



République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي والبحث العلمي
University ECHAHID HAMMA LAKHDAR - EL OUED
جامعة الشهيد حمه لخضر - الوادي



Mémoire de fin d'études

MASTER ACADEMIQUE

Domaine : Mathématiques et Informatique

Section : Informatique

Spécialité : Systèmes distribués et intelligence artificielle

Theme

Conception et développement d'une décision Réseau pour simuler L'environnement économique et social

Réalisé par :

Mr. Akram Mahmoudi

Mr. Eboubaker Bekkouche

Encadré par :

Mr. Mohammed Charaf Eddine Meftah

Promotion : 2024/2023

Dédicaces

“

*À mes chers parents ma mère et mon père Pour leur
patience, leur amour, leur soutien et leurs encouragement,
À mes frères, À mes amies et mes camarades,
Sans oublier tous les professeurs que ce soit du Primaires,
du moyen, La dédicace commence par la lettre alif et se
termine par la lettre nonne, du secondaire ou de
L'enseignement supérieur,
À tous ceux qui me sont chers, à vous tous
Merci.*

”

- **Akram**

“

*À ma mère, À mon père,
À mes frères et soeurs, À ma famille,
À mes amis, À mes collègues et À mes professeurs,
À mes lecteurs,
À tous ceux qui me sont chers, à vous tous
j'ai dédié ce travail.*

”

- **Eboubaker**

Remerciements

Avant tout nous remercions Allah le tout puissant qui nous a donné la force, la patience et le courage pour qu'on puisse accomplir ce modeste travail.

Nous remercions tout particulièrement notre encadreur, monsieur **Meftah Mohammad Charaf Eddine**, pour l'aide compétente qu'il nous a apportée, pour sa patience et ses encouragements à accomplir ce travail.

Nous exprimons nos reconnaissances à nos enseignants, nos amis, nos collègues de promotion système distribué et intelligence artificiel 2022

Je souhaite aussi remercier l'équipe pédagogique et administrative de Université Hama Lakhdar de El Oued pour leurs efforts dans le but de nous offrir une excellente formation.

Pour finir, je souhaite remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Abstract

In the modern world, the socio-economic environment has become increasingly complex. Previously, decision-making processes were relatively straightforward, but the current landscape, characterized by large volumes of data and intricate interactions among various organizations and institutions, has significantly complicated these processes. As a result, managers and decision-makers often struggle to foresee the long-term consequences and effects of their decisions.

This thesis aims to propose and design a decision network and a system for simulating the economic and social environment based on the proposed network. The system is intended to assist leaders in making informed and appropriate decisions by providing a simulated representation of the complex economic and social interactions.

Keywords : Decision, Network, System, socio-economic environment, Decision-Aid-Software

Résumé

Dans le monde moderne, l'environnement socio-économique est devenu de plus en plus complexe. Auparavant, les processus de prise de décision étaient relativement simples, mais le paysage actuel, caractérisé par de grandes quantités de données et des interactions complexes entre diverses organisations et institutions, a considérablement compliqué ces processus. En conséquence, les gestionnaires et les décideurs ont souvent du mal à prévoir les conséquences à long terme et les effets de leurs décisions.

Cette thèse vise à proposer et concevoir un réseau de décision et un système pour simuler l'environnement économique et social basé sur le réseau proposé. Le système est destiné à aider les dirigeants à prendre des décisions éclairées et appropriées en fournissant une représentation simulée des interactions économiques et sociales complexes.

Mots-clés : Décision, Réseau, Système, environnement socio-économique, Logiciel d'aide à la décision

Contents

Introduction Générale	1
1 Systèmes de Décision	2
1.1 Décision	3
1.1.1 Définition	3
1.1.2 Outils de Soutien à la Décision	3
1.1.3 Typologie des Décisions	3
1.2 Système de Décision	4
1.2.1 Définition	4
1.2.2 Utilisations des Systèmes de Prise de Décision	4
1.2.3 Applications des Systèmes de Prise de Décision	4
2 Réseaux de Décision	6
2.1 Définition	7
2.2 Modèle Graphique Probabiliste	7
2.3 Types de Réseaux de Décision	8
2.3.1 Arbres de Décision	8
2.3.2 Diagrammes d’Influence	10
2.3.3 Réseaux Bayésiens	10
2.3.4 Processus de Décision de Markov (MDPs)	11
2.3.5 Processus de Décision de Markov Partiellement Observables (POMDPs)	12
2.3.6 Réseaux de Décision Dynamiques (DDNs)	13
2.3.7 Différences entre les Réseaux de Décision	15
3 Spécification et modélisation mathématique du réseau proposé	17
3.1 Introduction	18
3.2 Architecture et caractéristiques du réseau proposé	18
3.2.1 Environnement économique et social comme réseau :	18
3.2.2 Caractéristiques fonctionnelles	20
3.2.3 Caractéristiques Techniques	22
3.2.4 Modélisation Mathématique du Réseau Proposé :	24
4 Conception du Système de Simulation basé sur le réseau de décision proposé	28
4.1 Introduction	29
4.1.1 Architecture générale du système proposé pour le réseau de décision proposé	29
4.1.2 Caractéristiques de Notre Réseau et Programme :	31

4.1.3	Comparaison avec les Outils Existant	31
4.1.4	Explication de la structure de l'application et des étapes :	33
4.1.5	Le Langage UML	48
4.1.6	Avantages et Inconvénients de UML	48
4.1.7	Diagrammes de cas d'utilisation	48
4.1.8	Diagrammes d'activité	51
4.1.9	Diagramme de séquence	53
4.1.10	Diagramme de classes	55
4.2	Conclusion	57
5	Réalisation du réseau de décision proposé	58
5.1	Introduction	59
5.2	L'environnement de développement	60
5.2.1	L'environnement Machine	60
5.2.2	L'environnement logiciel	61
5.3	Technologies Utilisées	65
5.4	Interfaces Graphiques	71
5.4.1	Écran Principal de l'Éditeur	71
5.4.2	Ajout de nœud	71
5.4.3	Modifier le nœud	73
5.4.4	Enregistrer le nœud modifié	74
5.4.5	Réseau final à exécuter	75
5.4.6	Rapport d'exécution	76
5.5	Conclusion Générale	77

List of Figures

1.1	Applications des Systèmes de Prise de Décision	5
2.1	Analyse descriptive, diagnostique, prédictive et prescriptive avec les réseaux bayésiens	7
2.2	Un réseau bayésien simple, connu sous le nom de réseau Asie	8
2.3	L'arbre de décision apprend la courbe d'une fonction sinus	9
2.4	Diagramme d'influence montrant les relations entre le produit, le prix, la stratégie marketing, la demande du marché, les unités vendues, le chiffre d'affaires, les coûts et les bénéfices	10
2.5	Réseau Bayésien pour calculer la probabilité d'être en retard en utilisant différentes variables	11
2.6	Exemple de MDPs	12
2.7	Un réseau bayésien dynamique simple	14
3.1	Environnement économique et social comme réseau	19
3.2	Réseau comme ensemble de : Nœuds et Liens	19
3.3	Nœud (ou sommet)	19
3.4	Liens (relations ou arêtes)	20
4.1	Architecture générale du système proposé pour le réseau de décision proposé	30
4.2	Réseau de Décision de Tarification des Produits	36
4.3	Scénario d'Exécution Avant (Les Nœuds de Quantité et de Prix de Base sont sélectionnés pour l'exécution)	39
4.4	Scénario d'Exécution Arrière (Le Nœud de Prix Final est exécuté)	40
4.5	Exécution en Boucle dans un Système de Gestion des Stocks	42
4.6	Exécution de Boucle avec Facteur Temps dans un Système de Gestion des Stocks	47
4.7	Diagramme de cas d'utilisation général	49
4.8	Diagramme d'activité : Cycle de vie d'un nœud.	52
4.9	Diagramme de séquence d'exécution de nœud	54
4.10	Diagramme de classes.	56
5.1	Interface de la plateforme Draw.io.	61
5.2	Interface de Visual Studio Code.	62
5.3	Interface de IntelliJ IDEA Ultimate	63
5.4	Interface de XAMPP.	64
5.5	Logo MySQL.	65
5.6	Logo TailWind CSS.	66
5.7	Logo de Javascript.	67

5.8	Logo NodeJS	68
5.9	Logo Vue.js.	69
5.10	Logo NPM.	70
5.11	Éditeur principal de l'application	71
5.12	Après l'ajout de 1 nœud	72
5.13	Modifier le nœud	73
5.14	Enregistrer le nœud modifié	74
5.15	Réseau complet	75
5.16	Rapport d'exécution pour 2 exécutions de simulation	76

Introduction Générale

L'environnement socio-économique est devenu plus complexe dans le monde moderne. Autrefois, la prise de décision était relativement facile, mais dans le monde moderne, la prise de décision est devenue plus difficile parce que les données sont devenues volumineuses, les organisations et les institutions se sont compliquées et les interactions entre elles sont devenues complexes. Ainsi, le processus de prise de décision est devenu difficile. Par exemple, un manager peut prendre une décision. Mais en raison de la complexité de l'environnement, il ne connaîtra pas les conséquences et les effets à long terme de ses décisions.

Le but de cette thèse est de proposer et de concevoir un réseau de décision et un système pour simuler l'environnement économique et social basé sur le réseau proposé. Ce système simule l'environnement économique et social pour aider à prendre la décision appropriée. Le système simulera l'environnement économique et social pour aider les dirigeants à prendre les décisions appropriées.

Pour résoudre ce problème, nous posons les questions suivantes :

- Quel est ce réseau et comment le concevoir et le créer ?
- Quelles sont les caractéristiques de ce réseau complexe ?
- Quelles sont les organisations ou entités ?
- Comment ces entités sont-elles représentées au sein de ce réseau ?

Pour répondre à ces questions, nous organiserons cette thèse en quatre chapitres comme suit :

- **Premier Chapitre** : Définition des Systèmes de Décision, leurs utilisations et applications.
- **Deuxième Chapitre** : Définition des Réseaux de Décision et comparaison de leurs différents types.
- **Troisième Chapitre** : Spécification et modélisation mathématique du réseau proposé.
- **Quatrième Chapitre** : Conception du Système de Simulation basé sur le réseau de décision proposé.
- **Cinquième Chapitre** : Réalisation du réseau de décision proposé.

Chapter 1

Systèmes de Décision

1.1 Décision

1.1.1 Définition

Une décision implique la sélection d'une option ou d'un plan d'action parmi un ensemble d'alternatives. Elle est le résultat d'un processus cognitif qui conduit à un choix final. [4]

La prise de décision dans l'entreprise est la méthodologie utilisée lors de la prise de décisions au sein d'une organisation ou d'une entreprise. La prise de décision dans une entreprise est le processus ou la méthode impliquée dans le choix d'une ou de plusieurs alternatives dans le but de résoudre un problème ou d'exploiter une opportunité tout en prenant en compte les effets que cela peut avoir sur les objectifs et les ressources de l'organisation. Ce processus est essentiel dans la planification stratégique et l'efficacité opérationnelle d'une entreprise, impliquant certains niveaux de gestion et parfois des parties prenantes. [8].

1.1.2 Outils de Soutien à la Décision

Ils sont de nature mathématique, statistique ou computationnelle.[8]

Outils de Soutien à la Décision Mathématiques ou Statistiques

- Utilisation de méthodes de résolution de planification : MPM, PERT, diagramme de Gantt
- Mise en œuvre de la théorie des graphes, gestion des flux (algorithme de Ford-Fulkerson)
- Utilisation de méthodes statistiques (espérance mathématique et écart type)
- Programmation linéaire
- Méthodes de remise en valeur
- Résolution de problèmes de flux
- Gestion des files d'attente
- Méthodes multicritères : matrice des résultats, arbre de décision, table de décision, espérance mathématique, méthode de théorie des jeux, ajustement...

Outils de Soutien à la Décision Computationnels

- Systèmes experts : Système d'Aide à la Décision (SAD).

1.1.3 Typologie des Décisions

Simon a adopté une analyse pour montrer comment les managers prennent des décisions, à cet égard il a proposé une typologie des décisions :

- **Décisions Programmables** : Décisions répétitives et quantitatives, où le problème est bien défini et l'information est disponible.

- **Décisions Semi-Programmées** : Présence de nombreux paramètres, informations partiellement fiables et difficiles à obtenir, nécessitant des outils de soutien à la décision.
- **Décisions Non-Programmables** : Degré élevé d'incertitude, informations nombreuses ou insuffisantes, reposant sur l'intuition et l'expérience des dirigeants.

1.2 Système de Décision

1.2.1 Définition

Les Systèmes de Prise de Décision font référence à des outils et cadres qui aident les individus, groupes ou organisations à faire des choix parmi des alternatives.

Ces systèmes sont conçus pour soutenir le processus de prise de décision en fournissant des informations pertinentes, en facilitant l'analyse et en prédisant souvent les résultats. Ils peuvent être largement catégorisés en deux types :

- **Systèmes de prise de décision basés sur l'humain** : Ceux-ci impliquent des processus et techniques qui améliorent les capacités de prise de décision humaine. Les exemples incluent les arbres de décision, les organigrammes et la méthode Delphi.
- **Systèmes de prise de décision basés sur l'informatique (SPDI)** : Ceux-ci utilisent des logiciels et des algorithmes pour analyser les données et prendre des décisions. Les types courants incluent les systèmes experts, les modèles d'intelligence artificielle et les systèmes de soutien à la décision basés sur les données.

1.2.2 Utilisations des Systèmes de Prise de Décision

Les systèmes de prise de décision sont utilisés pour :

- Améliorer la précision : En réduisant l'influence des biais et erreurs humains, ces systèmes aident à prendre des décisions plus précises.
- Accroître l'efficacité : Ils rationalisent le processus de prise de décision, réduisant le temps nécessaire pour recueillir des informations et analyser les options.
- Soutenir les décisions complexes : Pour les décisions impliquant de grandes quantités de données ou nécessitant l'intégration de diverses sources d'information, ces systèmes fournissent des capacités analytiques essentielles.
- Prédire les résultats : De nombreux systèmes de prise de décision avancés utilisent l'analyse prédictive pour prévoir les résultats de différents chemins de décision, aidant à une prise de décision plus éclairée. .

1.2.3 Applications des Systèmes de Prise de Décision

- **Gestion des Affaires** : Les entreprises utilisent les systèmes de prise de décision pour la planification stratégique, la gestion des risques et les décisions opérationnelles, telles que la gestion des stocks et la planification.
- **Santé** : Dans le domaine de la santé, ces systèmes aident au diagnostic, à la gestion des patients et à la planification des traitements, en intégrant des données médicales complexes et des modèles prédictifs.

Real-World Applications of Decision Support Systems

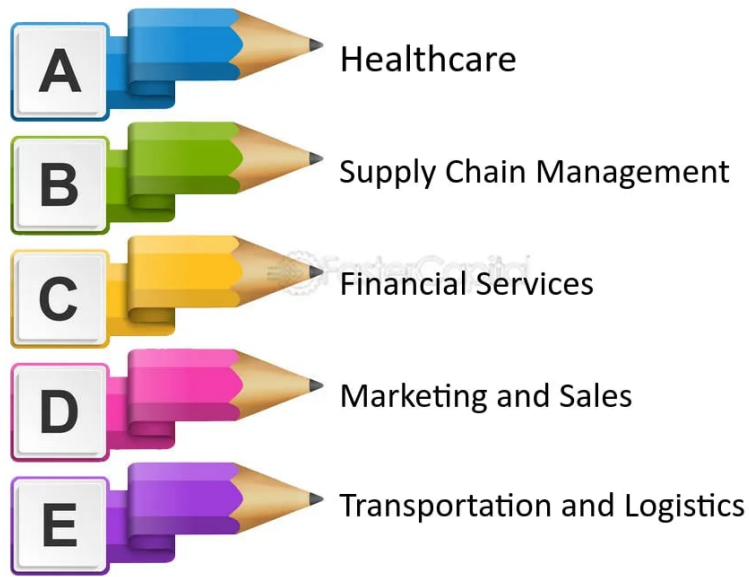


Figure 1.1 – Applications des Systèmes de Prise de Décision

- Finance : Les institutions financières utilisent ces systèmes pour l'évaluation du crédit, la gestion des risques, les décisions d'investissement et la détection des fraudes.
- Politique Publique : Les gouvernements appliquent les systèmes de prise de décision à la formulation des politiques, à l'allocation des ressources et à la gestion des crises, utilisant souvent des simulations et des analyses de scénarios pour prévoir l'impact de diverses politiques.
- Gestion Environnementale : Les systèmes de prise de décision sont utilisés pour gérer les ressources naturelles, prévoir les impacts environnementaux et planifier des initiatives de développement durable.

