toutes ses transitions le sont.

III.5.5. Absence de blocage : [DRI 14]

Cette propriété implique seulement que le réseau a toujours la possibilité d'évoluer et de progresser. [DRI 14]

III.5.6. Ré-initialisabilité : [DRI 14]

Un RdP est ré-initialisable si et seulement si pour tout marquage M , il existe une séquence de transitions qui permet de revenir au marquage initial M_0 . [DRI 14]

Le tableau suivant résume des exemples de toutes les propriétés étudiées dans ce mémoire :

Propriété	Exemple
Réseau 3-borné	
Réseau non borné	
Réseau pseudo-vivant	
Réseau quasi-vivant	
Réseau vivant	

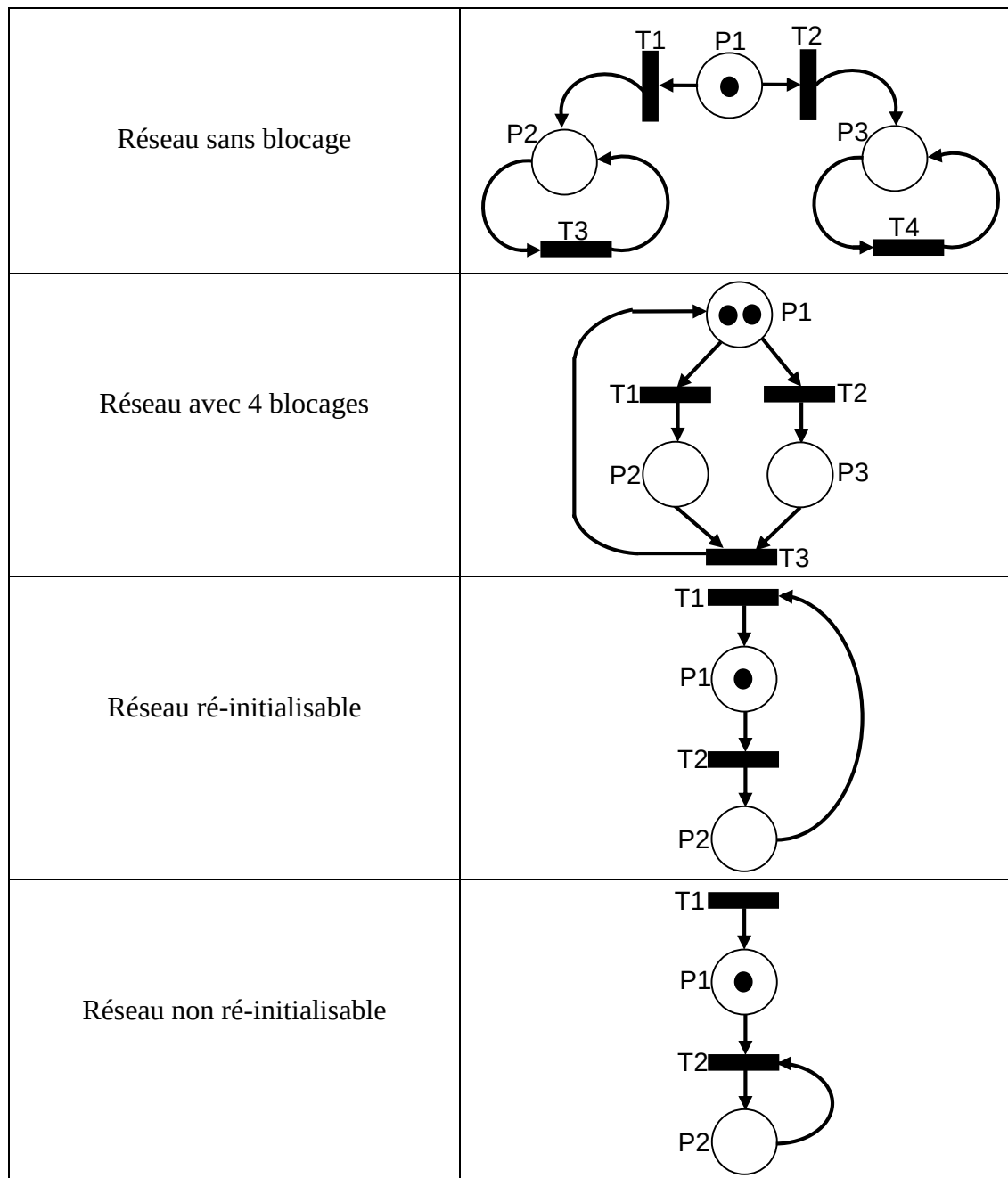


Tableau.3.1.Exemples des propriétés des RdP.

Ces propriétés doivent être complétées par leur analyse ou leur vérification, cela se fait grâce aux méthodes d'analyse.

IV. Les méthodes d'analyse :

Il existe plusieurs méthodes pour analyser les RdP (par le graphe de marquage, par l'algèbre linéaire, par réduction ...etc). L'intérêt de ces techniques de validation est leur caractère formel qui permet de tirer des propriétés du système par une simple observation du graphe, dans notre travail on va utiliser la méthode de graphe de marquages accessibles.

IV.1. Le graphe des marquages accessibles :

L'idée la plus naturelle pour étudier les propriétés d'un RdP est de construire le graphe de tous les marquages accessibles. Un graphe de marquage est un ensemble de sommets (ou de nœuds) qui représente des marquages accessibles (toutes les situations possibles du RdP au cours de son évolution à partir du marquage initial), et un ensemble d'arcs qui sont symbolisés par les noms des transitions franchissable faisant passer d'un marquage à l'autre pour un marquage initial M_0 . [DRI 14]

Et on déduit des propriétés grâce aux techniques de théorie de graphe :

- ❖ Un réseau (R, M_0) est *1-borné* si et seulement si les nœuds ne contiennent que 0 ou 1.
- ❖ Le réseau marqué (R, M_0) admet un *blocage* si et seulement si le graphe des marquages contient un sommet sans successeur.
- ❖ Le réseau marqué (R, M_0) est *sans blocage* si et seulement si chaque nœud de son graphe est l'origine d'au moins un arc.
- ❖ Un réseau marqué (R, M_0) est *quasi-vivant* si et seulement si, pour toute transition t , il existe dans le graphe des marquages un arc d'étiquette t , les transitions absentes sont mortes.
- ❖ Un réseau marqué (R, M_0) borné est *vivant* si et seulement si chaque *composante fortement connexe* (c.f.c) du graphe qui n'a pas d'arc sortant contient au moins un arc marqué par chaque transition.

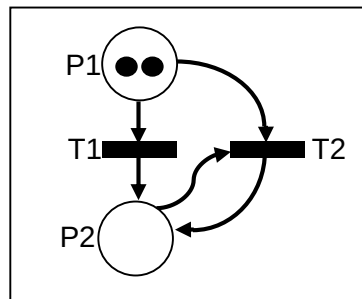


Figure.3.12. RdP marqué et borné.

On construit le graphe des marquages accessibles :

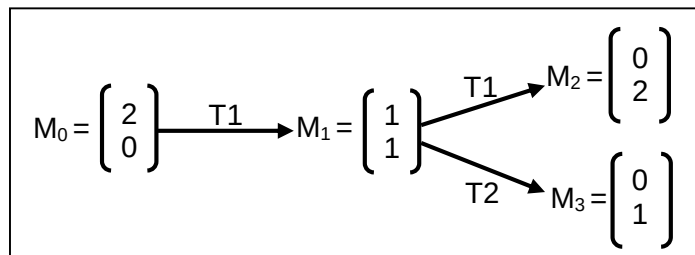


Figure.3.13. Graphe des Marquages accessibles.

→ Le réseau est : 2-borné, contient 2 états de blocage, non vivant, quasi-vivant.

Un graphe des marquages accessibles ne peut plus être construit quand le réseau est non borné. D'où le recours au graphe dit de couverture.

IV.1.1. L'arbre / arborescence & le graphe de couverture :

Un graphe de couverture noté GC(R). L'arbre de couverture est un graphe particulier tel que : pas de boucle, pas de circuit, elle comporte un nombre fini de sommets.

Pour construire ce graphe, on utilise le marquage symbolique Omega « ω » qui désigne un nombre de jetons dans une place P_i qui peut atteindre un nombre très grand (l'infinie).

Et on déduit des propriétés suivantes :

- ❖ Une place « p » d'un réseau marqué R est *non-bornée* s'il existe un sommet Q de GC(R) tel que : $Q(p) = \omega$.
- ❖ Un RdP non borné *n'est pas vivant* si son graphe de couverture possède au moins une composante fortement connexe (c.f.c) du graphe sans arc sortant et dont l'union des transitions attachées aux arcs n'est pas l'ensemble des transitions.
- ❖ Un RdP non borné est *avec blocage* si son graphe de couverture contient un nœud qui n'est l'origine d'aucun arc.

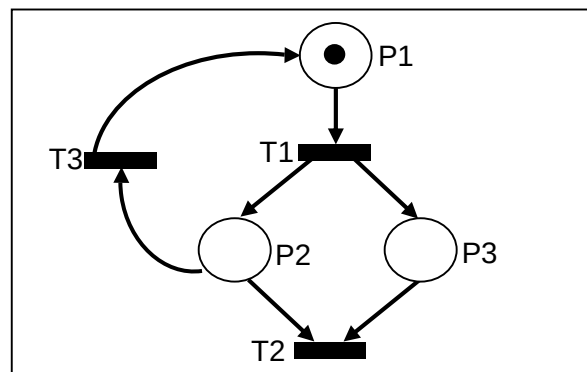


Figure.3.14. Exemple d'un RdP non borné.

On obtient le graphe de couverture en fusionnant les nœuds de l'arbre de couverture qui correspondent au même marquage.