Chapter 2

Réseaux de Décision

2.1 Définition

Un **réseau bayésien**, également connu sous le nom de **réseau Bayes**, **réseau de Bayes**, **réseau de croyance**, ou **réseau de décision** [2], est un type de modèle graphique probabiliste représentant un ensemble de variables et leurs dépendances conditionnelles via un graphe acyclique dirigé (DAG), qui peut être utilisé pour construire des modèles à partir de données et/ou de l'opinion d'experts [3]. Les réseaux de décision représentent des informations sur :

- L'état actuel de l'entité.
- Ses actions possibles.
- L'état qui résultera de l'action de l'entité.
- L'utilité de cet état (également appelé, Diagrammes d'Influence) [10].

Ils peuvent être utilisés pour une très large gamme de tâches, notamment le diagnostic, le raisonnement, la modélisation causale, la prise de décision sous incertitude, la détection d'anomalies, l'analyse automatisée et la prédiction. La figure 2.2 ci-dessous montre ces capacités en termes des quatre principales disciplines analytiques, à savoir l'analyse descriptive, l'analyse diagnostique, l'analyse prédictive et l'analyse prescriptive, ainsi que l'IA causale.

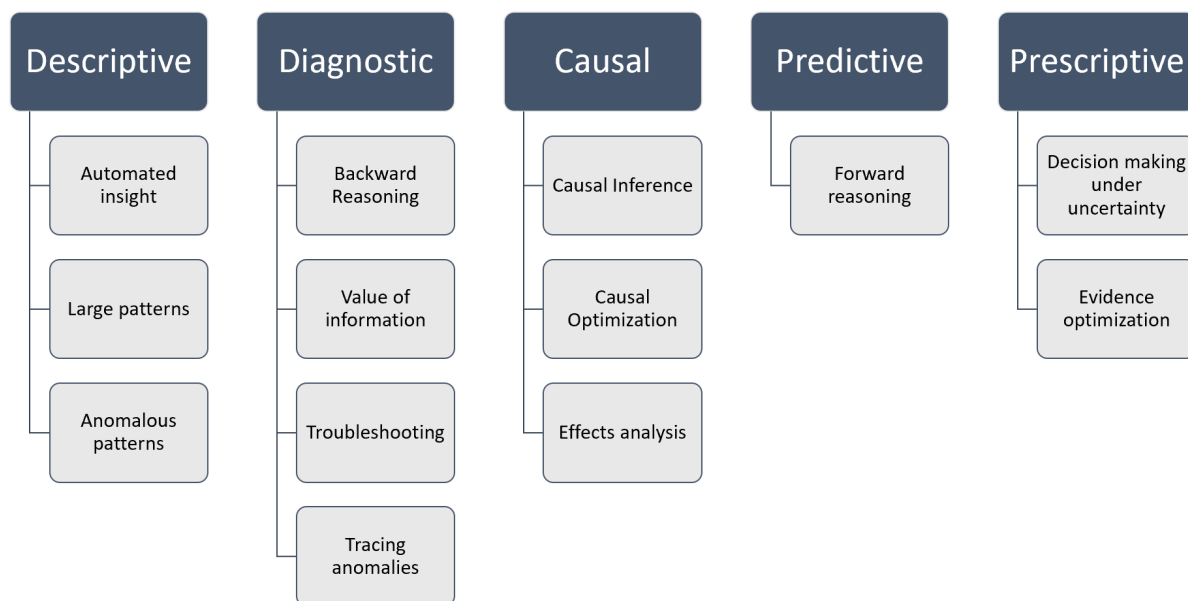


Figure 2.1 – Analyse descriptive, diagnostique, prédictive et prescriptive avec les réseaux bayésiens

2.2 Modèle Graphique Probabiliste

Les réseaux bayésiens sont probabilistes car ils sont construits à partir de distributions de probabilité et utilisent ensuite les lois de la probabilité pour le raisonnement, les diagnostics, l'IA causale, la prise de décision sous incertitude, etc. Les réseaux bayésiens peuvent être représentés graphiquement, comme le montre la figure 2.2 ci-dessous, qui illustre le célèbre *réseau Asie*. La visualisation est facultative mais constitue un bon moyen

de comprendre un modèle.

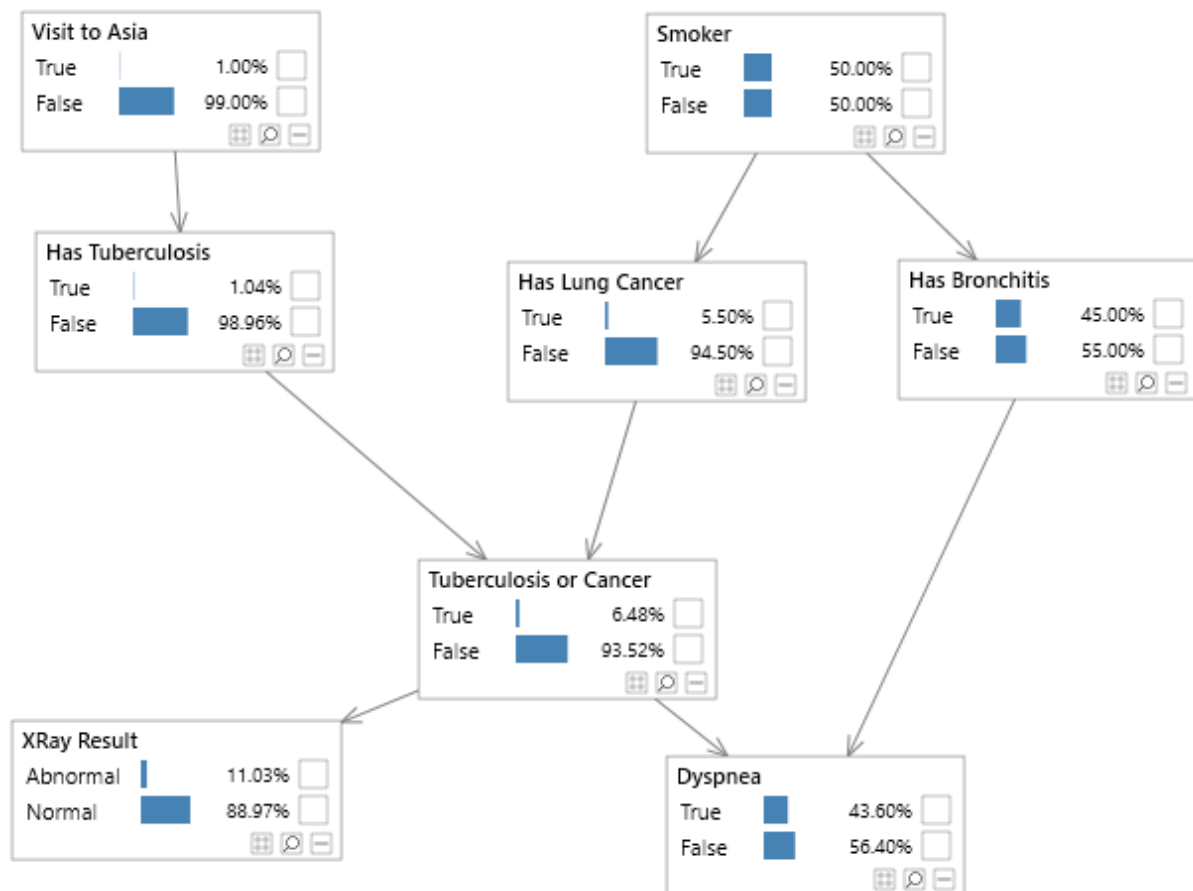


Figure 2.2 – Un réseau bayésien simple, connu sous le nom de réseau Asie

2.3 Types de Réseaux de Décision

2.3.1 Arbres de Décision

Les arbres de décision (DTs) sont une méthode d'apprentissage supervisée non paramétrique utilisée pour la classification et la régression. L'objectif est de créer un modèle qui prédit la valeur d'une variable cible en apprenant des règles de décision simples inférées des caractéristiques des données. Un arbre peut être vu comme une approximation constante par morceaux [9].

Quelques avantages des arbres de décision

- Simplicité de compréhension et d'interprétation.
- Nécessite peu de préparation des données.
- Capable de gérer des problèmes à plusieurs sorties.

Quelques inconvénients des arbres de décision

- Ils sont instables car de petites variations dans les données peuvent entraîner la génération d'un arbre complètement différent.
- Les prédictions des arbres de décision ne sont ni lisses ni continues, mais des approximations constantes par morceaux comme on peut le voir dans la *Figure 2.3*. Par conséquent, ils ne sont pas bons pour l'extrapolation.

Dans l'exemple ci-dessous, la *Figure 2.3* montre comment les arbres de décision apprennent à partir des données pour approximer une courbe sinus avec un ensemble de règles de décision du type si-alors. Plus l'arbre est profond, plus les règles de décision sont complexes et plus le modèle est ajusté.

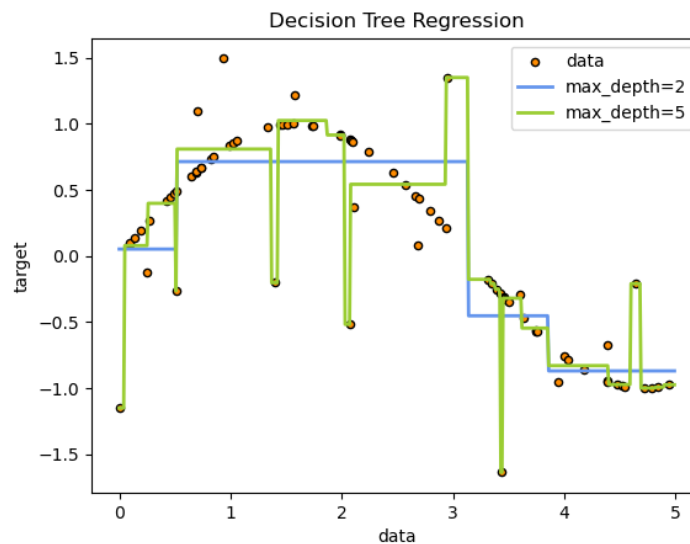


Figure 2.3 – L'arbre de décision apprend la courbe d'une fonction sinus

But des Arbres de Décision

Modéliser les décisions et les résultats possibles de manière claire et visuelle. Aide à prendre une décision en divisant les données en sous-ensembles basés sur un critère quelconque.

Application des Arbres de Décision

- Classification et Régression Utilisés en apprentissage automatique pour modéliser les prédictions des résultats en fonction des données d'entrée.
- Prise de Décision en Entreprise Aide à l'évaluation des différentes politiques d'entreprise et des résultats possibles de cette politique.
- Diagnostic Médical Utilisés par les médecins pour diagnostiquer les maladies en utilisant les données liées à un patient.

2.3.2 Diagrammes d’Influence

Les diagrammes d’influence sont une méthode de visualisation qui montre la relation entre les principaux éléments d’un cadre de décision [6].

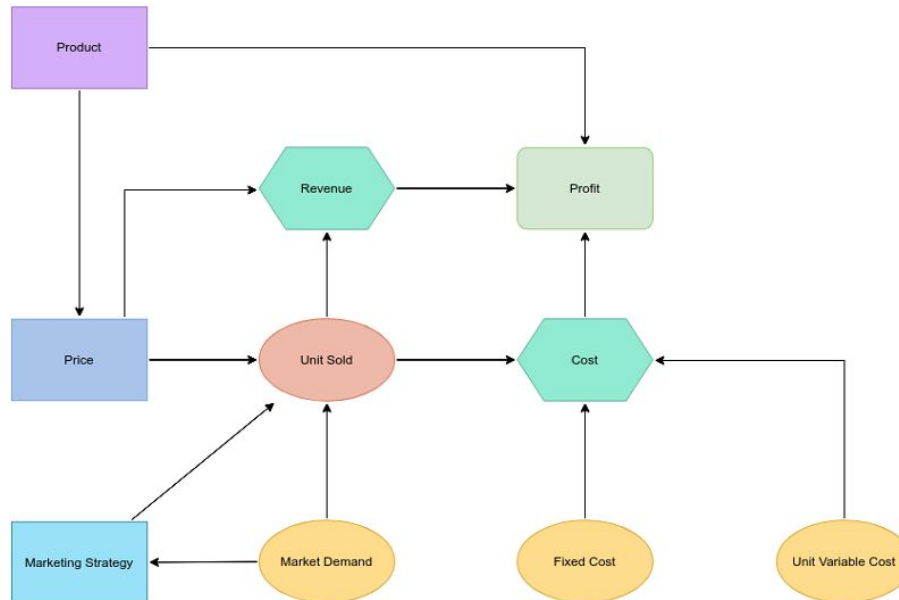


Figure 2.4 – Diagramme d’influence montrant les relations entre le produit, le prix, la stratégie marketing, la demande du marché, les unités vendues, le chiffre d’affaires, les coûts et les bénéfices

But des Diagrammes d’Influence

Modéliser et analyser les problèmes de décision pour montrer les relations entre les décisions, les incertitudes et les objectifs. Permettre de comprendre la structure des problèmes de décision complexes.

Applications des Diagrammes d’Influence

Planification Stratégique : Utilisée par les organisations pour planifier systématiquement les décisions et les impacts potentiels. Analyse des Politiques : Permet aux décideurs de regarder les implications possibles des différentes options politiques. Gestion de Projet : Les plans de projet aident à identifier les décisions critiques ainsi que les risques critiques auxquels un projet peut être confronté.

2.3.3 Réseaux Bayésiens

Un modèle graphique probabiliste qui représente un ensemble de variables et leurs dépendances conditionnelles via un graphe acyclique dirigé (DAG).

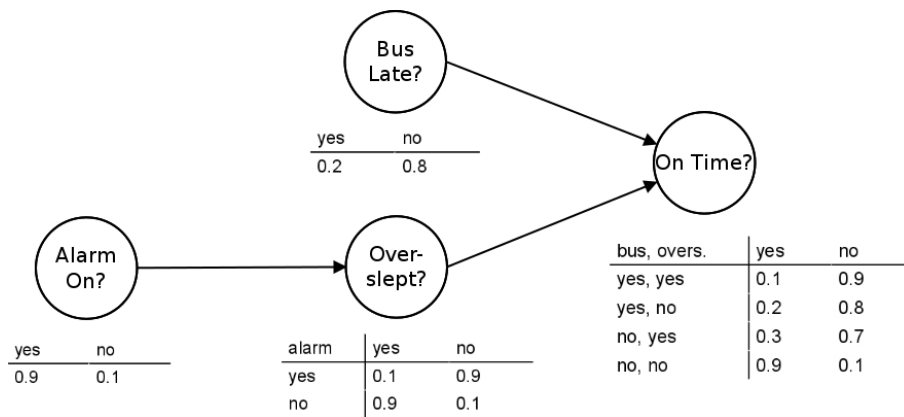


Figure 2.5 – Réseau Bayésien pour calculer la probabilité d’être en retard en utilisant différentes variables

But des Réseaux Bayésiens

Modéliser les relations entre les variables et inférer la probabilité de différents résultats. Fournir un cadre pour raisonner sous l’incertitude.

Application des Réseaux Bayésiens

Les applications vont de diagnostics médicaux à l’évaluation des risques dans des domaines aussi divers que la finance et l’ingénierie, en passant par la détection de fraudes basée sur la modélisation des comportements normaux et anormaux.

2.3.4 Processus de Décision de Markov (MDPs)

Un processus mathématique pour prendre une séquence de décisions dans un environnement stochastique afin de maximiser une récompense.

Example: Racing

- A robot car wants to travel far, quickly
- Three states: *Cool*, *Warm*, *Overheated*
- Two actions: *Slow*, *Fast*
- Going faster gets double reward

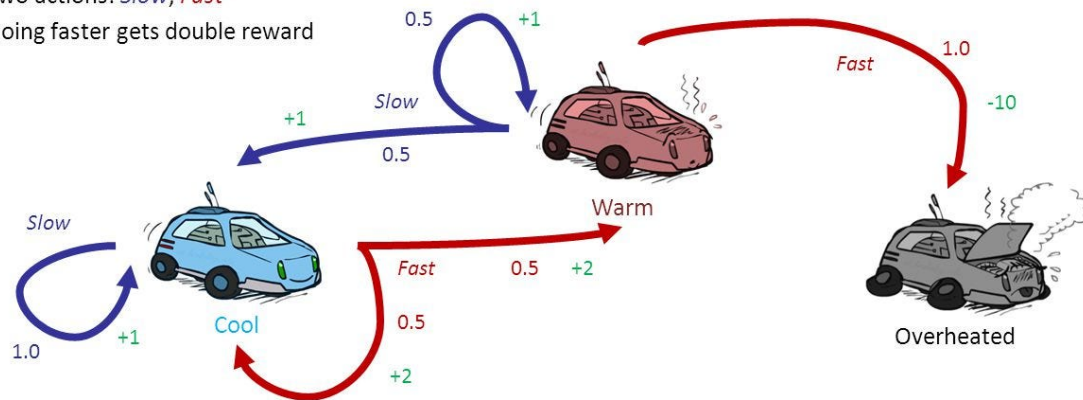


Figure 2.6 – Exemple de MDPs

But des Processus de Décision de Markov (MDPs)

Modéliser les relations probabilistes entre les variables et réévaluer la probabilité de différents résultats. Fournir un cadre pour raisonner sous l'incertitude.

Applications des Processus de Décision de Markov (MDPs)

- Diagnostic Médical : Cette fonctionnalité est utile pour prédire les maladies et les probabilités en fonction des symptômes saisis et de l'historique du patient.
- Évaluation des Risques : Il s'agit de l'identification et de la quantification des processus d'analyse des risques qui incluent la finance et l'ingénierie.
- Détection de Fraudes : Dans la détection de fraudes, il est utilisé pour développer un modèle des comportements normaux et anormaux des utilisateurs.

2.3.5 Processus de Décision de Markov Partiellement Observables (POMDPs)

Une extension des MDP qui tient compte de l'observabilité partielle, où l'agent n'a pas d'information parfaite sur l'état actuel.

But des Processus de Décision de Markov Partiellement Observables

Il peut modéliser des problèmes de prise de décision dans lesquels l'agent a une information incomplète sur l'état du monde.

Les MDP peuvent également optimiser les décisions sous incertitude concernant l'état du système et les résultats des actions.

Applications des Processus de Décision de Markov Partiellement Observables

Robotique : Navigation et prise de décision dans des environnements avec des informations incomplètes. **Santé :** prend des décisions pour le traitement dans la situation où l'état des patients est partiellement observable.

Planification en IA : peut être appliqué dans des systèmes d'IA avec une planification sous incertitude pour le jeu ou les techniques de conduite autonome.

2.3.6 Réseaux de Décision Dynamiques (DDNs)

Les réseaux bayésiens dynamiques sont des extensions du réseau bayésien standard qui intègrent le concept de temps. Les réseaux bayésiens dynamiques peuvent être utilisés lorsque nous devons modéliser une série temporelle - sa série ou séquence. Ils sont capables de modéliser des séries temporelles multivariées complexes car nous pouvons modéliser les relations entre de nombreuses séries temporelles dans le même modèle et également différents régimes de comportement, car les séries temporelles se comportent souvent différemment dans différents contextes. [1]

Caractéristiques des Réseaux Bayésiens Dynamiques

Certaines caractéristiques importantes des réseaux bayésiens dynamiques dans Bayes Server sont énumérées ci-dessous [1]:

- Prend en charge les séries temporelles multivariées (c'est-à-dire pas limitées à une seule série temporelle/séquence)
- Prise en charge des séries temporelles et des séquences, ou les deux dans le même modèle.
- Prise en charge de la détection d'anomalies
- Requêtes temporelles complexes telles que $P(A, B[t=8], B[t=9], C[t=8] \mid D, E[t=4])$
- Séquence la plus probable
- Prédiction, filtrage, lissage
- Variables temporelles latentes (discrètes et continues)
- Mélange de variables temporelles et non temporelles dans le même modèle
- Apprentissage des paramètres des modèles temporels
- Apprentissage structurel des modèles temporels
- Log-vraisemblance - utile pour la détection d'anomalies dans les séries temporelles

Noeuds temporels Un réseau bayésien dynamique peut se composer à la fois de noeuds temporels (de nature temporelle) et de ceux que l'on trouve dans un réseau bayésien standard. Ils prennent également en charge à la fois les variables continues et discrètes. Plus d'une variable qui représente des séries temporelles différentes mais (probablement) connectées peut exister dans le même modèle. Les dépendances entre elles peuvent être modélisées, ce qui conduit à des modèles capables de faire des prédic-

tions de séries temporelles multivariées. En d'autres termes, au lieu d'utiliser soit une seule série temporelle, soit de prendre une moyenne de toutes pour faire une prévision, nous pouvons utiliser de nombreuses séries temporelles et leurs interrelations pour faire des prévisions de meilleure qualité.

Noeuds initiaux et terminaux : Il existe quelques types spéciaux de noeuds dans un réseau bayésien :

- Noeuds initiaux : se connectent uniquement à des noeuds temporels à $t=0$.
- Noeuds terminaux : se connectent uniquement à partir de noeuds temporels à la dernière période (qui peut varier d'un cas à l'autre).

Liens temporels : Dans Bayes Server, un lien temporel est assez similaire à un lien standard dans un réseau bayésien, à quelques exceptions près :

- Il a un ordre associé (délai)
- Il ne peut lier que deux variables temporelles
- Il peut lier un noeud à lui-même

La Figure 2.7 ci-dessous montre un réseau bayésien dynamique simple avec une seule variable X . Il a deux liens, tous deux reliant X à lui-même à un point ultérieur dans le temps. Le premier a l'étiquette (ordre) 1, ce qui signifie que le lien connecte la variable X au temps t à elle-même au temps $t+1$. Le second est d'ordre 2, reliant $X(t)$ à $X(t+2)$. La Figure 3 montre le même réseau déroulé pour 5 tranches de temps, ce qui facilite la compréhension de la structure.

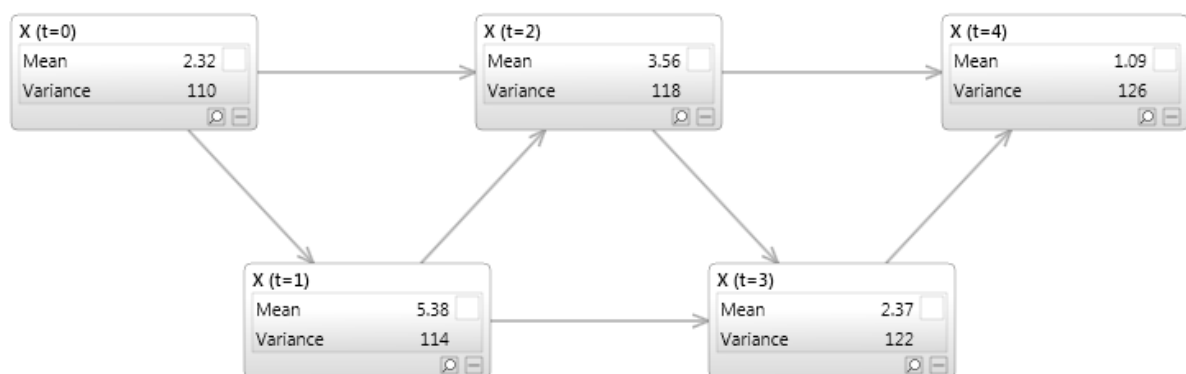


Figure 2.7 – Un réseau bayésien dynamique simple

But des Réseaux de Décision Dynamiques (DDNs)

Améliorer les diagrammes d'influence en incorporant l'élément du temps, permettant de modéliser des problèmes de prise de décision qui se déroulent sur des périodes. Considérer comment les décisions et les incertitudes dépendent les unes des autres dans le temps.

Applications des Réseaux de Décision Dynamiques (DDNs)

Les principales applications des DDNs sont :

Planification Financière : Aide à développer et à ajuster les stratégies d'investissement

dans le temps, aidant les investisseurs à prendre des décisions éclairées en fonction des conditions changeantes.

Épidémiologie : Utilisé pour modéliser la propagation des maladies et évaluer l'efficacité des différentes stratégies d'intervention dans le temps. Gestion de la Chaîne d'Approvisionnement : Soutient une meilleure planification et prise de décision pour la gestion des stocks et de la logistique, garantissant que les chaînes d'approvisionnement fonctionnent correctement dans le temps.

2.3.7 Différences entre les Réseaux de Décision

Voici les différences entre les réseaux de décision mentionnés précédemment :

Arbres de Décision

- **Structure** : Modèle en forme d'arbre de décisions et de leurs conséquences possibles.
- **Utilisation** : Souvent utilisé pour des tâches de classification et de régression.
- **Composants** : Noeuds (décisions ou résultats), branches (règles de décision) et feuilles (résultats finaux ou étiquettes de classe).
- **Avantages** : Simple à comprendre et à interpréter ; facile à visualiser.
- **Inconvénients** : Sujet au surajustement ; peut être instable avec de petits changements dans les données.

Diagrammes d'Influence

- **Structure** : Graphe acyclique dirigé avec des noeuds représentant des décisions, des incertitudes et des objectifs.
- **Utilisation** : Utilisé pour l'analyse décisionnelle afin de modéliser des scénarios de prise de décision complexes.
- **Composants** : Noeuds de décision, noeuds de chance (incertitudes) et noeuds de valeur (objectifs).
- **Avantages** : Fournit une représentation graphique claire du processus de prise de décision ; montre les dépendances entre les variables.
- **Inconvénients** : Peut devenir complexe et difficile à interpréter avec de nombreuses variables.

Réseaux Bayésiens

- **Structure** : Graphe acyclique dirigé où les noeuds représentent des variables aléatoires et les arêtes représentent des dépendances probabilistes.
- **Utilisation** : Utilisé pour modéliser des relations probabilistes entre les variables ; tâches d'inférence et de prédiction.
- **Composants** : Noeuds (variables aléatoires) et arêtes (dépendances conditionnelles).

- **Avantages** : Représentation efficace des distributions de probabilité conjointes ; peut gérer les données manquantes.
- **Inconvénients** : Intensif en calcul pour les grands réseaux ; nécessite une expertise en raisonnement probabiliste.

Processus de Décision de Markov (MDPs)

- **Structure** : Cadre mathématique pour modéliser la prise de décision où les résultats sont en partie aléatoires et en partie sous le contrôle du décideur.
- **Utilisation** : Utilisé pour les problèmes de prise de décision séquentielle ; couramment appliqué en apprentissage par renforcement.
- **Composants** : États, actions, probabilités de transition et récompenses.
- **Avantages** : Fournit une base formelle pour modéliser les processus de décision ; bien adapté aux environnements dynamiques.
- **Inconvénients** : Peut être coûteux en calcul pour résoudre des espaces d'états et d'actions importants.

Processus de Décision de Markov Partiellement Observables (POMDPs)

- **Structure** : Extension des MDPs où l'état du système n'est pas entièrement observable.
- **Utilisation** : Utilisé pour la prise de décision dans des environnements avec une incertitude sur l'état actuel.
- **Composants** : États, actions, probabilités de transition, probabilités d'observation et récompenses.
- **Avantages** : Gère l'incertitude dans l'information sur l'état ; adapté aux problèmes complexes du monde réel.
- **Inconvénients** : Encore plus intensif en calcul que les MDPs ; résoudre les POMDPs peut être très difficile.

Réseaux de Décision Dynamiques (DDNs)

- **Structure** : Extension des Réseaux Bayésiens pour modéliser des problèmes de décision séquentiels dans le temps.
- **Utilisation** : Utilisé pour la prise de décision dans des environnements dynamiques où le processus de prise de décision se déroule sur plusieurs étapes temporelles.
- **Composants** : Tranches temporelles des Réseaux Bayésiens, avec des noeuds représentant des variables à différents moments.
- **Avantages** : Capture les dépendances temporelles et les changements dans le processus de prise de décision au fil du temps.
- **Inconvénients** : Peut devenir très complexe et exiger beaucoup de calculs pour des horizons temporels longs.

Chapter 3

Spécification et modélisation mathématique du réseau proposé

3.1 Introduction

Le développement d'un système de prise de décision efficace nécessite une analyse et une modélisation soigneuses pour s'assurer qu'il fonctionne correctement et pour garantir qu'il répond aux besoins de tous les responsables.

Dans ce chapitre, nous procéderons à une analyse des besoins et à la modélisation du système de prise de décision à l'aide de diagrammes UML. Tout d'abord, nous identifierons le type de réseau de décision qui sera utilisé. Nous utiliserons ensuite des diagrammes UML pour modéliser les différents aspects du système, y compris ses composants, ses interactions et son comportement. Nous expliquerons également la structure et le fonctionnement du réseau.

L'objectif principal du réseau est de simuler l'environnement économique et social, car il s'agit d'un environnement de données (et de processus) vaste et complexe, difficile à étudier et à suivre de manière manuelle, car il inclut des données dynamiques, qui sont des données qui changent avec le temps en fonction des interactions et des influences entre les unités interactives (économiques et sociales), par exemple :

- Variables économiques : prix, production, demande, emploi, produit intérieur brut, taux d'inflation, chômage, investissements, commerce international, monnaies, et autres.
- Variables sociales : telles que le niveau de vie, les taux de mariage, de divorce, d'immigration, d'éducation, de santé, de culture, de politique, et autres.
- Variables environnementales : telles que les conditions météorologiques, les ressources naturelles, la pollution, les catastrophes naturelles, etc.
- Variables stratégiques : elles peuvent inclure divers développements sur le marché mondial, la technologie, la politique et la sécurité.
- Prévisions et analyses : elles peuvent inclure des estimations des attentes futures et des analyses du comportement passé et actuel pour orienter les décisions et l'élaboration des politiques.

3.2 Architecture et caractéristiques du réseau proposé

3.2.1 Environnement économique et social comme réseau :

Le concept de l'environnement économique et social peut être compris comme un réseau complexe de composants interagissant qui peuvent être modélisés comme un système de nœuds et de liens (relations). Dans ce réseau, les nœuds représentent les différents composants, tels que les organisations, les institutions ou les entités économiques, tandis que les liens représentent les interactions ou relations entre ces nœuds. Chaque nœud et lien est associé à un intervalle de temps spécifique, soulignant la nature dynamique de cet environnement.