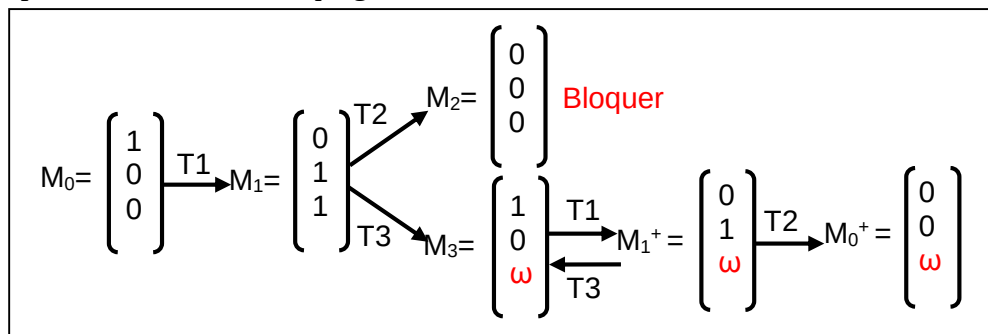


Figure.3.15. Graphe de couverture.

→ À partir de ce graphe de couverture le RdP est : non borné, avec blocage.

V. La transformation des diagrammes d'activités d'UML vers les réseaux de Pétri :

V.1. Transformation de Modèle :

Définition :

Une transformation est la génération automatique d'un modèle cible à partir d'un modèle source, selon une définition de transformation. Une définition de transformation est un ensemble de règles de transformation qui décrivent **comment** un modèle source peut être transformée en un modèle cible. [GUE 11]

Selon le standard UML, les diagrammes d'activité ont une sémantique similaire à celle des RdP, leur sémantique opérationnelle est basée sur un jeu de jeton similaire à celui des RdP. Et donc, le passage des diagrammes d'activités d'UML vers leurs équivalents en réseaux de Pétri afin de profiter des avantages de ce dernier est possible. Cela se fait par une transformation des modèles suivant un ensemble de règles décrites ci-dessous.

V.2. Nœud initial : [FOU 11]

- ❖ Création d'une place marquée et une transition qui permet de fournir à chaque arc sortant un jeton.

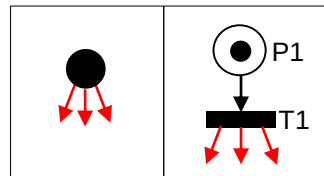


Figure.3.16. La notation de RdP équivalente au nœud initial.

V.3. Nœud final d'activité : [FOU 11]

- ❖ Création d'une transition pour chaque arc entrant dans le *nœud final d'activité*, afin d'activer la place finale et donc démarquer l'ensemble de l'activité.

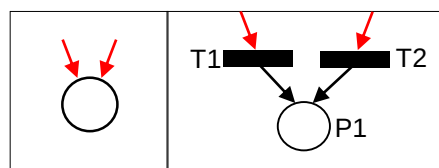


Figure.3.17. La notation de RdP équivalente au nœud final d'activité.

V.4. Nœud final de flux : [FOU 11]

- ❖ Création d'une transition pour chaque arc entrant dans le *nœud final de flux*.

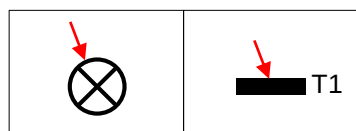


Figure.3.18. La notation de RdP équivalente au nœud final de flux.

V.5. Nœud de décision : [FOU 11]

- ❖ Une transition (T1) attend le flux de contrôle (et / ou le flux de donnée).
- ❖ Une place d'attente (P1) durant la phase de décision.
- ❖ Création des sorties (ici les transitions T2, T3) en fonction du nombre de sorties de *nœud de décision*.

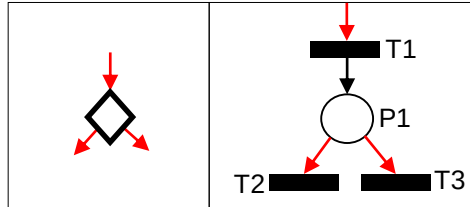


Figure.3.19. La notation de RdP équivalente au nœud de décision.

V.6. Nœud de fusion : [FOU 11]

- ❖ Création d'une transition (ici T1, T2) pour chaque entrée du nœud de fusion.
- ❖ Une place qui redistribue directement chaque jeton en sortie (T3).

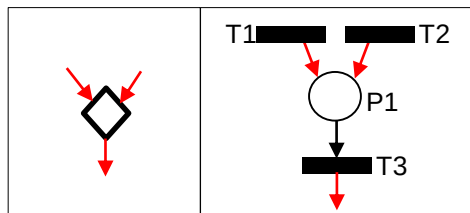


Figure.3.20. La notation de RdP équivalente au nœud de fusion.

V.7. Nœud de bifurcation : [FOU 11]

- ❖ Création d'une transition qui permet de redistribuer autant de jetons que d'arcs sortants de façon synchrone.

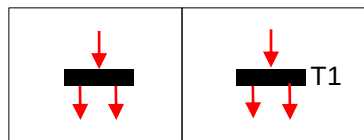


Figure.3.21. La notation de RdP équivalente au nœud de bifurcation.

V.8. Nœud d'union : [FOU 11]

- ❖ Création d'une transition qui a les mêmes propriétés de synchronisation dans les RdPs.

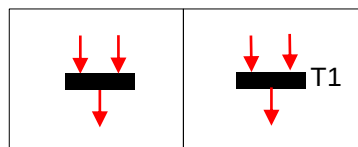


Figure.3.22. La notation de RdP équivalente au nœud d'union.

V.9. Nœud d'action : [FOU 11]

- ❖ Création d'une transition d'entrée et d'une transition de sortie pour la gestion du nœud d'union et du nœud de bifurcation implicite.

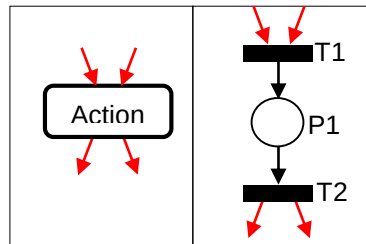


Figure.3.23. La notation de RdP équivalente au nœud d'action.

V.10. Nœud d'objet : [SPI 15]

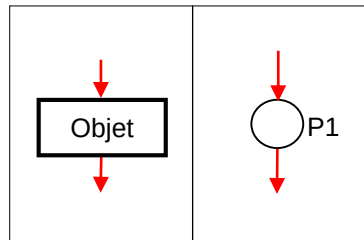


Figure.3.24. La notation de RdP équivalente au nœud d'objet.

V.11. Nœud d'envoi de signal : [FOU 11]

- ❖ Ajout d'une place de communication asynchrone.

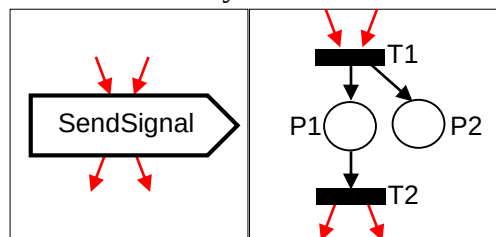


Figure.3.25. La notation de RdP équivalente au nœud d'envoi de signal.

V.12. Nœud de réception de signal et le nœud de temps d'attente : [SPI 15]

- ❖ « ...There is an *accept signal*...This type of signal is always enabled, hence it initiates an activity or series of actions. The other type of signal node is a *repetitive time event*. It means that this event occurs at a regular time intervals. In these cases a Petri token generator fits closely with this behavior. Alternatively, a single place, with tokens always available can be used ». [SPI 15] « ...C'est le nœud de *réception de signal*...Ce type de signal est toujours activé, d'où il démarre une activité ou une série d'actions. L'autre type de signal est le nœud de *temps d'attente*. Cela signifie que cet événement se produit à un intervalle de temps régulier. Dans ces cas, un générateur de jetons Pétri correspond étroitement avec ce comportement. Alternativement, une seule place, avec des jetons toujours disponibles peut être utilisé ». (paragraphe traduit)

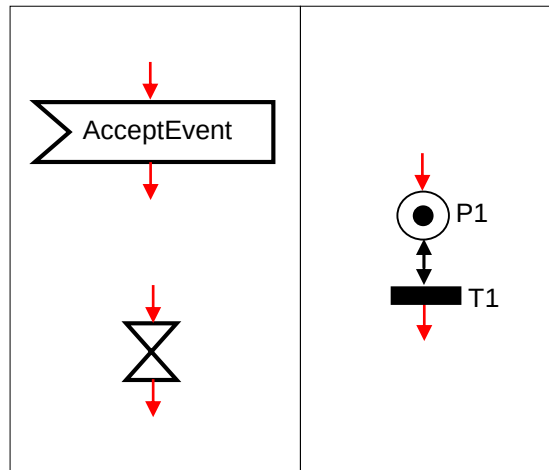


Figure.3.26. La notation de RdP équivalente au nœud de réception de signal et de temps d'attente.

V.13. Arc de flux de contrôle : [FOU 11]

❖ Un simple arc de RdP pourrait sembler suffisant.

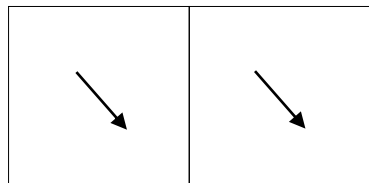


Figure.3.27. La notation de RdP équivalente à un arc de flux de contrôle.

V.14. Arc de flux d'objet : [FOU 11]

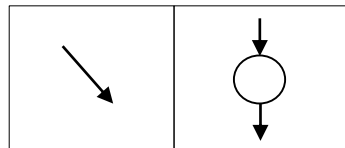


Figure.3.28. La notation de RdP équivalente à un arc de flux d'objet.

VI. Conclusion :

Dans ce chapitre, nous avons abordé les réseaux de Pétri, qui sont un outil graphique de modélisation, d'analyse et de vérification de nombreuses propriétés importantes, comme la vivacité, l'absence de blocage...etc et par plusieurs méthodes mathématiques, comme la méthode que nous avons adopté : la méthode de graphe de marquage.

Avant d'effectuer cette vérification pour monter la sûreté de fonctionnement de système modélisé, nous sommes dans l'obligation de passer d'un diagramme d'activité d'UML vers leur équivalent en réseau de Pétri, par l'utilisation d'un ensemble de règles de transformation. Ces règles de transformation, ne traitent pas tous les éléments offerts par le

diagramme d'activité du standard UML, tels que : la région d'expansion, le Handler (ou le gestionnaire d'exception)...

La modélisation des diagrammes d'activités d'UML, la transformation, la vérification et la simulation des réseaux de Pétri que nous avons abordées dans les chapitres précédents, seront implémentées dans notre application.