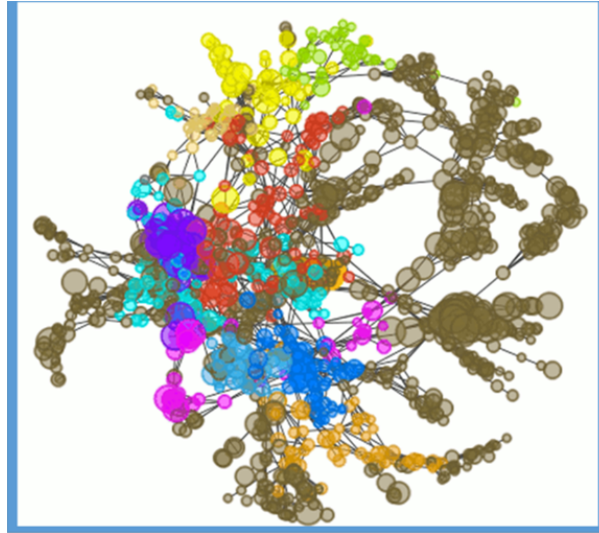


Figure 3.1 – Environnement économique et social comme réseau

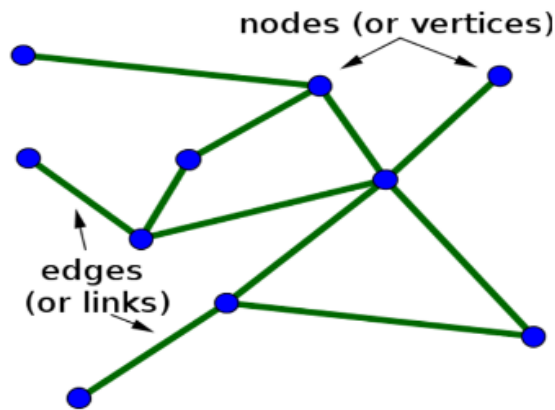


Figure 3.2 – Réseau comme ensemble de : Nœuds et Liens

Les nœuds (ou sommets) ont des entrées et des sorties, qui sont influencées par le flux de ressources, d'informations ou d'influence. La durée d'un nœud fait référence à l'intervalle de temps entre le moment où il reçoit ses entrées et celui où il génère ses sorties. Cette période peut varier en fonction de la nature du nœud et de son rôle au sein du réseau. Par exemple, une entreprise de fabrication peut avoir des intrants sous forme de matières premières et des extrants sous forme de produits finis, la durée représentant le cycle de production.

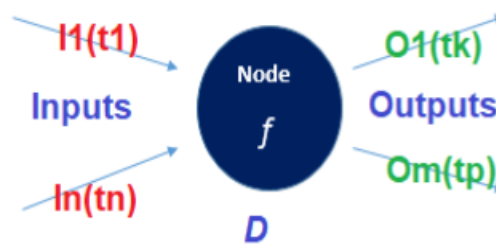


Figure 3.3 – Nœud (ou sommet)

Les liens (relations ou arêtes) De même, les liens (relations) entre les nœuds sont caractérisés par le temps nécessaire pour qu’une sortie d’un nœud devienne une entrée pour le nœud connecté. Cette durée du lien reflète le temps de transfert des biens, services, informations ou autres formes d’interaction. Par exemple, dans une chaîne d’approvisionnement, la durée du lien pourrait représenter le temps de livraison entre un fournisseur et un fabricant.

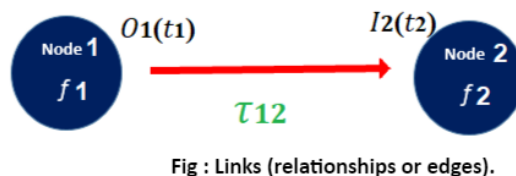


Figure 3.4 – Liens (relations ou arêtes)

Comprendre l’environnement économique et social comme un réseau de nœuds et de liens avec des durées spécifiques fournit un cadre pour analyser comment les différents composants interagissent au fil du temps. Cela permet d’examiner les retards, les goulots d’étranglement et l’efficacité globale du système. Cette approche en réseau met également en évidence les interdépendances et les processus dynamiques qui façonnent les résultats économiques et sociaux, soulignant l’importance du temps dans le flux des activités et des interactions au sein du système.

3.2.2 Caractéristiques fonctionnelles

Ce réseau proposé est basé sur des réseaux de décision dynamique (DDNs) et des modèles mathématiques pour simuler les interactions économiques et sociales. Ses caractéristiques comprennent :

Formation du réseau (Structure de réseau connectée)

Le réseau est un ensemble de nœuds interconnectés, représentant des unités d’interaction économique et sociale. Les relations entre les nœuds sont représentées graphiquement, montrant les influences mutuelles et les interactions.

Représentation des nœuds (Nœuds indépendants)

Un nœud correspond à un programme informatique indépendant qui calcule les informations d’entrée et génère les informations de sortie. Les nœuds sont implémentés sous forme de fonctions spécifiques avec certaines entrées et des sorties spécifiées.

Relations inter-nœuds (Influence mutuelle)

Les relations entre les nœuds entraînent des influences mutuelles et des interactions entre différentes unités économiques et sociales. Les unités peuvent s'influencer mutuellement directement en utilisant des connexions ou indirectement en influençant l'environnement général.

Échange de Données (Mécanisme d'Entrée/Sortie)

Un nœud envoie ses sorties sous forme d'entrées à un ou plusieurs autres nœuds. Ainsi, le système d'entrée/sortie fournit un échange de données et, par conséquent, des interactions au sein du réseau. La mise en œuvre de plusieurs nœuds de sortie entraîne une interdépendance complexe.

Échange d'Influence (Influence Directe et Indirecte)

L'interaction entre les nœuds peut se réaliser soit par un échange d'influence directe, en utilisant explicitement des connexions, soit par une influence indirecte par la modification de l'environnement dans lequel ils sont impliqués. Cela permet, à la fois, de modéliser l'échange d'influence directe et une influence environnementale plus large.

Représentation du Comportement Dynamique (Modélisation Mathématique)

Le comportement dynamique du réseau est décrit en se basant sur les DDNs, qui reflètent la complexité de l'interaction entre les nœuds. La modélisation prend en compte la variation dépendante du temps et la boucle de rétroaction à l'intérieur du réseau.

Analyse Informatique (Implémentation par Programme)

Le réseau est implémenté par la programmation informatique de manière à permettre une analyse et une expérimentation plus détaillées. Les utilisateurs doivent modéliser le réseau et pouvoir simuler à partir de différents scénarios pour révéler et comprendre l'interaction économique et sociale au sein du réseau.

Génération de Rapports (Rapport des Résultats de Simulation)

Des résultats détaillés de la simulation peuvent être déduits par les utilisateurs sous forme de tableau complet. Les rapports peuvent comparer différentes exécutions de simulation, facilitant l'analyse des effets obtenus en modifiant les entrées initiales.

Comparaison de Simulation (Comparaison des Résultats)

L'utilisateur comparera plusieurs exécutions de simulation différentes pour voir quels sont les effets lorsque les conditions initiales sur le comportement du réseau et ses sorties finales sont modifiées.

Gestion des Boucles (Valeurs Initiales pour les Boucles)

Les nœuds formant des boucles dans le réseau pourraient être exécutés normalement si des valeurs initiales sont fournies pour au moins l'un des nœuds formant une boucle, en veillant à ce que les dépendances cycliques soient exactement simulées.

Sorties Dépendantes du Temps (Sorties de Nœuds Différées)

Chaque nœud pourrait être configuré pour produire une sortie après que le temps de simulation se soit écoulé, par exemple, un an, un mois, ou X jours. Cela aura des résultats de simulation où les nœuds qui ne sont pas prêts à produire une sortie ne figureront pas dans le tableau de rapport jusqu'à ce que le temps indiqué se soit écoulé.

Applications du réseau proposé

Parmi les phénomènes qui peuvent être étudiés à l'aide du réseau :

- Multiples interactions et influences mutuelles entre les nœuds.
- Création de circuits de rétroaction positifs ou négatifs.
- Changements d'état et de comportement général du réseau en fonction des changements dans les facteurs d'entrée initiaux ou environnementaux.
- Étude de modèles complexes de comportement dynamique tels que les phénomènes économiques cycliques, les effondrements de marché et les transformations sociales.

3.2.3 Caractéristiques Techniques

Caractéristiques Techniques du Réseau Proposé Le réseau proposé dans l'application du système d'aide à la décision est conçu pour simuler les interactions économiques et sociales en utilisant des Réseaux de Décision Dynamique (DDNs) et des modèles mathématiques. Cette section explore les caractéristiques techniques du réseau, fournissant des explications détaillées et un pseudocode pour illustrer ses fonctionnalités.

1. Formation du Réseau Le réseau est formé par un ensemble de nœuds interconnectés, représentant chacun une unité d'interaction économique ou sociale. Les relations entre ces nœuds sont représentées visuellement, montrant les influences mutuelles et les interactions.

Représentation des Nœuds :

Chaque nœud fonctionne comme une unité de calcul indépendante avec des entrées et des sorties spécifiques. Les nœuds sont implémentés à l'aide de fonctions JavaScript qui encapsulent la logique de décision.

Gestion des Connexions : Les connexions entre les nœuds représentent le flux de données ou d'influence. Les connexions sont gérées à travers des flèches dans l'interface utilisateur, montrant comment les nœuds interagissent les uns avec les autres.

2. Gestion des Nœuds et des Connexions :

Les nœuds et les connexions sont les éléments de base du réseau, chaque nœud

représentant un point de décision et chaque connexion représentant la relation entre les nœuds.

Configuration des Nœuds :

Les nœuds peuvent être configurés avec des propriétés spécifiques telles que leur nom, leurs variables d'entrée et de sortie. Les propriétés des nœuds peuvent être éditées à travers l'interface utilisateur, permettant aux utilisateurs de définir la logique de calcul de chaque nœud.

Création Dynamique de Nœuds :

Les utilisateurs peuvent créer des nœuds dynamiquement à travers l'interface utilisateur. Les nœuds sont déplaçables et peuvent être positionnés n'importe où dans l'éditeur.

3. Exécution de la Logique de Décision

Chaque nœud dans le réseau peut être exécuté pour évaluer sa logique de décision basée sur les entrées qu'il reçoit. Les sorties d'un nœud peuvent être utilisées comme entrées pour d'autres nœuds, créant un flux de données à travers le réseau.

Flux d'Exécution :

Les nœuds sont exécutés dans une séquence déterminée par leurs dépendances. Les nœuds sans dépendances (racines) sont exécutés en premier, suivis de leurs nœuds dépendants.

Les résultats de l'exécution sont propagés à travers le réseau, chaque nœud traitant ses entrées et générant des sorties.

4. Représentation du Comportement Dynamique :

Des modèles mathématiques sont employés pour représenter le comportement dynamique du réseau, capturant la complexité des interactions entre les nœuds.

Modélisation Mathématique :

La logique de décision à l'intérieur de chaque nœud peut inclure des formules mathématiques et des algorithmes pour simuler les interactions économiques et sociales du monde réel.

Les modèles tiennent compte des changements variables dans le temps et des boucles de rétroaction à l'intérieur du réseau.

5. Simulation et Analyse :

Le réseau est implémenté en utilisant la programmation informatique, permettant une analyse détaillée et une expérimentation avec différents scénarios.

Exécution de la Simulation :

Les utilisateurs peuvent simuler le processus de prise de décision en exécutant le réseau. Les résultats de la simulation montrent comment les données circulent à travers le réseau et les résultats des points de décision. La simulation prend en compte les interactions complexes et les boucles de rétroaction définies dans les modèles mathématiques.

Génération de Rapports :

Après avoir exécuté une simulation, les utilisateurs peuvent générer des rapports dé-

taillés des résultats. Les rapports sont présentés sous forme de tableau complet, montrant les sorties de chaque nœud.

Les utilisateurs peuvent comparer plusieurs exécutions de simulation pour analyser les effets de différentes conditions initiales et configurations.

6. Persistance et Collaboration

L'application offre des fonctionnalités pour sauvegarder et charger des projets, permettant aux utilisateurs de préserver leur travail et de collaborer avec d'autres.

Sauvegarde et Chargement des Projets :

Les utilisateurs peuvent sauvegarder l'état actuel de leur projet, y compris tous les nœuds, les connexions et les configurations, dans un fichier.

Les projets peuvent être chargés à partir d'un fichier, restaurant l'état du réseau et permettant aux utilisateurs de poursuivre leur travail.

7. Gestion des Boucles et Sorties Différées :

Le réseau prend en charge les nœuds formant des boucles et les nœuds avec des sorties différées pour simuler des interactions plus complexes et des comportements dépendant du temps.

Valeurs Initiales pour les Boucles :

Les nœuds formant des boucles peuvent être exécutés en spécifiant des valeurs initiales pour au moins un nœud dans la boucle, assurant une simulation précise des dépendances cycliques.

Sorties Différées :

Les nœuds peuvent être configurés pour produire des résultats après un laps de temps spécifié (par exemple, 1 an, 1 mois, ou un certain nombre de jours). Cela affecte les résultats de la simulation, car les nœuds qui ne doivent pas produire de résultats n'apparaîtront pas dans le tableau de rapport jusqu'à ce que leur temps spécifié se soit écoulé.

3.2.4 Modélisation Mathématique du Réseau Proposé :

Pour modéliser mathématiquement le réseau économique et social, nous pouvons utiliser un ensemble de fonctions mathématiques et d'équations basées sur le temps pour représenter les nœuds et les liens.

Représentation des Nœuds

Chaque nœud N_i peut être représenté comme une fonction mathématique dépendant du temps et de ses entrées. Supposons que les entrées du nœud N_i soient $I_i(t)$ et ses sorties soient $O_i(t)$. Le nœud peut être représenté comme :

$$O_i(t_2) = f_i(I_i(t_1), D)$$

où :

- $O_i(t_2)$: Sortie du nœud N_i au temps t_2 .
- $I_i(t_1)$: Entrée du nœud N_i au temps t_1 .
- f_i : Une fonction mathématique représentant les opérations internes du nœud N_i et l'effet du temps sur ce nœud.
- D : La durée d'un nœud fait référence à l'intervalle de temps entre le moment où il reçoit ses entrées et celui où il génère ses sorties.

$$t_2 = t_1 + D$$

Représentation des Liens

Les liens entre les nœuds peuvent être représentés par des fonctions dépendantes du temps qui décrivent la durée nécessaire pour que les sorties d'un nœud deviennent les entrées d'un autre nœud. S'il y a un lien entre le nœud N_i et le nœud N_j , le lien L_{ij} peut être représenté comme suit :

$$L_{ij} : I_j(t_2) = O_i(t_1, \tau_{ij})$$

où :

- $I_j(t_2)$: Entrée du nœud N_j au temps t_2 .
- $O_i(t_1, \tau_{ij})$: Sortie du nœud N_i au temps t_1 .
- τ_{ij} : Le temps nécessaire pour transférer les sorties du nœud N_i au nœud N_j .

Réseau Global

En utilisant ces représentations, l'ensemble du réseau peut être décrit par une matrice de fonctions dépendantes du temps qui relie chaque nœud aux autres. Supposons que nous ayons un réseau simple avec trois nœuds N_1 , N_2 et N_3 , avec des liens entre chaque paire de nœuds. Le système peut être représenté comme suit :

$$O_1(t_2) = f_1(I_1(t_1), D_1)$$

$$O_2(t_4) = f_2(I_2(t_3), D_2)$$

$$O_3(t_6) = f_3(I_3(t_5), D_3)$$

Et les liens entre les nœuds (en cas de boucle fermée dans le système) :

$$L_{12} : I_2(t_3) = O_1(t_2 - \tau_{12})$$

$$L_{23} : I_3(t_5) = O_2(t_4 - \tau_{23})$$

$$L_{31} : I_1(t_7) = O_3(t_6 - \tau_{31})$$

Exemple

Supposons que le nœud N_1 représente une usine produisant des biens à partir de matières premières, le nœud N_2 représente un distributeur recevant des biens de l'usine, et le nœud N_3 représente un magasin vendant des biens aux consommateurs.

Les fonctions pourraient ressembler à ceci :

$$O_1(t) = f_1(I_1(t), t) = k_1 \cdot I_1(t)$$

où k_1 est une constante représentant l'efficacité de production.

$$O_2(t) = f_2(I_2(t), t) = k_2 \cdot I_2(t)$$

où k_2 est une constante représentant l'efficacité de distribution.

$$O_3(t) = f_3(I_3(t), t) = k_3 \cdot I_3(t)$$

où k_3 est une constante représentant l'efficacité de vente.

Les liens basés sur le temps pourraient être :

$$I_2(t) = O_1(t - 15)$$

indiquant que le transport de l'usine au distributeur prend 15 unités de temps.

$$I_3(t) = O_2(t - 20)$$

indiquant que le transport du distributeur au magasin prend 20 unités de temps.

Ainsi, nous pouvons modéliser le réseau économique et social comme un système dynamique en utilisant des fonctions mathématiques et temporelles pour représenter les interactions complexes entre différents nœuds.

Conclusion :

Le système d'aide à la décision proposé exploite les Réseaux de Décision Dynamique (DDNs) et les modèles mathématiques pour simuler des interactions économiques et sociales complexes. En fournissant des caractéristiques techniques détaillées et un pseudocode, nous illustrons comment le système facilite la création, la gestion et l'analyse de réseaux de décision. Cette approche complète garantit

Chapter 4

Conception du Système de Simulation basé sur le réseau de décision proposé

4.1 Introduction

La modélisation consiste à créer une représentation virtuelle d'une réalité de manière à mettre en évidence les points d'intérêt. Dans ce projet, nous utilisons le langage UML pour la modélisation.

4.1.1 Architecture générale du système proposé pour le réseau de décision proposé

A. Conception du réseau de décision

À cette étape, les nœuds et les relations entre eux doivent être configurés.

B. Configuration de la conception du réseau de décision

À cette étape, les entrées et les sorties des nœuds sont déterminées et modifiées, et les caractéristiques temporelles du réseau peuvent être configurées.

C. Conversion de la conception en code de programme

À cette étape, la conception du réseau est transformée en un programme, où l'ensemble des nœuds et les liens entre eux sont transformés d'une forme de conception en un programme qui comprend des fonctions et des appels dans un ordre spécifique.

D. Configuration du code de programme

Le contenu de code approprié est ajouté à chaque nœud de classe par l'utilisateur, selon son rôle computationnel prévu.

E. Exécution du code de programme

À cette étape, l'exécution est déterminée (avant ou arrière) et les fonctions des nœuds sélectionnés sont exécutées en respectant les configurations temporelles et la conception du réseau.

F. Extraction du rapport d'exécution

À cette étape, le programme générera un décompte des sorties résultant de certaines entrées.

Si les sorties du rapport sont appropriées pour la décision - les entrées au processus de prise de décision sont approuvées.

Si les sorties du rapport ne conviennent pas à la décision - retour à l'étape (4.1.1.B).

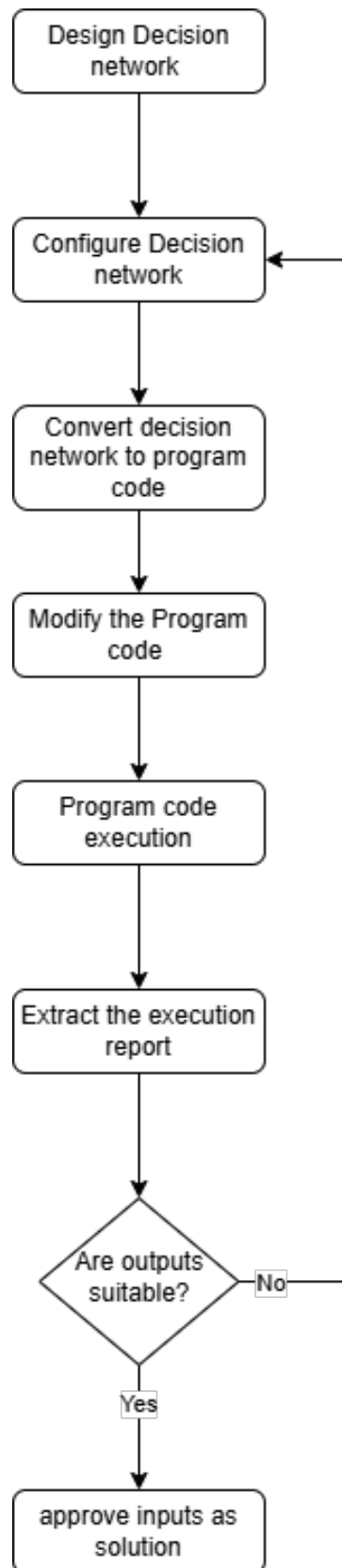


Figure 4.1 – Architecture générale du système proposé pour le réseau de décision proposé

4.1.2 Caractéristiques de Notre Réseau et Programme :

Notre système d'aide à la décision offre toute une gamme de fonctionnalités avancées qui le distinguent des travaux précédents. Ces fonctionnalités sont conçues pour améliorer la flexibilité, l'interactivité et l'efficacité des processus de prise de décision :

- Données du réseau basées sur SVG : Toutes les données du réseau sont enregistrées au format SVG, ce qui permet une manipulation et un déplacement faciles.
- Gestion des Images : Charger et sauvegarder le réseau sous forme d'image, facilitant le partage.
- Prise de Décision Graphique : Contrairement à la prise de décision basée sur Excel, notre application utilise une interface graphique pour une visualisation et une interaction améliorées.
- Importation de Croquis : Charger directement dans l'application tout croquis d'image avec des nœuds et des flèches.
- Modes d'Exécution Doubles : Exécuter les nœuds en mode avant ou arrière.
- Exécution par Zone : Exécuter uniquement les nœuds sélectionnés dans le réseau pour une analyse ciblée.
- Intégration de Code Personnalisé : Ajouter du code JavaScript personnalisé à l'intérieur des nœuds, offrant des possibilités infinies pour les fonctionnalités des nœuds.
- Intégration avec des Tiers : Connecter les nœuds à des services tiers, appeler des API et traiter des données à partir de fichiers .csv ou Excel.
- Communication avec le Serveur : Envoyer et recevoir des requêtes vers/depuis les serveurs, facilitant le partage des décisions et l'acquisition de données.
- Exécution Chronométrée : Les nœuds peuvent être configurés pour produire des résultats après une période de temps spécifiée (par exemple, 1 jour, 3 semaines).
- Fonctionnalité Copier-Coller : Copier et coller facilement des nœuds sélectionnés dans l'éditeur.
- Multiplateforme : Accessible sur plusieurs plates-formes en utilisant les technologies web.

4.1.3 Comparaison avec les Outils Existant

Pour mettre en avant les avancées de notre système d'aide à la décision, nous le comparerons à plusieurs outils existants couramment utilisés pour la prise de décision et la conception de réseaux. Ces outils incluent Arena Simulator, Draw.io et YEd Graph Editor. Bien que chacun de ces outils ait ses propres forces, ils ont également des limitations que notre système surmonte.

Arena Simulator

Arena Simulator est un logiciel de simulation d'événements discrets et d'automatisation développé par Rockwell Automation. Il est largement utilisé pour simuler des processus métier, des systèmes de fabrication et des opérations logistiques.

Limitations:

- Données du Réseau Basées sur SVG : Arena ne prend pas en charge l'enregistrement des données du réseau au format SVG, limitant la flexibilité pour la manipulation visuelle.
- Prise de Décision Graphique : Bien qu'Arena offre des capacités de simulation, il lui manque l'interface graphique intuitive pour la prise de décision que l'on trouve dans notre système.
- Intégration de Code Personnalisé : Arena utilise son propre langage propriétaire pour la logique du modèle, ce qui peut être restrictif par rapport à l'intégration de code JavaScript personnalisé dans notre système.
- Intégration de Tiers : Capacités limitées pour se connecter à des services et API tiers.
- Accessibilité Multiplateforme : Arena est principalement basé sur le bureau et n'offre pas le même niveau d'accessibilité multiplateforme.

Draw.io

Draw.io (diagrams.net) est un outil de diagramme en ligne qui permet aux utilisateurs de créer des organigrammes, des diagrammes de réseau et d'autres types de représentations visuelles.

Limitations:

- Logique d'Exécution : Draw.io est conçu pour les diagrammes statiques et ne prend pas en charge la logique d'exécution dynamique pour les réseaux de décision.
- Intégration de Code Personnalisé : Aucune capacité d'ajouter du code personnalisé à l'intérieur des nœuds pour la logique de traitement.
- Exécution Temporisée : Draw.io ne prend pas en charge l'exécution temporelle des nœuds ou des résultats différés.
- Intégration de Tiers : Capacités d'intégration limitées avec des services externes et des API.
- Prise de Décision Graphique : Bien que Draw.io offre de bons outils visuels, il lui manque les fonctionnalités interactives de prise de décision de notre système.

YEd Graph Editor

YEd est une puissante application de bureau pour créer rapidement des diagrammes de haute qualité. Il est utilisé pour créer des organigrammes, des diagrammes de réseau et des diagrammes UML.

Limitations:

- Données du Réseau Basées sur SVG : Ne prend pas en charge nativement l'enregistrement des données du réseau au format SVG pour une manipulation facile.
- Prise de Décision Graphique : Principalement axé sur la création de diagrammes statiques plutôt que sur la prise de décision interactive.
- Intégration de Code Personnalisé : Pas de support intégré pour ajouter du code JavaScript personnalisé aux nœuds.
- Intégration de Tiers : Manque d'intégration avancée avec des services tiers et des API.

- Exécution Temporisée : Pas de support pour configurer les nœuds pour produire des résultats après une période de temps spécifiée.

4.1.4 Explication de la structure de l'application et des étapes :

1-Conception du Réseau de Décision :

Dans cette section, nous explorerons la structure du système d'aide à la décision et les étapes impliquées dans la conception d'un Réseau de Décision. Cette conception sera basée sur les fichiers HTML et JavaScript fournis qui décrivent les composants de l'interface utilisateur et les interactions au sein d'un éditeur basé sur SVG.

Structure de l'Application Le Système d'Aide à la Décision est construit en utilisant des technologies web avec un accent sur SVG pour la représentation visuelle et JavaScript pour l'interactivité. Voici les principaux composants et leurs rôles :

Structure HTML :

Éditeur SVG : Le canevas principal où les nœuds et les connexions (flèches) sont affichés et manipulés.

Dialogues : Dialogues modaux pour éditer les propriétés des nœuds et afficher les résultats de l'exécution.

Barre d'Actions : Contient des boutons pour exécuter le réseau et sauvegarder l'état actuel.

Fonctions JavaScript :

Gestion des Nœuds : Fonctions pour créer, éditer et supprimer des nœuds dans le réseau de décision.

Gestion des Flèches : Fonctions pour créer, déplacer et supprimer des connexions entre les nœuds.

Gestion des Interactions : Écouteurs d'événements pour la fonctionnalité de glisser-déposer et les interactions de la souris.

Logique d'Exécution : Fonctions pour exécuter le réseau, traiter les entrées et générer les sorties.

Gestion de l'État : Fonctions pour sauvegarder et charger l'état de l'éditeur.

Afficher le Rapport : Fonctions pour générer et afficher un rapport de l'exécution du réseau de décision.

Étapes dans la Conception du Réseau de Décision :

Initialisation : L'application s'initialise en configurant l'éditeur SVG et en attachant les écouteurs d'événements nécessaires pour les interactions. La fonction `setUpEditor` configure l'éditeur avec des modèles déplaçables et des zones de corbeille.

Création et Configuration des Nœuds : Les nœuds sont créés en faisant glisser des modèles depuis le menu de formes sur le canevas. Chaque nœud représente un point de décision ou une opération dans le réseau. Les nœuds peuvent être édités via une boîte de dialogue modale, où les utilisateurs peuvent définir les noms des nœuds et définir la

logique de traitement à l'aide d'un éditeur de code intégré à Monaco Editor.

Création et Gestion des Flèches : Les connexions entre les nœuds sont représentées par des flèches. Elles sont créées en faisant glisser depuis les points d'ancrage sur les nœuds. Chaque flèche peut avoir des noms de variables qui lui sont associés, représentant les flux de données ou de décision entre les nœuds.

Exécution : Les nœuds peuvent être sélectionnés et exécutés pour simuler les résultats des décisions. La fonction `executeNode` traite de manière récursive les nœuds en fonction de leurs entrées et de la logique définie. Les résultats de l'exécution sont affichés dans une boîte de dialogue modale, montrant les sorties du réseau de décision.

Afficher le Rapport : Après l'exécution, les utilisateurs peuvent générer un rapport complet de l'exécution du réseau de décision. La fonction `generateReport` compile les résultats de chaque nœud dans un format lisible et la fonction `showReportDialog` affiche ce rapport dans une boîte de dialogue modale.

Gestion de l'État : L'état actuel de l'éditeur, y compris les nœuds et les connexions, peut être sauvegardé et chargé. Cela permet aux utilisateurs de persister leur travail et de le revoir ultérieurement. La fonction `saveEditor` gère le processus de sauvegarde, convertissant l'état de l'éditeur en un fichier SVG.

Validation et Gestion des Erreurs : Tout au long des étapes de conception et d'exécution, diverses vérifications de validation assurent que les nœuds et les flèches sont correctement configurés. Les erreurs dans la logique des nœuds sont détectées et affichées à l'utilisateur.

Exemple de Workflow

Interaction Utilisateur : Un utilisateur fait glisser un modèle de nœud sur le canevas, créant ainsi un nouveau nœud de décision. L'utilisateur ouvre la boîte de dialogue d'édition du nœud, définit un nom et détermine la logique de traitement à l'aide de l'éditeur de code. L'utilisateur crée des connexions entre les nœuds en faisant glisser depuis les points d'ancrage, définissant ainsi le flux de données ou de décisions.

Exécution : L'utilisateur sélectionne des nœuds et clique sur le bouton d'exécution pour lancer la simulation. Le système traite chaque nœud, exécute la logique définie et affiche les résultats.

Sauvegarde et Chargement : L'utilisateur sauvegarde l'état actuel du réseau, qui peut ensuite être chargé pour continuer à travailler à partir du même point. Cette approche de conception exploite SVG pour une représentation visuelle et flexible des réseaux de décision, tandis que JavaScript fournit l'interactivité et les capacités d'exécution nécessaires. La structure modulaire permet une expansion et une personnalisation faciles en fonction des besoins spécifiques.

2-Conversion des nœuds et des relations en fonctions :

Fonctions des Nœuds :

Chaque nœud dans le réseau de décision représente un point de décision ou un proces-

sus spécifique. Les nœuds sont configurés pour exécuter une logique personnalisée définie par l'utilisateur. Voici comment nous pouvons gérer les fonctions des nœuds :

Définition de la Logique des Nœuds : Les utilisateurs peuvent définir la logique de chaque nœud à l'aide d'un éditeur de code intégré dans l'application. Cette logique est écrite en JavaScript et peut inclure diverses opérations telles que des calculs, des conditionnels et des transformations de données.

Exécution de la Logique des Nœuds : Pendant l'exécution, la fonction de chaque nœud est évaluée, et les sorties sont générées en fonction des entrées fournies. La fonction `executeNode` gère cela en évaluant la logique du nœud et en renvoyant les sorties.

Fonctions des Flèches : Les flèches représentent les relations et les flux de données entre les nœuds. Chaque flèche peut transporter plusieurs variables, et les valeurs de ces variables sont transmises entre les nœuds connectés.

Définition des Variables des Flèches :

Les utilisateurs peuvent spécifier des noms de variables pour chaque flèche. Ces variables représentent les données transférées entre les nœuds. **Gestion des Connexions des Flèches :**

Les flèches sont créées en faisant glisser depuis les points d'ancrage sur les nœuds. Les points de départ et d'arrivée de chaque flèche sont liés à des nœuds spécifiques, établissant ainsi un flux de données.

Aperçu du Scénario :

Dans cet exemple de scénario, nous avons un réseau de décision qui calculera le prix final d'un produit en fonction de 2 variables : le prix et la quantité.

3-Scénarios d'Exécution:

Pendant l'exécution, l'utilisateur peut choisir entre 2 types d'exécution.