Chapitre 4

Implémentation

I. Introduction :

Dans ce chapitre, nous allons présenter l'objectif de notre travail, consistant à réaliser une application, pour la modélisation des processus métiers en Diagramme d'Activité d'UML, la vérification et la validation de leur bon fonctionnement on se basant sur les Réseaux de Pétri.

Notre application sert à :

- ❖ Dessiner le Diagramme d'Activité d'UML et implémenter les paramètres de chaque élément (les noms des nœuds et les conditions sur les arcs sortants de nœud de décision et les noms des couloirs);
- ❖ Définir les rôles de chaque collaborateurs;
- ❖ Transformer le Diagramme d'Activité en Réseau de Pétri ;
- ❖ Calculer les matrices pré-condition, post-condition et incidence, et extraire les caractéristiques du réseau de Pétri;
- ❖ Calculer le graphe des marquages accessibles et extraire les propriétés du réseau grâce aux techniques des théories des graphes ;
- ❖ La simulation du Réseau de Pétri.

En premier lieu, nous commencerons par une description de l'exemple d'application concernant un processus de « gestion d'une commande d'un produit », pour montrer les différentes étapes : modélisation, vérification, et simulation, en suite nous présenterons les différents outils nécessaires (le langage de programmation, la plateforme .NET et le .NET Framework) qui nous ont permis de mettre en œuvre cet application, et on terminera par l'implémentation de cette dernière.

II. Exemple d'application :

Pour ce projet, nous avons pris comme exemple « la gestion d'une commande de produits ». Cet exemple représente un processus métiers, qui se compose de plusieurs activités à réaliser par plusieurs participants qui coopèrent entre eux, tel que : le client, le service comptable et le service de livraison ; que nous allons les présenter par la suite ; afin de réaliser un but commun.

II.1. Le client : qui commande un produit et qui paie la facture.

II.2. Le service comptable : qui s'occupe de vérifier la disponibilité du produit commandé par le client, de traiter et de facturer la commande.

II.3. Le service de livraison : qui est responsable de préparer et de livrer la commande.

Le Diagramme d'Activité correspondant à cet exemple est illustré dans la **Figure 4.1**. Il répartit l'ensemble des actions du processus dans les trois partitions décrites auparavant, pour représenter les entités responsables, en utilisons le concept de « Swimlane » d'UML.

Le premier responsable qui est le « Client », démarre le processus par le lancement de la commande de produits et de la passer au « Service comptable ». Ce dernier, fait le calcul de prix après avoir vérifié la disponibilité des produits commandés dans le stock, et si c'est oui, cette action résulte un objet "Devis" qui sera démarré l'action d'acceptation de prix généré par le « Client ». Ici, le « Client » doit faire une décision, soit de valider la commande, ou de la modifier, ou de l'annuler, cette dernière décision, va arrêter toutes les activités du diagramme, et par conséquent la terminaison de processus.

Si le « Client » valide la commande, le « Service comptable » établir la facture, et en parallèle, il coopère avec le troisième participant qui est le « Service de livraison » pour préparer la commande. Ce parallélisme est représenté précisément en UML par le nœud de bifurcation.

Dans ce moment, le « Service de comptable », et après l'établissement de la facture, il l'envoie au « Client » qui devra la régler (paiement de la facture). Lorsque le « Service comptable » valide le paiement, et le « Service de livraison » prépare la commande (ici nous avons le mécanisme de synchronisation entre les deux services pour atteindre un but commun, ce mécanisme est représenté par le nœud d'union en UML), ce dernier livrer la commande au « Client », et le « Service comptable » clôture le dossier pour indiquer la fin de processus de « gestion d'une commande de produits ».

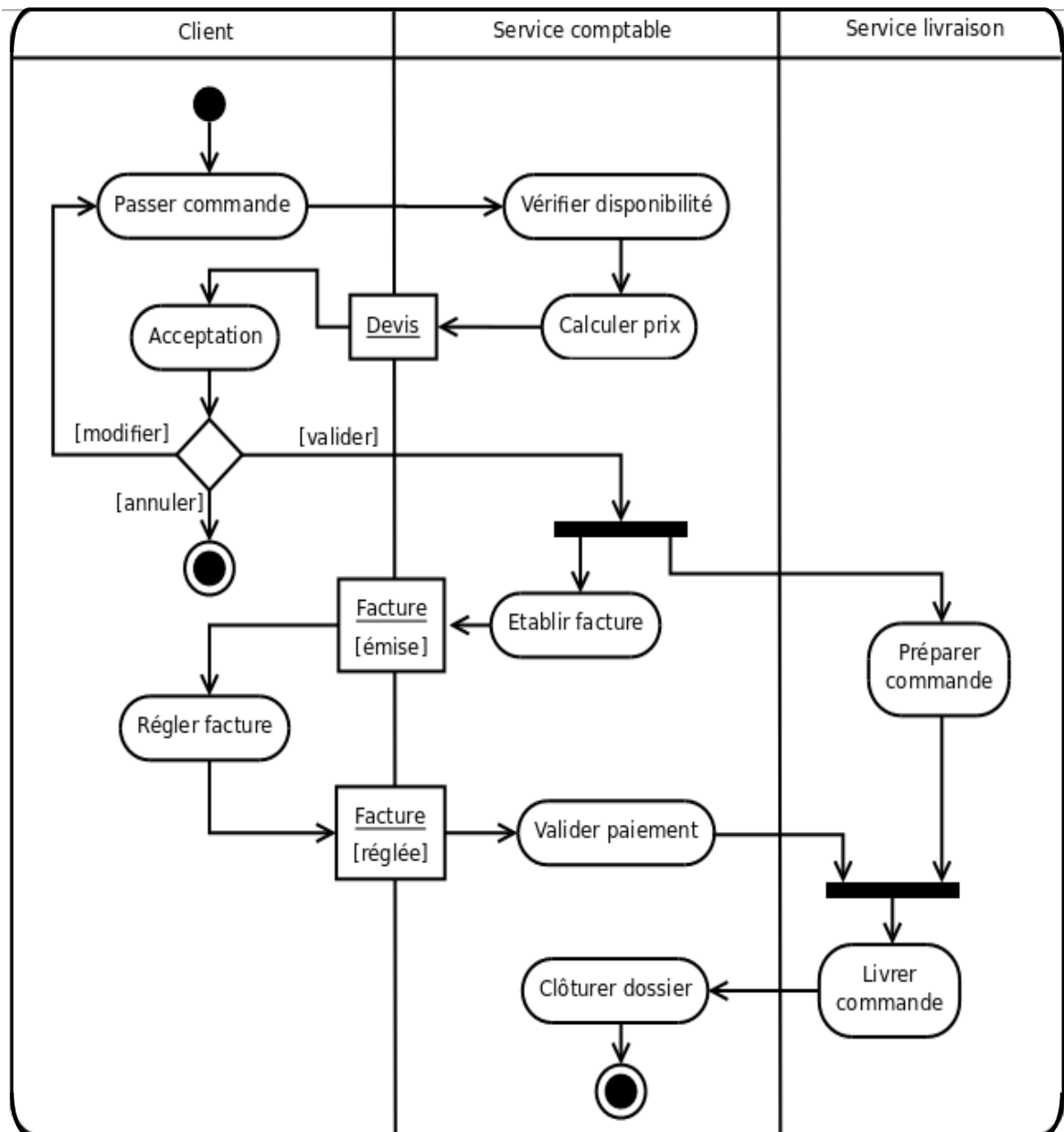


Figure.4.1. Exemple d'application de gestion d'une commande de produit. **[dev 16]**

D'après le concept de notre projet, d'avoir modélisés les processus métiers en Diagramme d'Activité d'UML, on doit passer de ce modèle au modèle Réseaux de Pétri, selon les règles de transformation qu'on a pu voir dans le chapitre précédent. La figure ci-dessous représente le Réseau de Pétri obtenue après la transformation du modèle original.