Avant: L'utilisateur sélectionne un nœud ou un groupe de nœuds et le programme les exécute, puis l'utilisateur modifie les configurations de ces nœuds et relance la simulation. Le programme générera alors un rapport des nœuds affectés par ces changements.

Arrière: L'utilisateur sélectionne un nœud ou un groupe de nœuds et calcule leurs sorties tout en exécutant les dépendances de ce nœud s'il y en a.

- **Nœud Prix de Base:** Ce nœud configure le prix de base du produit.
- **Nœud Quantité:** Ce nœud configure la quantité du produit dans la simulation.
- **Nœud Remise:** Ce nœud applique une remise au produit en fonction du prix de base et de la quantité.
- **Nœud Taxe:** Ce nœud calcule la taxe sur le produit en fonction du prix de base et de la quantité.
- **Nœud Prix Final:** Ce nœud calcule le prix final en fonction de toutes les sorties des nœuds précédents.

Ci-dessous se trouve l'architecture de ce réseau:

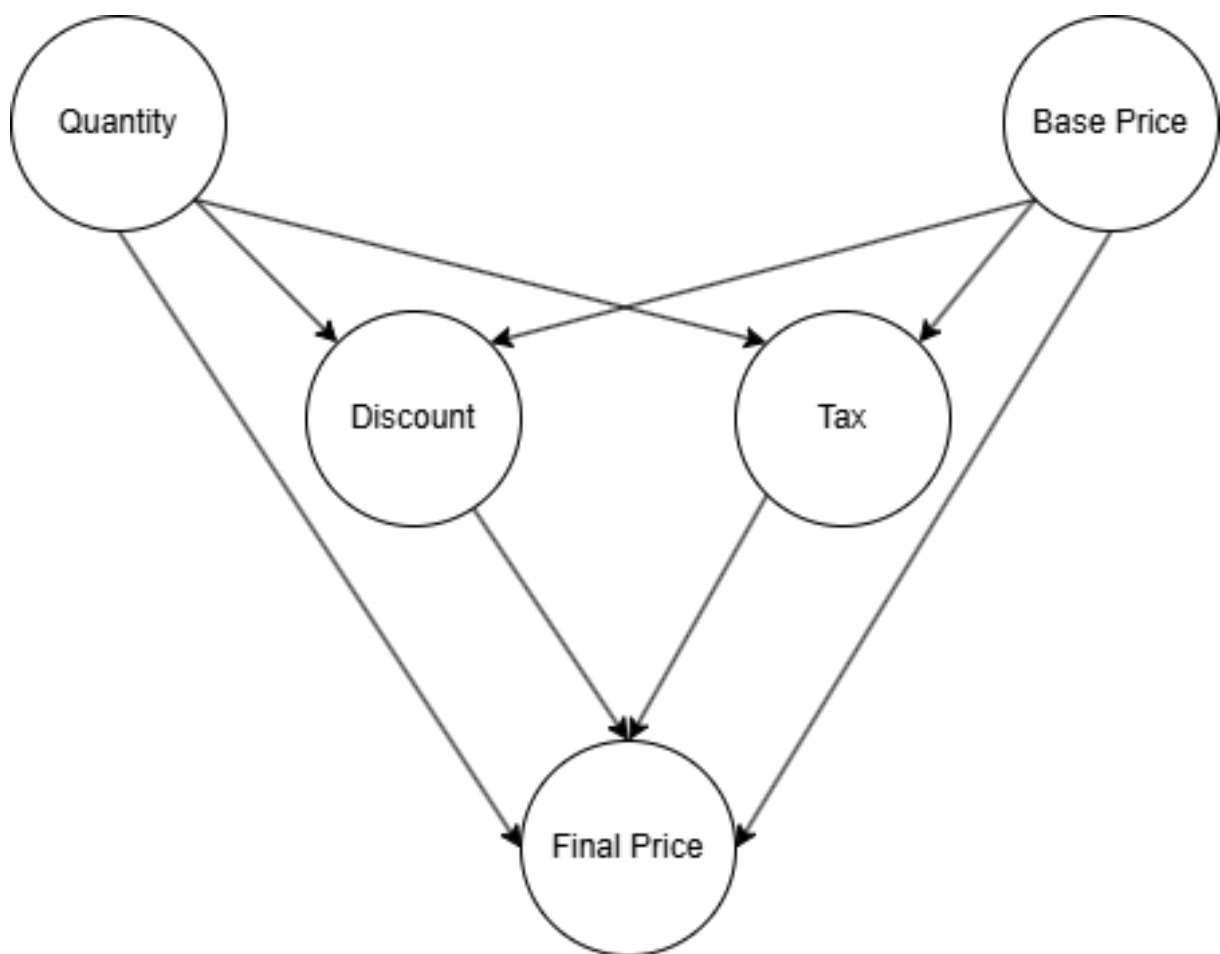


Figure 4.2 – Réseau de Décision de Tarification des Produits

Fonctions des Nœuds:

Nœud Prix de Base: Ce nœud configure le prix de base du produit.

```
function nodeFunction() {  
    return { basePrice: 100.0 }; // Le prix de base est fixé à 100,00 $  
}
```

Nœud Quantité: Configuration de la Quantité Ce nœud configure la quantité du produit dans la simulation.

```
function nodeFunction() {  
    return { quantity: 10 };  
}
```

Nœud Remise: Application de la Remise Ce nœud applique une remise au produit en fonction du prix de base et de la quantité.

```
function nodeFunction(basePrice, quantity) {  
    let discountAmount = 2; // Remise initiale de 2 $  
    if(basePrice < 25)  
        discountAmount = 0; // pas de remise si inférieur à 25 $  
    else if (quantity > 20)  
        // 1 gratuit pour un maximum de 200 $ de remise.  
        discountAmount = Math.min(basePrice, 200);  
    return { discountAmount };  
}
```

Chapter 4. Conception du Système de Simulation basé sur le réseau de décision proposé

Nœud Taxe: Application de la Taxe Ce nœud calcule la taxe sur le produit en fonction du prix de base et de la quantité.

```
function nodeFunction(discount, basePrice) {  
  let taxRate = 0.03; // Taux de taxe initial est de 3%  
  if (basePrice > 1000)  
    taxRate = 0.19  
  else if (quantity > 10)  
    taxRate = 0.08  
  return { taxRate };  
}
```

Nœud Prix Final: Calcul du Prix Final Ce nœud calcule le prix final en fonction de toutes les sorties des nœuds précédents.

```
function nodeFunction(basePrice, quantity, discount, taxRate) {  
  const total = basePrice * quantity;  
  const newTotal = total - discountAmount;  
  const totalTaxInc = newTotal + newTotal * taxRate;  
  return {  
    finalPrice: totalTaxInc  
  };  
}
```

Scénario d'Exécution Avant: Cette méthode d'exécution permet d'illustrer comment la sortie d'un nœud peut influencer d'autres nœuds pour dériver le prix final d'un produit.

Une fois l'exécution terminée, nous devrions obtenir un rapport des nœuds affectés. Nous pouvons également reconfigurer les nœuds sélectionnés et exécuter une deuxième simulation pour la comparer avec la précédente afin de prendre une bonne décision.

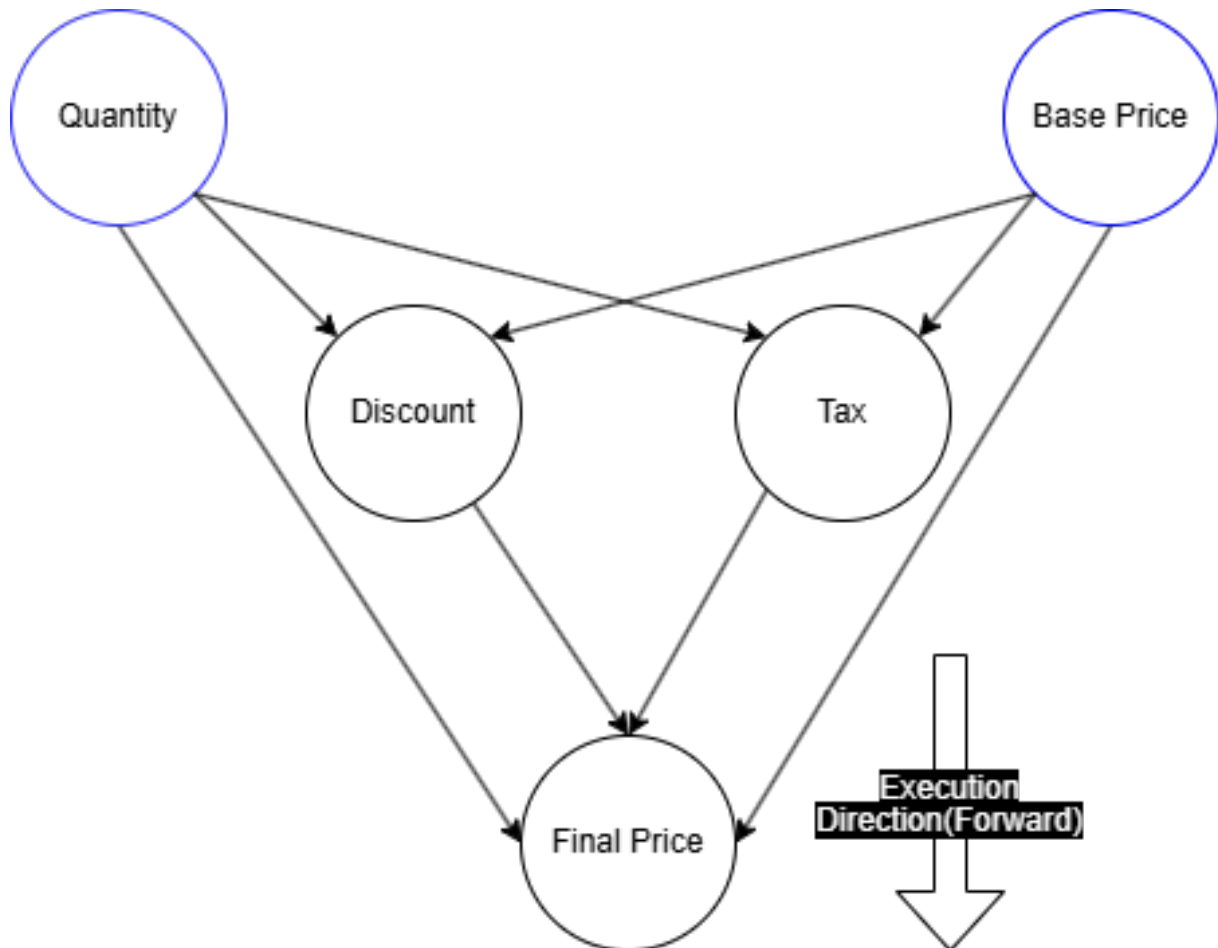


Figure 4.3 – Scénario d'Exécution Avant (Les Nœuds de Quantité et de Prix de Base sont sélectionnés pour l'exécution)

Scénario d'Exécution Arrière: Dans ce scénario, nous mettrons en œuvre un système d'exécution arrière où nous partons de la sortie finale souhaitée et remontons à travers les nœuds pour déterminer les entrées et les processus nécessaires. Cette approche aide à identifier les conditions nécessaires pour obtenir la sortie spécifiée. L'utilisateur devrait également pouvoir Fixer(verrouiller) les paramètres d'entrée d'un nœud afin que le système ignore les modifications apportées à ce nœud pendant l'évaluation.

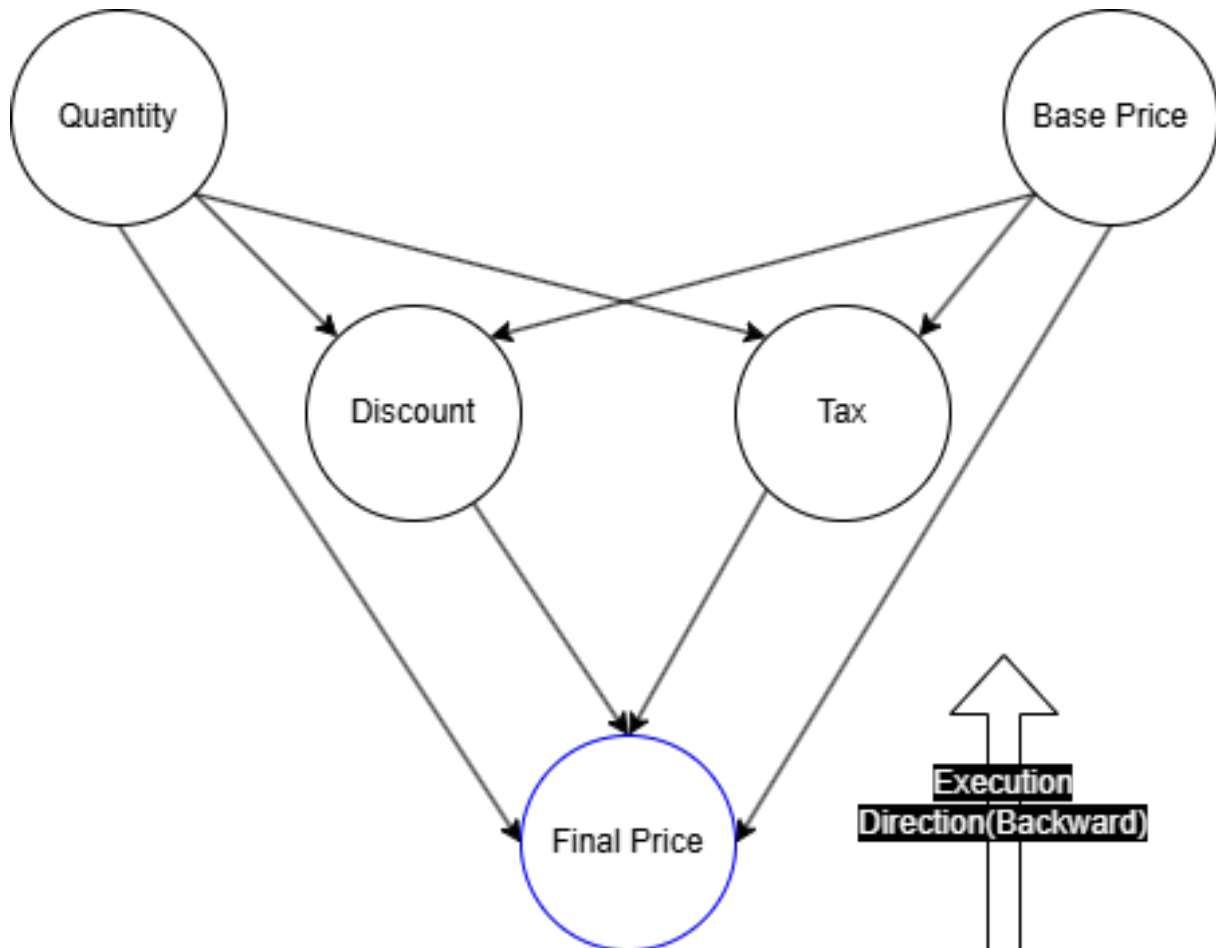


Figure 4.4 – Scénario d'Exécution Arrière (Le Nœud de Prix Final est exécuté)

Scénario d'Exécution de Boucle:

Exécution en Boucle dans un Système de Gestion des Stocks Dans ce scénario, nous mettrons en œuvre un système d'exécution en boucle où la sortie d'un cycle affecte les entrées du cycle suivant. Ce scénario est courant dans les systèmes de gestion des stocks où le niveau de stock à la fin d'une période influence les décisions pour la période suivante.

Aperçu du Scénario:

Nœud A (Prévision de la Demande): Prévoit la demande pour la prochaine période en fonction des données historiques.

Nœud B (Mise à Jour des Stocks): Met à jour les niveaux de stock en fonction de la demande et du stock actuel.

Nœud C (Décision de Commande): Décide de la quantité à commander pour la prochaine période en fonction des niveaux de stock mis à jour.

Exécution en Boucle: La sortie du Nœud C (quantité commandée) influencera la prévision de la demande et la mise à jour des stocks du prochain cycle.