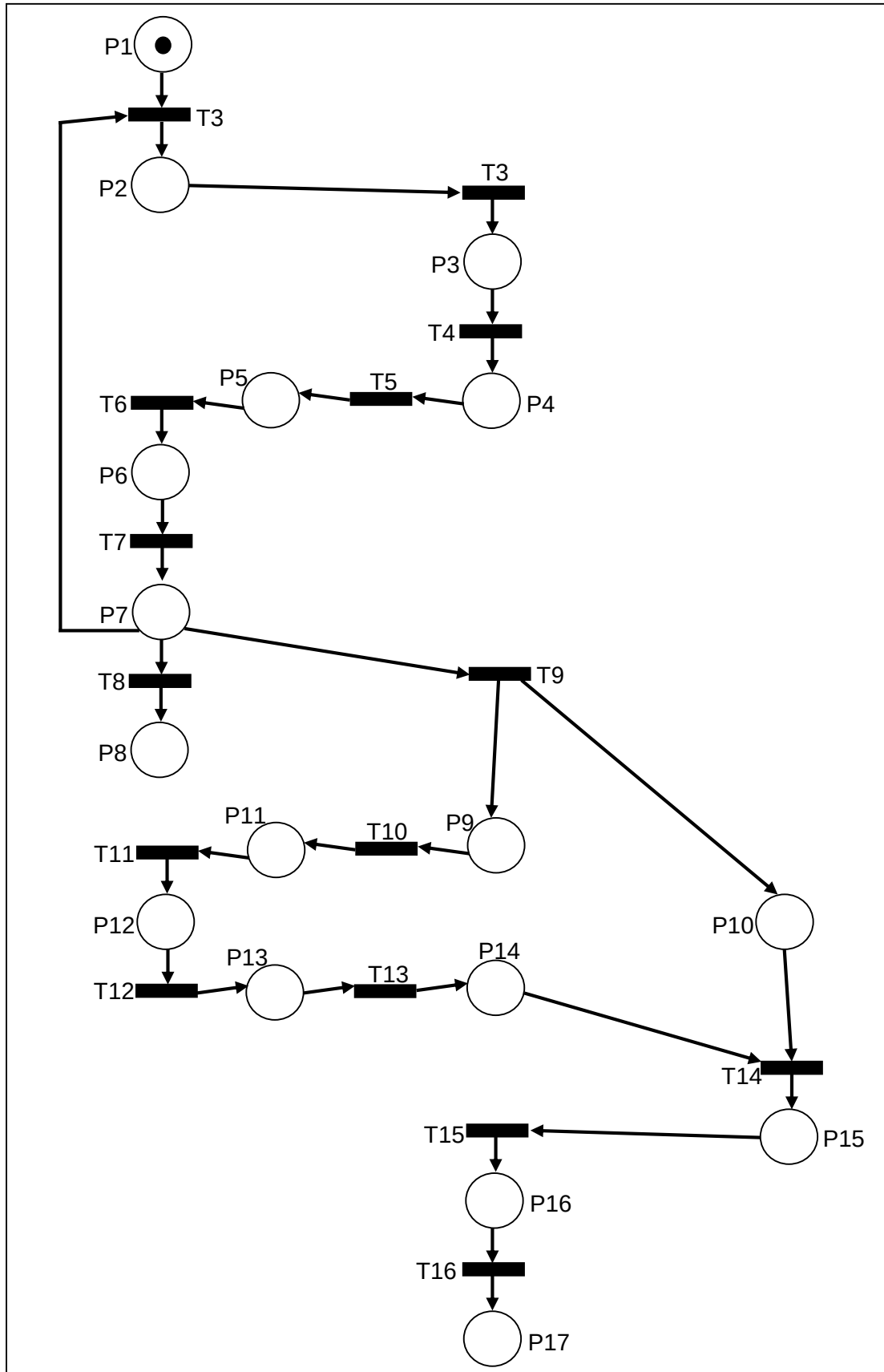


Figure.4.2. Le Réseau de Pétri obtenu après la transformation correspond à l'exemple d'application.

III. Les outils de développement :

III.1. Langage de programmation C#:

Dans notre projet nous avons utilisé le langage C# (écrit aussi C-Sharp). Le langage C# est un langage de programmation orienté objet (POO), qui a pour avantage de rendre une programmation complexe plus facile.

Le C# a été créé par Microsoft au début des années 2000. Ce langage est comme dit précédemment un langage de programmation orientée objet, étant un carrefour entre différents langages comme le Java, le C++ ou encore le Visual Basic, il ressemble plus au Java qu'au C ou au C++ contrairement à ce que son nom pourrait laisser penser. C'est un langage à typage fort, c'est à dire que le type des variables est fixé à la compilation et que celle-ci peut vérifier les erreurs de typage (à la différence du langage PHP par exemple). Le C-Sharp est très polyvalent, Il permet de coder de simples applications consoles jusqu'à de gros programmes avec une multitude de fenêtres, en passant par les jeux. [DRI 14].

III.2.Framework .NET :

Les programmes en C# s'exécutent sur le .NET Framework, qui est un espace de travail modulaire, il regroupe un ensemble d'outils pour le programmeur (accès réseau, création des fenêtres, appel à la base de donnée...). Un des avantages du Framework .Net, c'est qu'il est multi-langages. C'est-à-dire que l'on va pouvoir programmer en C#, en VB.NET, en C++, en J# et aussi en d'autres langages compatibles. [DRI 14]

III.3 L'environnement de développement :

L'environnement de développement de notre logiciel est Visual Studio 2010 qui nous permet de travailler et de développer sous la plateforme Framework.net. Cet outil est aujourd'hui en version 2015, et est disponible en plusieurs versions (Professional, Premium, Ultimate). Dans notre cas, nous avons travaillé avec la version **Microsoft Visual Studio 2010 Ultimate**.

Microsoft Visual Studio 2010 offre des assistants, des classes d'objets (dessin graphique, timers, etc.) prédéfinis, un debugger intégré, et beaucoup d'autres aides au développement d'applications pour : Windows, téléphones. Il offre aussi la possibilité d'adapter automatiquement des applications web à des appareils mobiles.

IV. Le logiciel « UML2RdP » :

Notre logiciel « UML2RdP » est développé en C#, destinée principalement aux utilisateurs qui gèrent les processus des entreprises afin de modéliser. Le travail du logiciel se déroule en cinq grandes parties :

1. La modélisation des processus métiers (workflows) en Diagramme d'Activité d'UML;
2. La transformation en Réseau de Pétri ;
3. La vérification de la structure du réseau ;
4. La vérification des propriétés du réseau ;
5. La simulation du réseau.

V. Description des interfaces du logiciel :

La réalisation d'une interface attractive aide beaucoup les utilisateurs à s'intégrer avec l'environnement de logiciel, car elle représente la partie visible du travail. Dans ce contexte, nous avons élaboré trois interfaces qui répondent aux besoins de leurs utilisateurs. Dans la partie suivante, on va essayer de décrire ces interfaces de notre logiciel.

V.1. L'accueil :

Au démarrage de logiciel « UML2RdP », une fenêtre s'affiche qui représente l'interface d'accueil de l'application, et qui donne la possibilité d'accéder à la deuxième fenêtre de notre application.



Figure.4.3. L'interface d'accueil du logiciel.

→ L'icône de bureau du logiciel « UML2RdP » représenté ci-dessous, permet de lancer l'exécution du logiciel.



Figure.4.4. L'icône du logiciel.

V.2. La deuxième interface (fenêtre de modélisation de D.A d'UML) :

L'interface illustrée ci-dessous, est celui qui a permis de modéliser les Diagrammes d'Activité d'UML. Elle est simple et facile à utiliser même par un utilisateur non expérimenté.

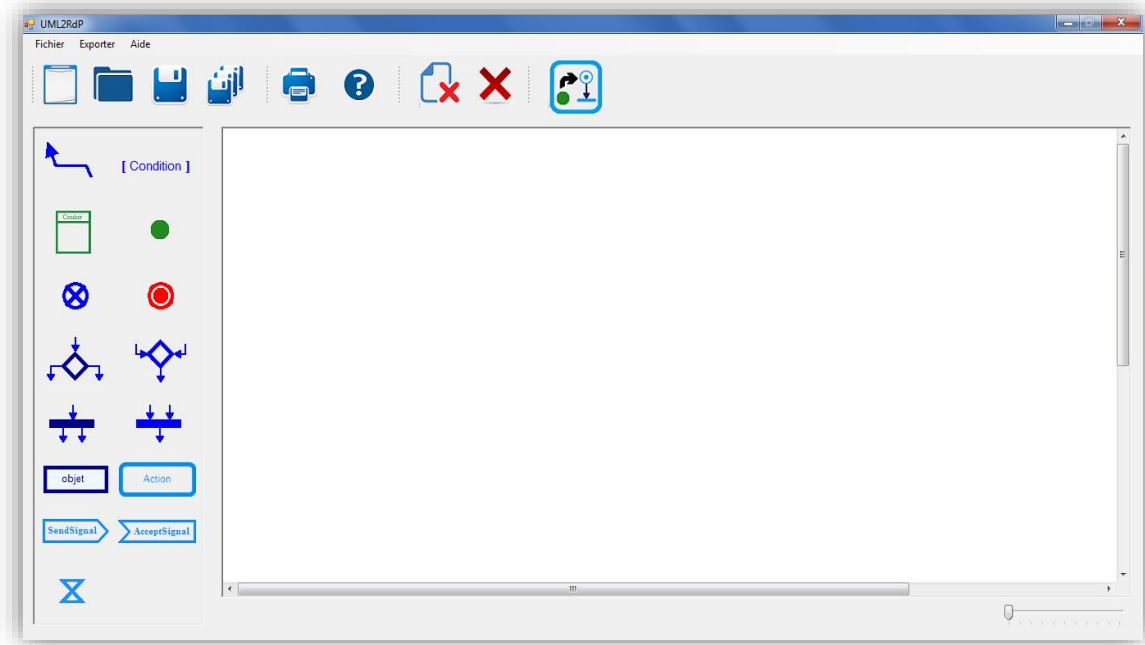


Figure.4.5. L'interface de modélisation des diagrammes d'activité d'UML du logiciel.

V.2.1. Ruban du logiciel :

V.2.1.1. Les menus :

Le ruban du logiciel contient trois menus : Fichier, Exporter, Aide.

1. A travers le menu « Fichier », on trouve :

- ❖ **Nouveau** : permet d'ouvrir une nouvelle zone de dessin pour créer un nouveau projet ;
- ❖ **Ouvrir** : permet d'ouvrir un fichier déjà existant à l'aide d'une boîte de dialogue ;
- ❖ l'enregistrement (**Enregistrer** et **Enregistrer sous**) des diagrammes d'activité modélisés qui est une tâche très importante sous l'extension « **.UML** » ;
- ❖ **Imprimer** : permet d'imprimer l'ensemble de la fenêtre du logiciel ;
- ❖ **Quitter** : permet de fermer le logiciel.

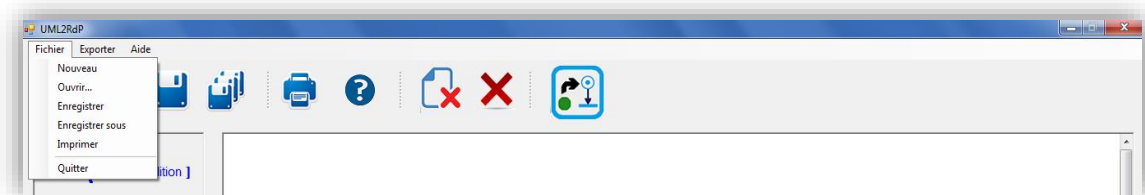


Figure.4.6. Le menu Fichier.

2. Le menu « Exporter » : Permet d'enregistrer le diagramme d'activité dessiné en image sous différents formats : « **.jpeg** », « **.gif** », « **.bmp** ».



Figure.4.7. Le menu Exporter.

3. Le menu « Aide » :

- ❖ **À propos :** c'est une boîte qui offre des informations concernons le logiciel comme les noms des développeurs...etc
- ❖ **Aide:** il fournit des pages HTML qui offrent un guide d'utilisation détaillé de ce logiciel.

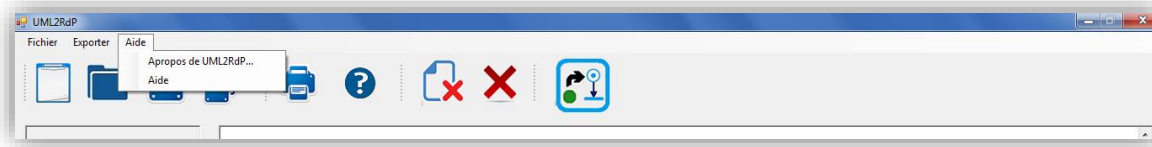


Figure.4.8. Le menu Aide.

V.2.1.2. La barre d'outils :

En plus des menus, le ruban du logiciel contient aussi une barre d'outils contenant un ensemble de groupes (quatre groupes), et chaque groupe à plusieurs boutons fonctionnels.

- ➔ Le premier et le deuxième groupe contient des boutons standards qui représentent des raccourcis de quelques taches (Nouveau, Ouvrir, Enregistrer, enregistrer sous, Imprimer, Aide).



Figure.4.9. Le premier et le deuxième groupe du ruban du logiciel.

- ➔ Le troisième groupe contient 2 boutons :

- ❖ **Le bouton supprimer :** il permet de supprimer un nœud ou un couloir une condition d'arc par un simple clic sur le bouton, puis sur l'un de ces trois objets.
- ❖ **Le bouton supprimer tous :** permet de supprimer l'ensemble de tous les éléments dessinés sur la zone de dessin dans le but de la réutilisée à nouveau.

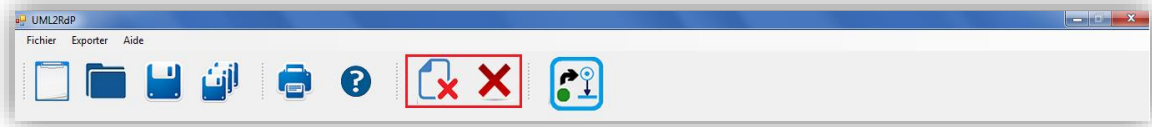


Figure.4.10. Le troisième groupe du ruban du logiciel.

- ➔ Le quatrième groupe contient un bouton très important qui est le bouton Transformer en RdP  : ce bouton est celui qui permet de passer à la troisième interface du logiciel. Il permet de transformer le diagramme d'activité d'UML dessiné en réseau de Pétri en se basant sur des règles de transformation prédéfinies.

V.2.2. La modélisation du dessin :

Pour la modélisation du dessin, ce logiciel offre une barre d'outils dans la partie gauche de la fenêtre, contenant les composants nécessaires du Diagramme d'Activité du standard UML pour modéliser le processus étudié. Ces composants sont mis à disposition de l'utilisateur pour modéliser les processus métiers de l'entreprise.

- ❖ **Le dessin des nœuds** : il suffit de faire un clic sur un des boutons des nœuds (il y a neuf types de nœuds et trois autres nouveaux définis en UML 2.0), puis cliquer sur la zone de dessin à l'emplacement choisi par l'utilisateur.
- ❖ **Le dessin d'un arc** : pour cela, il suffit de cliquer sur le bouton Arc et de lier les différentes formes dessinées. Notre logiciel permet de dessiner un arc composé de plusieurs lignes pour un but de garantir la lisibilité et de faciliter la modélisation pour l'utilisateur. Ceci se fait par des simples clics souris sur un des nœuds, puis sur la zone de dessin à l'emplacement choisi par l'utilisateur jusqu'à arriver au deuxième nœud. Comme on peut dessiner aussi un arc composé d'une seule ligne. Lorsque l'on relie un « nœud initial » avec un « nœud final de flux » ou un « nœud final d'activité », un message d'erreur apparaît pour éviter ce chevauchement. Et lorsque le premier clic est sur le vide ou sur un « nœud final de flux » ou un « nœud final d'activité » un autre message d'erreur va interdire ça, et même quand le deuxième clic est sur le « nœud initial » va être aussi traité par un message d'erreur.
- ❖ **Le dessin du couloir** : notre logiciel permet aussi de dessiner les couloirs pour définir les rôles, c'est la répartition des responsabilités de l'ensemble des actions du processus (c'est-à-dire : qui fait quoi ?), ce qui se fait par un simple clic sur le bouton Couloir, puis cliquer sur la zone de dessin où l'utilisateur veut le dessiner (c'est-à-dire : le clic doit être fait après l'ensemble des actions qui devront regrouper dans ce couloir).

- ❖ **La condition sur l'arc** : pour cela, une clique sur le bouton Condition, puis sur la zone de dessin auprès de l'arc de cette condition, est suffisant.

V.2.2.1. Paramétrage des objets :

Une fois le dessin effectué, il est possible de paramétrer chaque élément par l'intermédiaire d'une boîte de dialogue, ici on parle de l'affichage du nom des quelques nœuds comme le nœud d'Action, le nœud d'Objet, le nœud de Temps d'Attente, le nœud d'Envoi de Signal, le nœud de Réception de Signal, et aussi le nom de couloir et de la condition sur l'arc. On peut faire ça par un double-clic sur un objet.

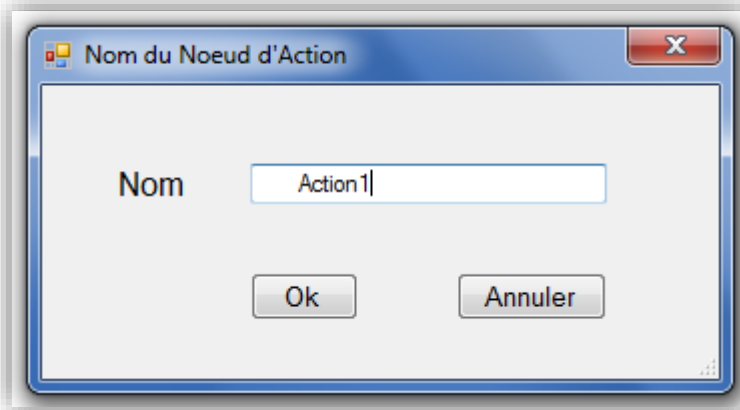


Figure.4.11. Le paramétrage des objets.

V.2.3. Le déplacement des objets :

Ceci permet de déplacer les nœuds ou le couloir ou la condition de l'arc lors des mouvements de la souris, en glissons puis en déposons (relâchons le bouton gauche de la souris).

V.3. La troisième interface (fenêtre de RdP):

Comme nous avons dit auparavant, qu'il y a un bouton important dans notre logiciel dans le quatrième groupe de la barre d'outils de la fenêtre de modélisation de D.A d'UML, qui est le bouton Transformer en RdP. Le clic sur ce bouton va nous permet de passer du premier modèle vers un Réseau de Pétri. Le résultat de ce passage est affiché dans la troisième fenêtre du logiciel afin d'analyser le réseau obtenu.

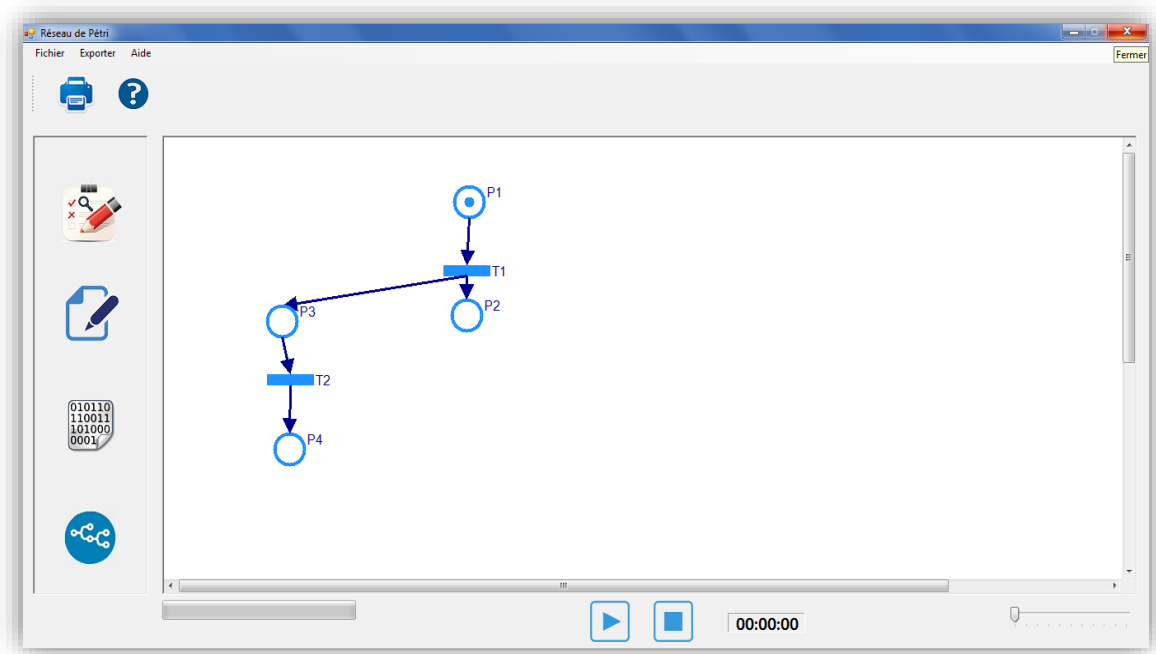


Figure.4.12. L'interface du RdP.

- Cette interface offre une barre des boutons dans sa partie gauche qui se compose de :
- ❖ **Le premier bouton** : permet d'afficher les propriétés du réseau suivantes dans une fenêtre afin de vérifier leur bon fonctionnement:
 - ➔ L'absence de blocage ;
 - ➔ Le caractère borné ;
 - ➔ Le réseau conforme ;
 - ➔ La vivacité ;
 - ➔ La pseudo-vivacité ;
 - ➔ La quasi-vivacité ;
 - ➔ La ré-initialisabilité.