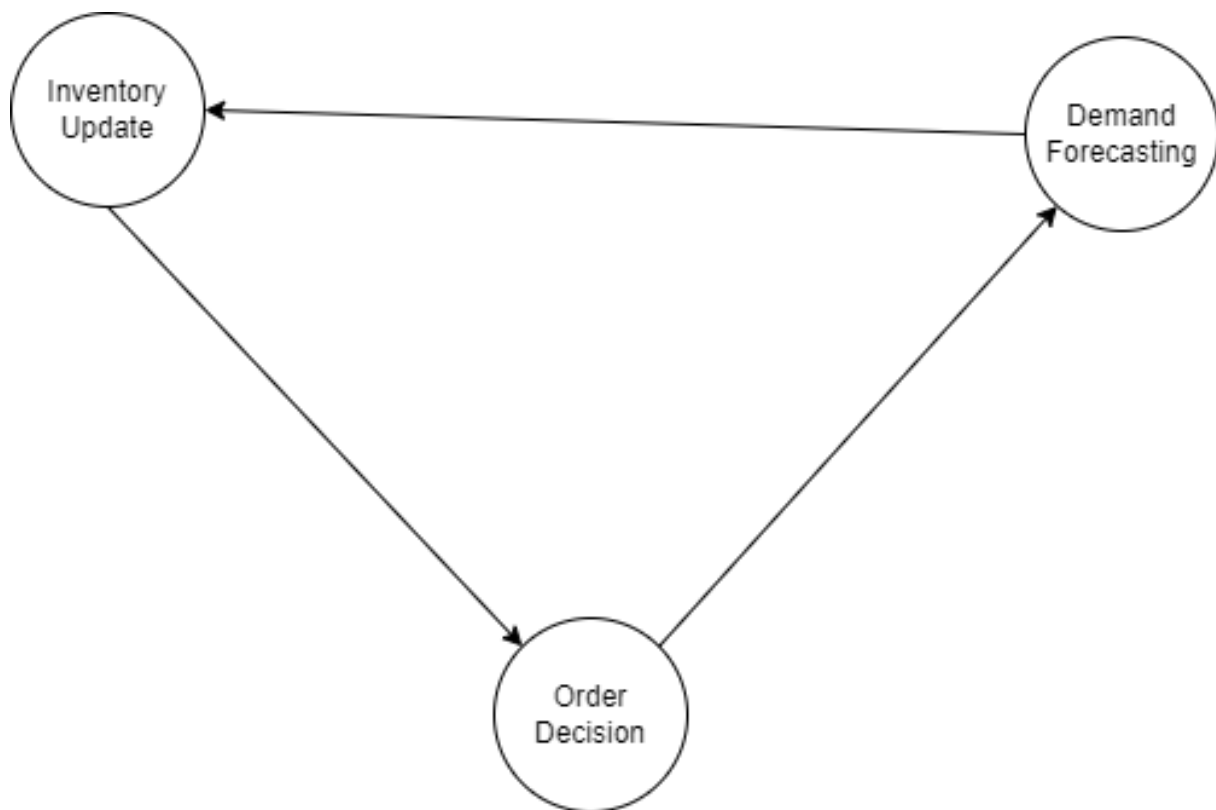


Figure 4.5 – Exécution en Boucle dans un Système de Gestion des Stocks

Boucle d'Exécution:

La fonction `executeLoop` simule le système sur plusieurs périodes. À chaque période:

Le Nœud A prévoit la demande en fonction des données historiques.

Le Nœud B met à jour les stocks en fonction de la demande prévue.

Le Nœud C décide de la quantité à commander nécessaire pour maintenir le niveau de stock désiré plus un stock de sécurité.

Les sorties d'une période servent d'entrées pour la période suivante, créant ainsi une boucle.

Scénario d'Exécution Temporelle :

Dans le système d'aide à la décision proposé, l'exécution temporelle du scénario est une fonctionnalité critique qui permet la simulation et l'analyse des réseaux de décision sur une période de temps spécifiée. Cette fonctionnalité est conçue pour capturer le comportement dynamique du réseau alors qu'il évolue, reflétant les dépendances temporelles et les retards inhérents aux systèmes du monde réel.

Caractéristiques Clés:

Sorties Dépendantes du Temps:

Les nœuds peuvent être configurés pour produire des sorties après un délai spécifique, tel qu'un certain nombre de jours, de mois ou d'années. Cela permet à la simulation de refléter des scénarios réels où les décisions ou actions prennent du temps à produire un effet. Par exemple, une décision politique économique pourrait mettre plusieurs mois à

avoir un impact sur les taux d'emploi, ou une campagne marketing pourrait influencer les ventes sur une période de semaines.

Valeurs Initiales pour les Boucles:

Dans les réseaux où les nœuds forment des boucles, il est essentiel de spécifier des valeurs initiales pour au moins un nœud de la boucle. Cela garantit que la simulation peut progresser sans tomber dans des cycles infinis et représente avec précision les dépendances cycliques dans le système. Les valeurs initiales fournissent un point de départ pour la simulation, permettant au modèle de gérer correctement les boucles de rétroaction.

Progression Temporelle de la Simulation:

La simulation s'exécute sur des pas de temps discrets, qui peuvent être des jours, des mois ou des années, selon la granularité requise. À chaque étape de temps, le réseau est évalué pour déterminer quels nœuds doivent produire des sorties en fonction de leurs retards configurés. Les nœuds qui ne doivent pas produire de sorties à l'étape de temps actuelle sont ignorés, ce qui permet à la simulation de gérer efficacement la progression du temps.

Gestion des Retards et des Dépendances:

Chaque nœud du réseau a un ensemble de dépendances et une configuration de retard. Le moteur de simulation garantit que les nœuds sont exécutés dans le bon ordre, en respectant leurs dépendances et leurs retards. Cela signifie qu'un nœud ne produira une sortie que si toutes ses dépendances ont été résolues et que sa période de retard s'est écoulée.

Rapport Basé sur le Temps:

Après l'exécution de la simulation, les résultats sont compilés dans un rapport complet qui reflète le comportement du réseau sur la période de simulation. Ce rapport comprend les sorties de chaque nœud à chaque étape de temps. Le rapport basé sur le temps permet aux utilisateurs d'analyser comment les décisions et les actions se propagent à travers le réseau au fil du temps, fournissant des informations sur la dynamique temporelle du système.

Scalabilité et Efficacité:

La conception du mécanisme d'exécution temporelle du scénario garantit que le système peut gérer de manière efficace des réseaux larges et complexes. En évaluant les nœuds uniquement lorsqu'ils doivent produire des sorties, le système minimise les calculs inutiles et optimise les performances.

Étapes de Mise en Œuvre:

Configuration des Nœuds: Les utilisateurs configurent chaque nœud avec sa logique de décision, ses variables d'entrée, ses variables de sortie et le délai avant qu'il ne produise une sortie. L'interface utilisateur permet aux utilisateurs de définir facilement ces paramètres, garantissant que le réseau est représenté de manière précise.

Configuration de la Simulation:

La configuration de la simulation implique de définir la période totale de simulation et les intervalles de pas de temps. Les utilisateurs peuvent spécifier les temps de début et de fin de la simulation. Les valeurs initiales pour les nœuds dans les boucles sont définies pour garantir que la simulation peut démarrer correctement.

Exécution de la Simulation:

Le moteur de simulation itère sur chaque pas de temps, évaluant les nœuds en fonction de leurs dépendances et de leurs configurations de retard. Les sorties sont enregistrées à chaque pas de temps, et les résultats sont stockés pour la génération de rapports.

Génération de Rapports:

Après la fin de la simulation, les résultats sont traités pour générer des rapports détaillés. Ces rapports montrent l'état du réseau à chaque pas de temps, mettant en évidence les sorties des nœuds et la propagation des décisions.

Les utilisateurs peuvent utiliser ces rapports pour analyser l'impact de différents scénarios et prendre des décisions éclairées en fonction des résultats de la simulation.

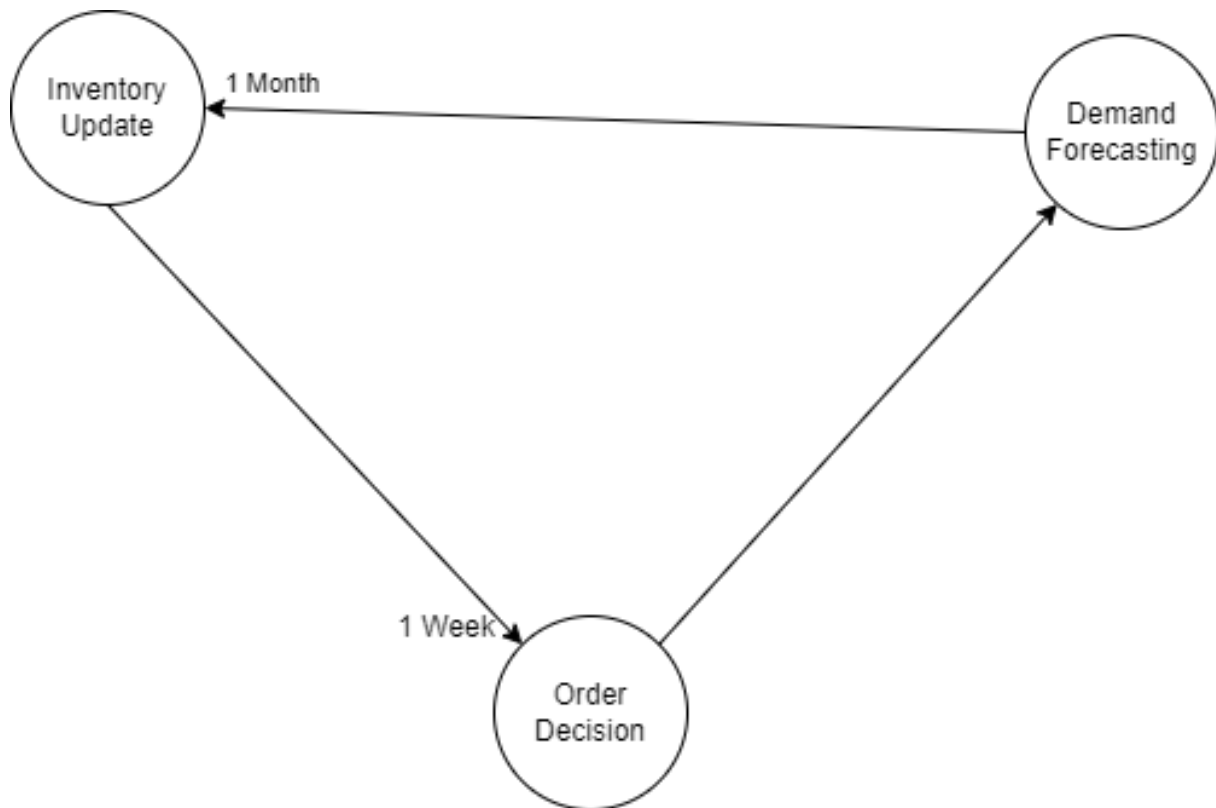


Figure 4.6 – Exécution de Boucle avec Facteur Temps dans un Système de Gestion des Stocks

Exemple:

La réception de la commande prend un mois, puis la décision de commande attend une mise à jour des stocks pendant une semaine

L'exécution temporelle du scénario dans le système d'aide à la décision est une fonctionnalité puissante qui permet aux utilisateurs de simuler et d'analyser la dynamique temporelle des réseaux de décision. En incorporant des sorties dépendantes du temps, en traitant les boucles avec des valeurs initiales, et en générant des rapports basés sur le temps, le système fournit un outil complet pour comprendre et optimiser les interactions économiques et sociales complexes.

4.1.5 Le Langage UML

UML est l'acronyme de "Unified Modeling Language". La notation UML est un langage visuel composé d'un ensemble de diagrammes, appelés diagrammes (14 au total), chacun donnant une vue différente du projet à traiter. Dans ce travail, nous présenterons seulement 3 diagrammes. Ces diagrammes sont:

- Le diagramme de cas d'utilisation.
- Le diagramme d'activité.
- Le diagramme de classe.
- Le diagramme de séquence.

4.1.6 Avantages et Inconvénients de UML

A. Avantages

- UML est un langage formel et standardisé.
- Offre une Précision Rigoureuse.
- Facile à utiliser.
- Garantit un puissant outil de communication.
- Facile à comprendre la personification abstraite composite et complexe.
- Fournit un moyen d'analyser les besoins.
- UML est un langage universel.

B. Inconvénients

- Mettre UML en pratique nécessite un apprentissage et passe par une période d'adaptation.
- Le processus est une autre clé du succès d'un projet, le processus est une tâche complexe et chronophage [12].

4.1.7 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont utilisés pour donner un aperçu du comportement fonctionnel d'un système logiciel. Ils sont utiles pour les présentations à la direction ou aux parties prenantes du projet. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. C'est une unité de travail significative. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

Diagramme de cas d'utilisation : Général

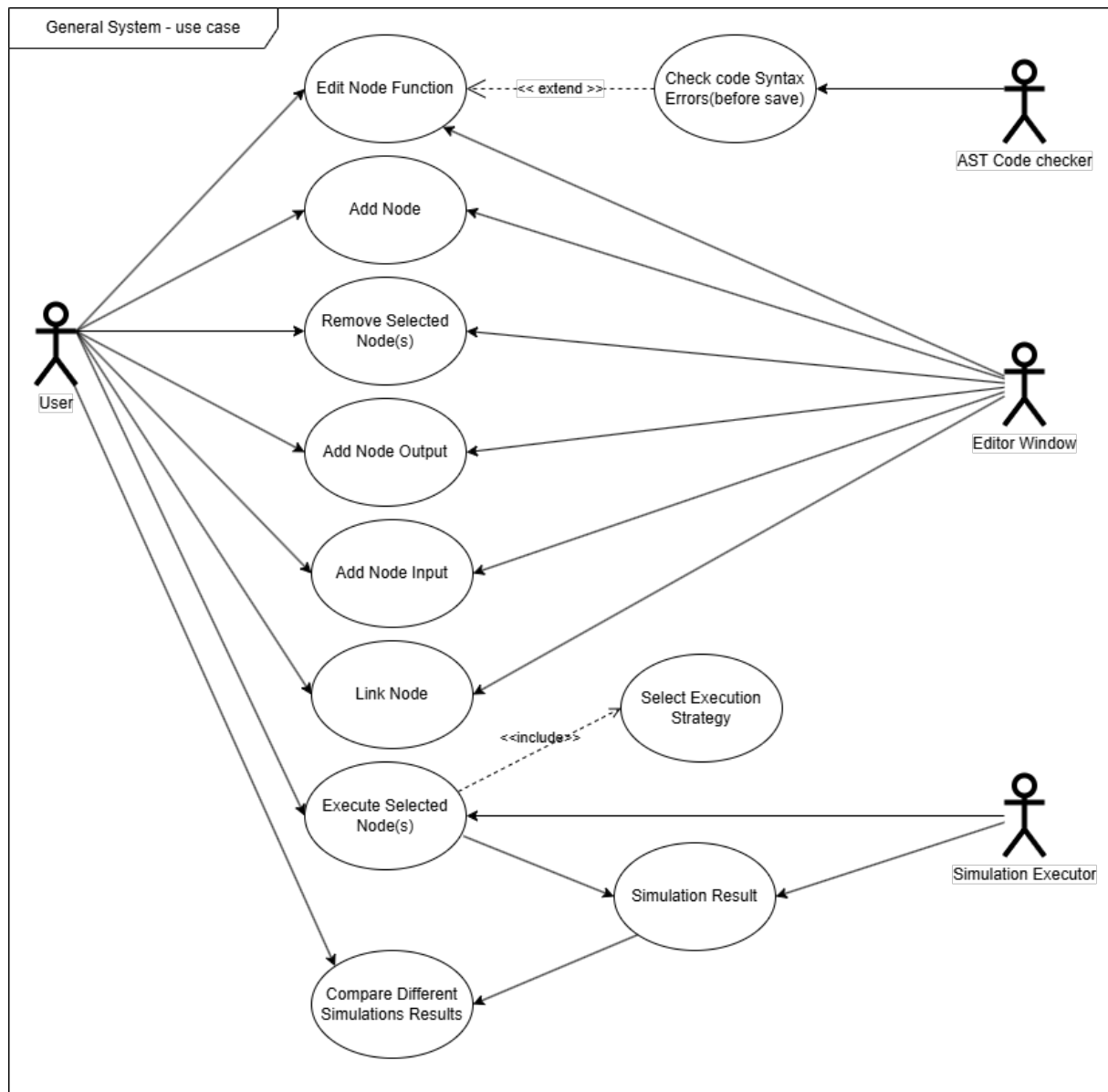


Figure 4.7 – Diagramme de cas d'utilisation général

Scénario pour le diagramme de cas d'utilisation

Acteurs :

- **Utilisateur** : L'acteur principal qui interagit avec l'éditeur pour créer, éditer et exécuter des nœuds dans la simulation.
- **Fenêtre de l'éditeur** : Un acteur secondaire (composant du système) qui fournit l'interface permettant à l'utilisateur de faire glisser et déposer des nœuds, d'éditer des propriétés et de visualiser des connexions.
- **Vérificateur de code AST** : Un autre acteur secondaire (composant du système) qui valide le code pour les erreurs de syntaxe et de logique pendant le processus d'édition.
- **Exécuteur de simulation** : Un acteur secondaire (composant du système) responsable du traitement des nœuds, de l'exécution de leurs fonctions et de la gestion des dépendances de données pendant la simulation.