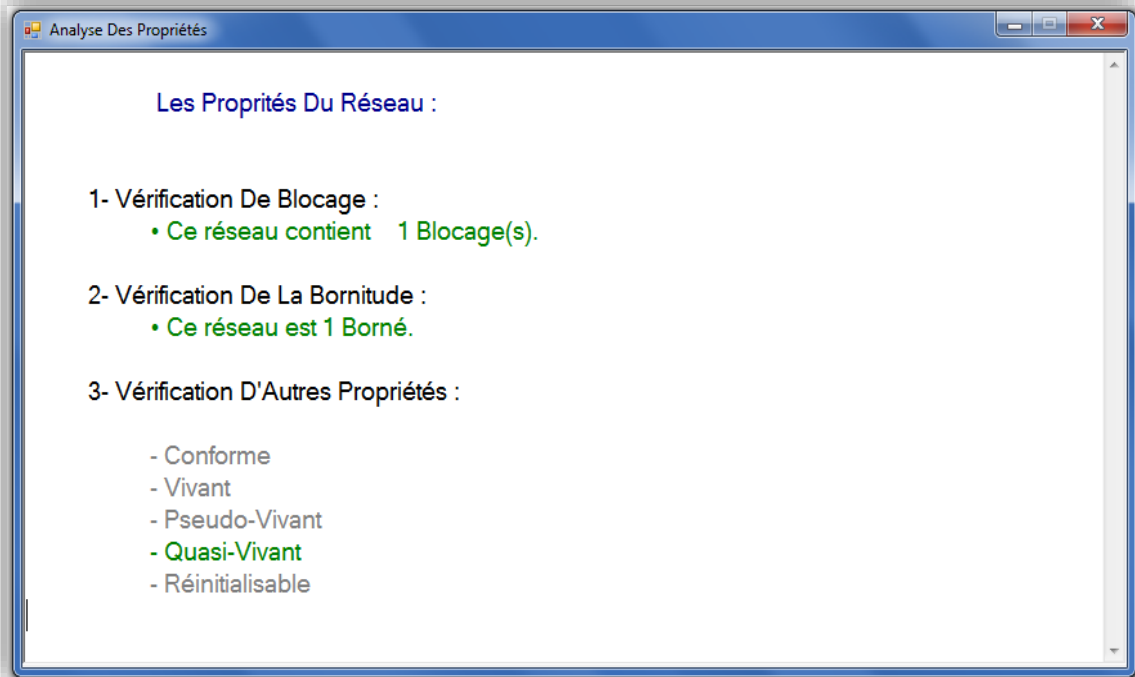


Figure.4.13. Les propriétés du RdP.

❖ **Le deuxième bouton** : permet de vérifier et d'afficher les caractéristiques reliés à la structure du réseau, comme :

- ➔ La vérification du conflit ;
- ➔ La vérification de la pureté ;
- ➔ La vérification de la simplicité;
- ➔ La vérification du choix libre.

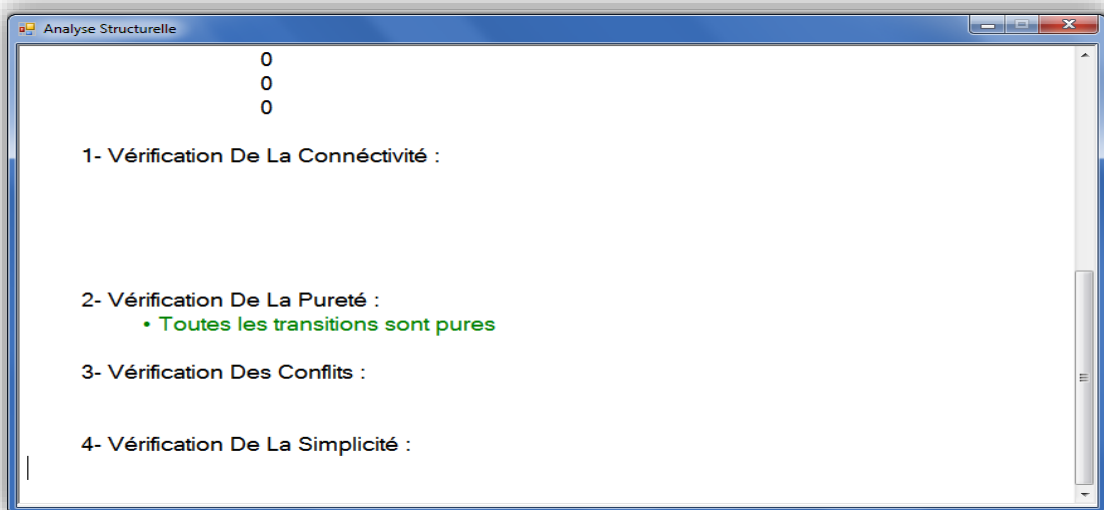


Figure.4.14. La structure du RdP -1-.

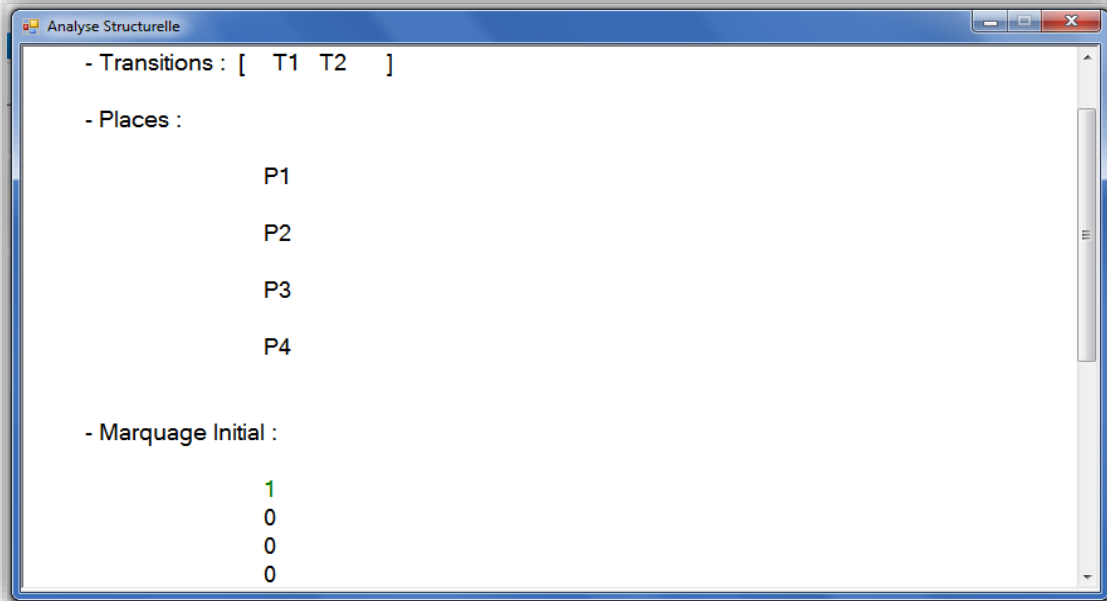


Figure.4.15. La structure du RdP -2-.

- ❖ **Le troisième bouton** : permet d'afficher dans une fenêtre, les matrices pré-condition, post-condition et la matrice d'incidence du réseau obtenu.

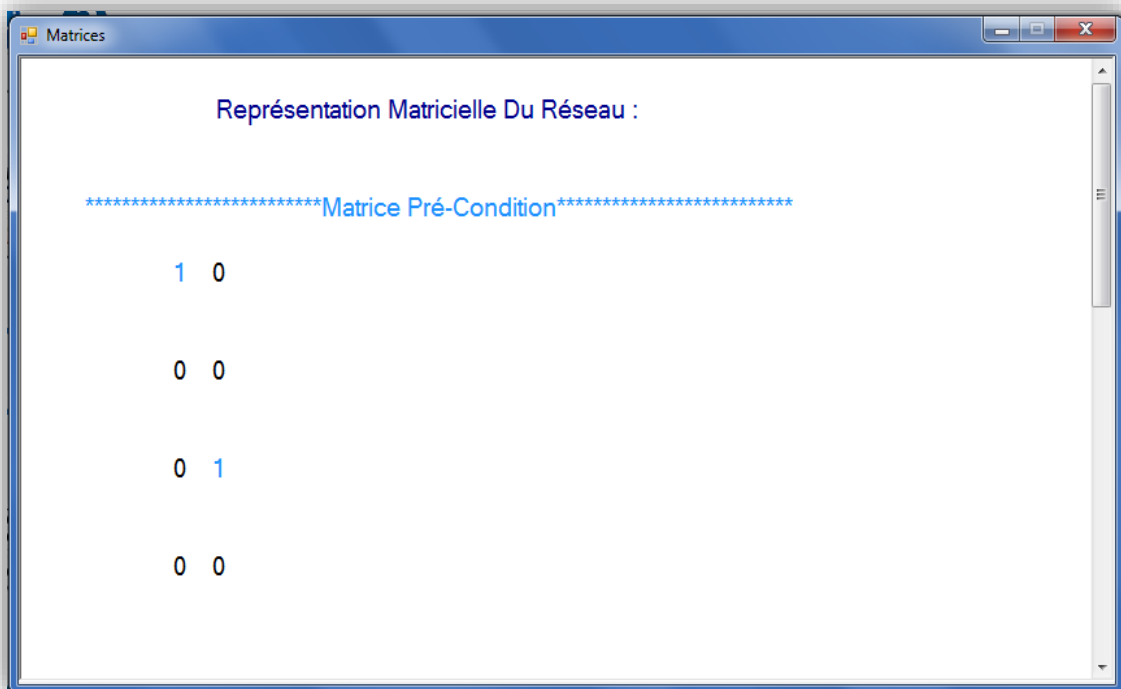


Figure.4.16. La représentation matricielle du RdP.

❖ **Le quatrième bouton** : permet d'afficher le graphe des marquages accessibles.

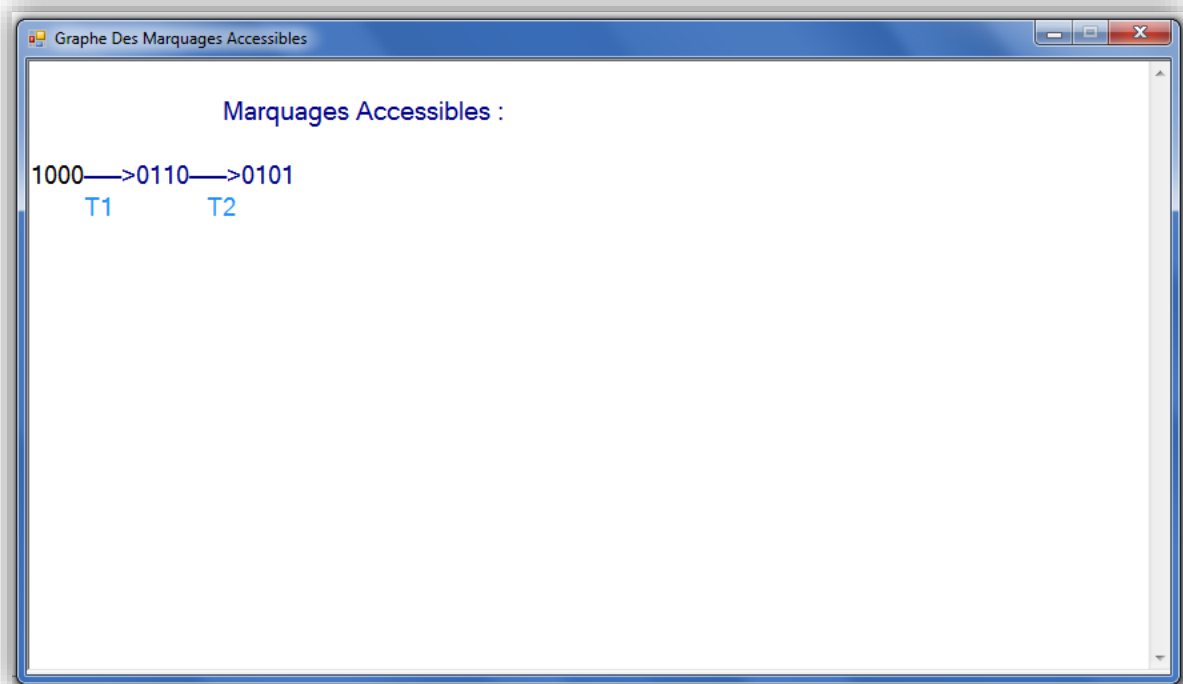


Figure.4.17. Le graphe de marquages accessibles du RdP.

V.3.1. Simulation du réseau : une fois le réseau construit, il reste à simuler son fonctionnement à l'aide du bouton Run.

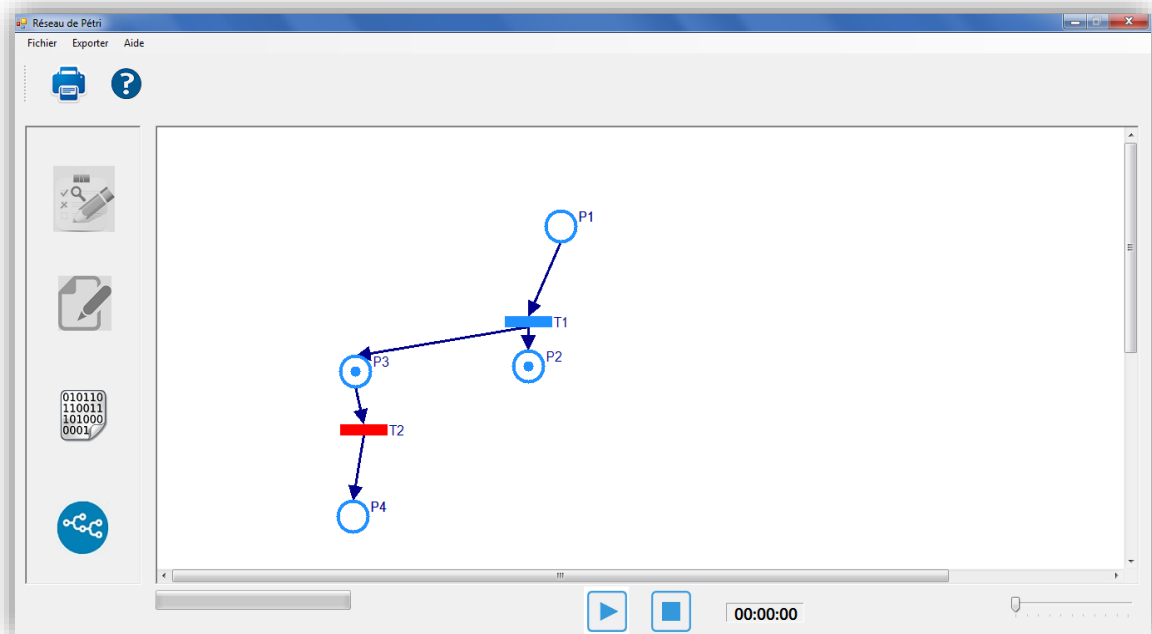


Figure.4.18. La simulation du RdP.

- ❖ **Le bouton Run**  : permet de détecter les transitions franchissables et les rendre en couleur rouge. Pour franchir une transition (ou bien déplacer les jetons), il suffit de cliquer dessus.
 - ❖ **Le bouton Stop**  : permet d'arrêter la simulation et réinitialiser le marquage.
- ➔ On trouve ces deux boutons en bas de la zone de dessin de la fenêtre de RdP.

VI. Conclusion :

Dans ce chapitre, nous vous proposons un outil pour la modélisation des différents processus métiers de l'entreprise en Diagrammes d'Activité du langage UML, en utilisant leurs différents éléments, il permet aussi de vérifier le bon fonctionnement du processus modélisé après l'avoir transformé en Réseau de Pétri. Il donne la possibilité de représenter le réseau obtenu sous forme de matrices et permet d'afficher leurs propriétés. Et enfin il offre la possibilité de simuler le fonctionnement du réseau.

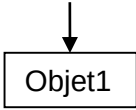
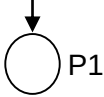
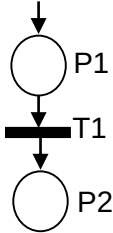
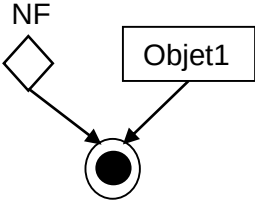
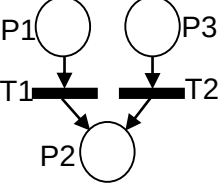
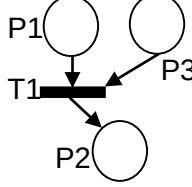
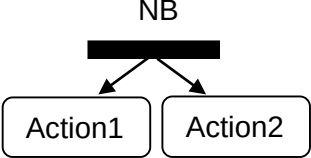
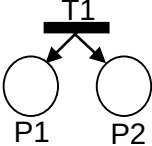
Conclusion générale et perspectives

La gestion des processus métier est un sujet important, sa pertinence du point de vue pratique vis à vie des nombreux défis pour les entreprises, a entraîné une vague de systèmes de gestion de workflow visant à l'automatisation de ces processus métier. Cependant, chacun de ces systèmes définissait sa propre façon de décrire ses processus tout en ignorant la vérification de leur cohérence, et par conséquent on a vite ressenti le besoin de mécanismes efficaces et d'outils de modélisation et d'analyse des processus workflow.

Pour cela, nous nous sommes fixé comme objectif dans se présent travail de développer un outil qui comporte le langage UML pour la modélisation workflow représenté par des Diagrammes d'Activités, la problématique consistera à convertir d'une façon automatique et transparente pour « l'utilisateur » ces Diagrammes d'Activités en Réseaux de Pétri pour les vérifier et l'analyser afin d'assurer le bon fonctionnement des systèmes modélisés.

Cependant, même si le travail présenté a abouti à une implémentation, un certain nombre de points restent à étudier, tel que :

- ❖ La modélisation d'autres diagrammes du langage UML ;
- ❖ La vérification de certaines propriétés comme : la répétitivité, réseau persistant, état d'accueil, atteignabilité. Et certains caractéristiques comme : le graphe d'état et graphe d'évènement.
- ❖ Et pour la transformation des modèles, nous avons rencontré plusieurs problèmes et ambiguïtés au niveau des règles de transformation.
 - ➔ Si on applique les règles qui permettent de passer du Diagrammes d'Activité d'UML vers les Réseaux de Pétri d'une façon directe, on obtiendra alors à la fin de la transformation, un ensemble de places et transitions qui n'ont aucun sens. Pour cela, on doit faire une optimisation des règles de transformation, pour que le réseau final soit compréhensible et nous permet de vérifier correctement son fonctionnement. Pour ce problème-là, nous avons essayé de résoudre le maximum des cas. Le tableau suivant résume les différents problèmes rencontrés lors du passage du D.A vers le RdP.

Exemple de D.A	Le RdP équivalent	Remarque
	Figure1  Figure2 	Dans ce cas, il y a une règle pour transformer l'arc de flux d'objet, qui dit qu'il faut le traduire en RdP en : Arc-Place-Arc. Et il y a une règle pour transformer le nœud d'objet en : Place seulement. Alors dans ce cas, et lors de la transformation de l'ensemble d'arc et nœud d'objet : est-ce qu'on doit fusionner les deux places (de l'arc et du nœud d'objet) en une seule place P1 (voir la Figure1) ? Ou, on doit rajouter une transition T1, comme il est monté dans la Figure2 ? Dans ce cas j'ai proposé d'utiliser la solution de la Figure 1.
	Figure1  Figure2 	Dans ce cas, est-ce qu'on doit rajouter la transition T2 pour le nœud d'objet (voir la Figure1) ? sachant que ce dernier est transformé en une seule place. Ou de lier la place P3 équivalente au nœud d'objet avec la transition T1 (voir la Figure2)?
		Dans ce cas, les deux Actions 1 et 2 sont gérés en parallèle par le nœud de Bifurcation. L'équivalent de ce diagramme en RdP est de créer une seule transition (ici T1) qui lance en même temps les deux places P1 et P2 pour respecter le principe défini en D.A.

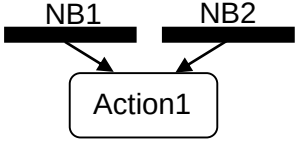
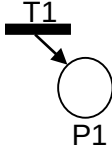
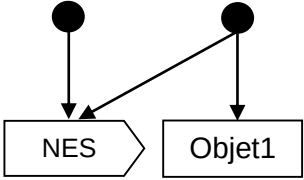
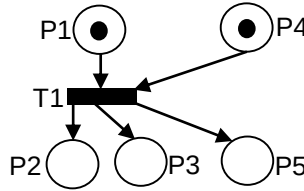
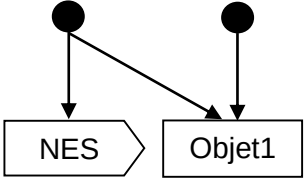
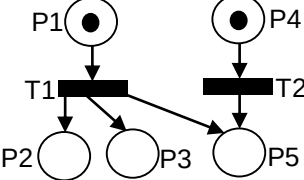
		Dans ce cas, deux nœuds de Bifurcation sont liés avec le même nœud d'Action. Alors en RdP, le NB est traduit par une seule transition, et le NA par Transition-Flèche-Place, (sachant que ce dernier nœud doit avoir une seule transition en entrée) → Pour cela, on a proposé de fusionner les transitions des nœuds de Bifurcation, pour respecter la règle de transformation du nœud d'Action.
		Dans ce cas, deux nœuds Initiaux sont liés avec le même nœud d'Envoi de Signal, et le deuxième nœud Initial est relié aussi avec un nœud d'Objet. Pour cela, on doit fusionner les deux transitions équivalentes aux nœuds Initiaux en une seule transition T1 pour respecter la règle de transformation de NES qu'elle s'agit d'une seule transition en entrée.
		Pour ce cas-là, c'est la même chose que le cas précédent, sauf que cette fois-ci c'est le nœud d'Objet qui est relié deux fois (par le nœud Initial 1 et 2). Il s'agit donc dans ce cas, de créer deux transitions T1 et T2 en RdP (sachant que le nœud d'Objet, comme nous avons cité auparavant, est transformé en Place seulement, c'est-à-dire : pas de transition, donc il n'est pas obligatoire de fusionner les deux transitions T1 et T2 équivalentes aux nœuds Initiaux en une seule transition).

Tableau.1. Ensemble des problèmes rencontrés lors du passage du D.A vers le RdP.

Bibliographie

- [AMA 02] Véronique AMAR, « Mise en œuvre d'outils de travail collaboratif à la Caisse d'Epargne Rhône-Alpes Lyon : Etude préalable à la mise en place d'un workflow et mise en ligne de la documentation du Système d'Information », Rapport de stage, Septembre 2002.
- [AOU 14] Mouna AOUAG, « Des diagrammes UML 2.0 vers les diagrammes orientés aspect à l'aide de transformation de graphes », Thèse pour l'obtention du diplôme de DOCTEUR 3ème cycle LMD de l'Université de Constantine 2, Option : Systèmes distribués, soutenue le 09 Octobre 2014.
- [BEN 10] Adla BENTELLIS, « Intégration des Applications d'Entreprises : Une approche basée objectif pour la gestion des processus métier flexibles », Thèse pour l'obtention du DOCTORAT en Sciences de l'Université Mentouri Constantine, spécialité : Informatique, pages 30-38, soutenue le 07 Février 2010.
- [BOU 13] Mouna BOUARIOUA, « Une approche basée transformation de graphes pour la génération de modèles de réseaux de Petri analysables à partir de diagrammes UML », Thèse pour l'obtention du diplôme de DOCTEUR 3ème cycle LMD en Informatique, de l'Université de Constantine 2, Option : Systèmes distribués, page 46, soutenue le 24 Avril 2013.
- [BOUA 12] Abdessamed BOUAICHA et Abdelatif OULD SI BOUZIANE, « Modélisation d'un processus de concurrent engineering : étude comparatif BPM / Workflow », Mémoire de fin d'études pour l'obtention du diplôme de MASTER en Informatique de l'Université de Abdelhamid Ibn Badis Mostaganem, Option : Ingénierie des Systèmes d'Information, pages 14-19-24, 2011 / 2012.
- [CRO 06] Yoann LE CROM, « Le travail collaboratif en entreprise », Livre blanc, pages 4-5, Juin 2006.
- [dev 16] Site web : <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>, Consulter Mai 2016.

- [DRI 14] Meriem DRISSI et Leila KADDOUR SEGHIR, « Conception d'un outil d'aide à la simulation la vérification et la validation de modèle des réseaux de Pétri », Mini Projet pour l'obtention du diplôme de LICENCE en Informatique, de l'Université de Abdelhamid Ibn Badis Mostaganem, pages 2-3-4-10-12, 2013 / 2014.
- [FOU 11] Damien FOURES, « Transformation des diagrammes d'activités SysML1.2 vers les réseaux de Pétri dans un cadre MDE », Rapport LAAS n°11864, Université Paul Sabatier - Laboratoire d'Analyse et d'Architecture des Systèmes, Groupe Ingénierie Système et Intégration, page 20, Juin 2011, soumis le 04 Décembre 2012 à ce site web : <https://hal.archives-ouvertes.fr/hal-00761053>
- [GLA 02] Olivier GLASSEY et Jean-Loup CHAPPELET, « Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML », UER: Management public / Systèmes d'information, pages 2-3, Novembre 2002.
- [GUE 11] Fayçal GUERROUF, « Une approche de transformation des diagrammes d'activités d'UML mobile 2.0 vers les réseaux de Pétri », Mémoire présenté en vue de l'obtention du diplôme de Magister en Informatique de l'Université de El Hadj Lakhdar – BATNA, Option : Système d'Information et de Connaissance (SIC), soutenue le 19 Avril 2011.
- [HAM 15] Houda HAMROUCHE, « Une Approche de transformation des diagrammes d'activité d'UML vers CSP basée sur la transformation de graphes », Mémoire présenté en vue de l'obtention du diplôme de MAGISTER en Informatique de l'Université de 20 AOUT 1955 – Skikda, Ecole Doctorale en Informatique de l'Est – EDI Est, Option : Génie Logiciel, pages 8-11-16-18-22, téléchargé le 6 Décembre 2015.
- [HAMA 13] Hadjer HAMAD et Soumia NADDOR, « Modélisation de processus métiers coopératifs à l'aide de BPMN : vérification et simulation par réseaux de Pétri », Mémoire de fin d'études pour l'obtention du diplôme de MASTER en Informatique de l'Université de Abdelhamid Ibn Badis Mostaganem, Option : Ingénierie des Systèmes d'Information, page 20, 2012 / 2013.
- [LAL 10] Colin LALOUE, « Apprentissages et couplages dans l'entreprise complexe : le cas de la conception collaborative dans le domaine aéronautique », Thèse pour obtenir le grade de DOCTEUR délivré par l'École nationale supérieure des Mines de Paris, spécialité : Sciences et Génie des Activités à Risques, soutenue le 20 Octobre 2010.

- [LAR 12] LAREDJ Mohamed Adnane, « Spécification d'un workflow pour la gestion d'interactions entre experts au sein d'un processus de maintenance coopératif », Thèse en vue de l'obtention du DOCTORAT de l'Université d'Oran, spécialité : Informatique, option : Informatique et Automatique, pages 64-73-74, 2011 / 2012.
- [MAL 15] Hugues MALGOUYRES & Jean-Pierre SEUMA-VIDAL & Gilles MOTET « Règles de cohérence UML 2.0 », Laboratoire d'Étude des Systèmes Informatiques et Automatiques, Institut National des Sciences Appliquées de Toulouse, département Génie Electrique, pages 93-94-101-107, téléchargé le 29 Décembre 2015.
- [PIQ 09] Alexandre PIQUET, « Guide pratique du travail collaboratif : Théories, méthodes et outils au service de la collaboration », document destiné au “Groupe Communication” du réseau Isolement Social, pages 9-13, Août 2009.
- [POT 07] Yvan POTIN, « Travail coopératif : quand la distance permet le rapprochement », CREG (Centre de Ressources en Économie-Gestion) de l'académie de Versailles, page 2, 2006 / 2007.
- [PRI 06] Yannick PRIÉ, « Introduction à la conception de systèmes d'information », UFR Informatique - Université Claude Bernard Lyon 1, M1 MIAGE - SIMA, page 6, 2005 / 2006.
- [ROD 09] Fabio RODA & Leo LIBERTI, « UML 2 Pratique de la modélisation », page 13, Juin 2009.
- [SAA 06] Ali SAADI, « Quelle méthode adopter pour modéliser les processus métier de l'administration? », Mémoire en vue de l'obtention du grade de MAITRE en sciences de l'Université de Laval, page 9, 2006.
- [SIN 13] SINI, « Méthodes et outils pour la gestion des workflow - Modélisation ontologique des processus pour l'analyse », Thèse en vue de l'obtention du DOCTORAT de l'Université de Mouloud Mammeri de Tizi Ouzou, Spécialité : Informatique, pages 10-13, soutenu le 07 Mars 2013.
- [SOU 13] Amen SOUISSI, « Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs », Thèse présentée pour obtenir le titre de DOCTEUR de l'Université des Sciences et Technologies de Lille I, spécialité : Informatique, page 25, soutenue le 20 Décembre 2013.

- [spa 15] Site web : http://www.sparxsystems.fr/resources/uml2_tutorial/uml2_activitydiagram.html, Consulter le 28 Décembre 2015.
- [SPI 15] Anthony SPITERI STAINES, « Intuitive transformation of UML2 activities into fundamental modeling concept Petri nets and colored Petri nets », Université de Malta, téléchargé le Novembre 2015.
- [TOU 07] Jihed TOUZI, « Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises », Thèse pour obtenir le grade de DOCTEUR de l'Institut National Polytechnique de Toulouse, spécialité : Systèmes Industriels, préparée au centre de Génie Industriel - Ecole des Mines d'Albi Carmaux, page 86, soutenue le 09 Novembre 2007.
- [ULM 11] Jean-Stéphane ULMER, « Approche générique pour la modélisation et l'implémentation des processus », Thèse en vue de l'obtention du DOCTORAT de l'Université de Toulouse, page 49, soutenue le 11 Février 2011.
- [VIG 10] Philippe VIGUIÉ, « Pilotage d'implémentations d'outils de TCAO au sein d'une organisation : une approche méthodologique », Thèse en vue de l'obtention du DOCTORAT de l'Université de Toulouse III - Paul Sabatier, spécialité : Informatique, pages 17-19, soutenue le 03 Novembre 2010.
- [ZEF 12] Sabrina ZEFOUNI, « Aide à la conception de workflows personnalisés : application à la prise en charge à domicile », Thèse en vue de l'obtention du DOCTORAT de l'Université de Toulouse III - Paul Sabatier, page 49, soutenue le 24 Septembre 2012.