Cas d'utilisation :

- **Créer un nœud**
 - **Acteur** : Utilisateur
 - **Description** : L'utilisateur fait glisser un modèle de nœud vers la fenêtre de l'éditeur pour créer un nouveau nœud.
- **Modifier un nœud**
 - **Acteur** : Utilisateur
 - **Description** : L'utilisateur sélectionne et modifie un nœud dans la fenêtre de l'éditeur.
 - **Composant du système impliqué** : Vérificateur de code AST
- **Connecter des nœuds**
 - **Acteur** : Utilisateur
 - **Description** : L'utilisateur connecte les nœuds en faisant glisser des flèches entre eux dans la fenêtre de l'éditeur.
- **Exécuter la simulation**
 - **Acteur** : Utilisateur
 - **Description** : L'utilisateur lance la simulation en cliquant sur le bouton Exécuter.
 - **Composant du système impliqué** : Exécuteur de simulation
- **Vérification du code AST («étendre»)**
 - **Acteur** : Vérificateur de code AST
 - **Description** : Vérifie automatiquement la syntaxe et la logique du code lorsqu'un nœud est modifié.

Déroulement des événements :

- **Création d'un nœud**
 - **Utilisateur** : Fait glisser et dépose un modèle de nœud.
 - **Fenêtre de l'éditeur** : Crée et affiche le nouveau nœud.
- **Modification d'un nœud**
 - **Utilisateur** : Double-clique sur un nœud pour le modifier.
 - **Vérificateur de code AST** : Valide le nouveau code et fournit des retours.
- **Connexion des nœuds**
 - **Utilisateur** : Connecte les nœuds avec des flèches.

- **Fenêtre de l'éditeur** : Gère la représentation visuelle des connexions.
- **Exécution de la simulation**
 - **Utilisateur** : Clique sur Exécuter.
 - **Exécuteur de simulation** : Traite les nœuds, exécute leurs fonctions et affiche les résultats.
- **Validation du code du nœud**
 - **Vérificateur de code AST** : Vérifie le code lors de la modification du nœud, s'assure qu'il est valide avant d'enregistrer les modifications.

Rôle des acteurs secondaires : Explication supplémentaire :

- **Fenêtre de l'éditeur** : Fournit une interface interactive pour l'utilisateur. Gère les actions de glisser-déposer, le positionnement des nœuds et la gestion des connexions.
- **Vérificateur de code AST** : Valide la syntaxe et la logique du code des nœuds en étendant le cas d'utilisation Modifier un nœud en garantissant la correction du code avant que la fonction du nœud ne soit enregistrée.
- **Exécuteur de simulation** : Exécute la simulation en fonction des nœuds définis et de leurs connexions. Il traite également les dépendances de données et génère des résultats.

4.1.8 Diagrammes d'activité

Le cas d'utilisation peut avoir plusieurs scénarios alternatifs. Il est alors difficile d'avoir une vision de toutes les actions, le diagramme d'activité est un moyen graphique de donner cette vue d'ensemble. Ce diagramme peut aider à voir un peu plus clairement les fonctionnalités et les activités pouvant être réalisées par l'application. Voici quelques-uns de nos diagrammes d'activité utilisés dans notre projet :

Diagramme d'activité : Cycle de vie d'un nœud

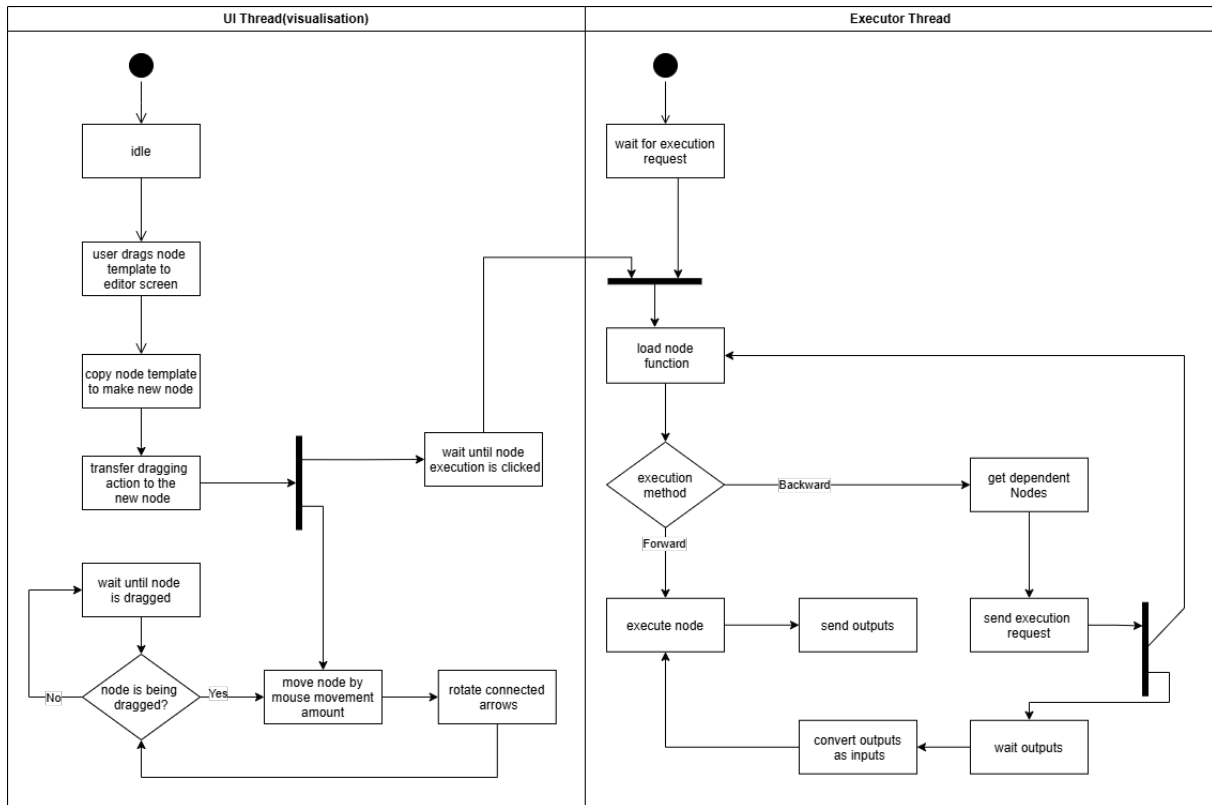


Figure 4.8 – Diagramme d'activité : Cycle de vie d'un nœud.

Thread Principal

- **État d'attente** Le système reste dans un état d'attente jusqu'à ce que l'utilisateur initie une action.
- **L'utilisateur fait glisser un modèle de nœud sur l'écran de l'éditeur** Lorsque l'utilisateur fait glisser un modèle de nœud du menu de formes vers l'écran de l'éditeur, cette action est capturée et gérée par le système.
- **Copier le modèle de nœud pour créer un nouveau nœud** Le système copie le modèle de nœud pour créer une nouvelle instance de nœud sur l'écran de l'éditeur.
- **Transférer l'action de glissement au nouveau nœud** L'action de glissement initiée par l'utilisateur est transférée au nouveau nœud créé.
- **Attendre jusqu'à ce que le nœud soit glissé** Le système attend que l'utilisateur termine l'action de glissement.
- **Le nœud est-il en train d'être glissé ? (Décision)** Ce point de décision vérifie si le nœud est toujours en train d'être glissé.
 - **Oui** : Le nœud est déplacé en fonction du déplacement de la souris, et les flèches connectées sont tournées en conséquence.
 - **Non** : Le système attend que le nœud soit à nouveau glissé.

Thread d'exécution :

- **Attendre la demande d'exécution** Le thread d'exécution reste dans un état d'attente jusqu'à ce qu'il reçoive une demande d'exécution du thread principal.
- **Charger la fonction du nœud** Après avoir reçu une demande d'exécution, le système charge la fonction associée au nœud.
- **Méthode d'exécution (Décision)** Le système détermine la méthode d'exécution :
 - **Avant** : La fonction du nœud est exécutée directement.
 - **Arrière** : Le système obtient d'abord les nœuds dépendants avant d'exécuter la fonction du nœud.
- **Exécuter le nœud** La fonction du nœud est exécutée, produisant des sorties.
- **Envoyer les sorties** Les sorties générées par la fonction du nœud sont envoyées à l'étape suivante du processus.
- **Obtenir les nœuds dépendants** Pour une exécution arrière, le système identifie et récupère les nœuds dépendants.
- **Envoyer une demande d'exécution** Une demande d'exécution est envoyée aux nœuds dépendants pour garantir qu'ils sont traités avant le nœud actuel.
- **Convertir les sorties en entrées** Les sorties des nœuds dépendants sont converties et utilisées comme entrées pour le nœud actuel.
- **Attendre les sorties** Le système attend que toutes les sorties des nœuds dépendants soient reçues avant de continuer avec l'exécution du nœud actuel.

4.1.9 Diagramme de séquence

Un diagramme de séquence est un type de diagramme d'interaction qui illustre comment les processus interagissent les uns avec les autres et dans quel ordre. Il montre les objets participant à l'interaction et la séquence des messages échangés entre eux pour effectuer la fonctionnalité du scénario.

Exécution d'un nœud

Ce prochain diagramme de séquence illustre le processus d'exécution d'un nœud, détaillant les étapes de l'action de l'utilisateur à l'exécution du nœud et à la gestion des entrées, des dépendances et des sorties.

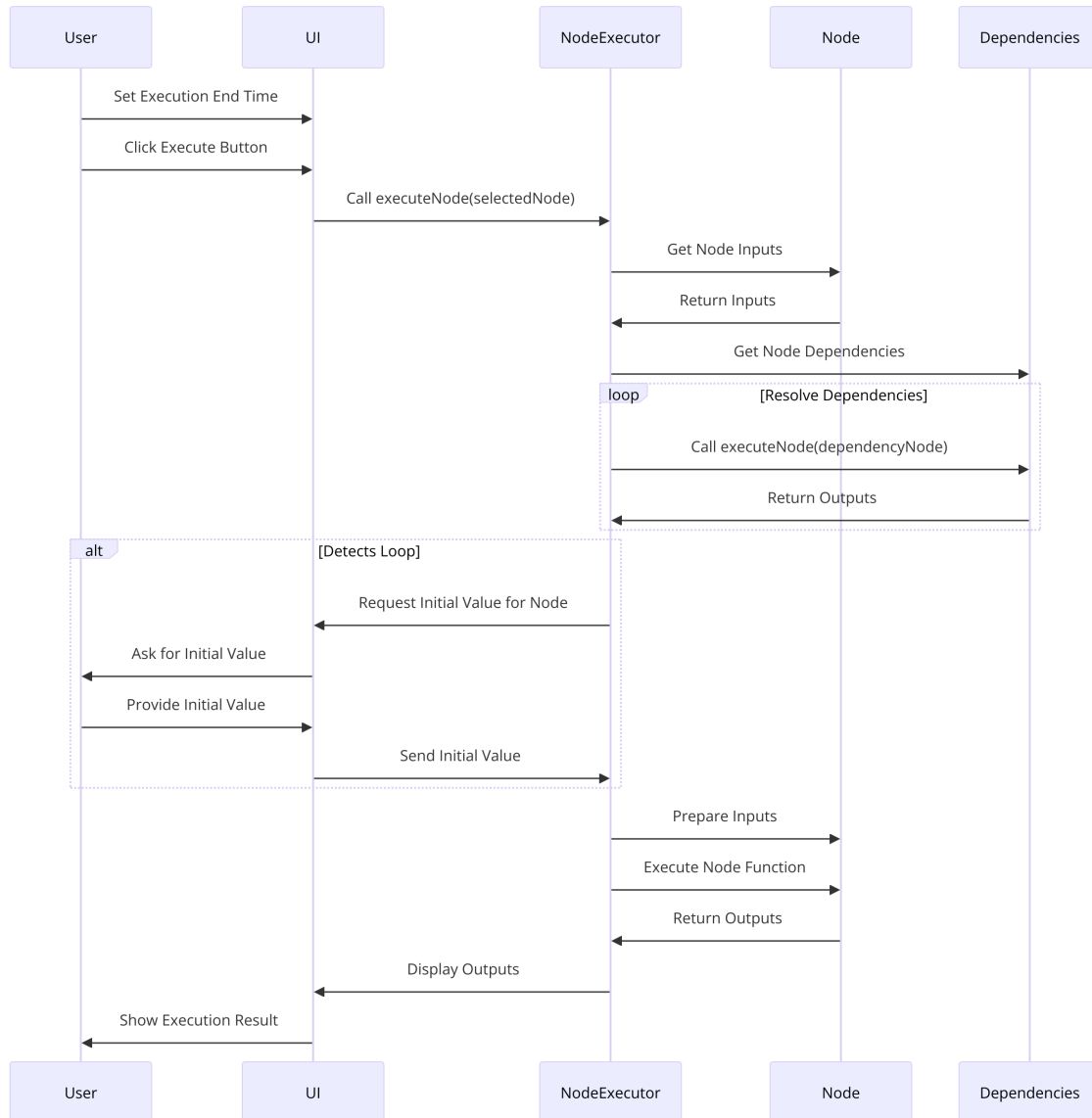


Figure 4.9 – Diagramme de séquence d'exécution de nœud

La séquence détaillée pour l'exécution d'un nœud est la suivante :

- **Démarrer l'exécution** : L'utilisateur clique sur le bouton Exécuter. Ensuite, la fonction `executeNode` est appelée avec le nœud sélectionné.
- **Obtenir les entrées du nœud** : Appel de `getNodeInputs` pour récupérer les variables d'entrée des flèches entrantes.
- **Résoudre les dépendances** : Appel de `getNodeDependencies` pour obtenir les dépendances (autres nœuds). Parcourir les dépendances et appeler récursivement `executeNode` pour chaque nœud dépendant afin d'obtenir leurs valeurs de sortie.
- **Préparer les entrées** : Collecte des valeurs d'entrée requises pour l'exécution du nœud actuel.
- **Exécuter la fonction du nœud** : Récupération du code de fonction du nœud en utilisant `getNodeFunction`. Ensuite, exécution de la fonction avec les entrées rassemblées.
- **Gérer les sorties** : Capture des valeurs de sortie. Journalisation ou stockage de la sortie pour un traitement ultérieur.
- **Fin de l'exécution** : Renvoi des valeurs de sortie.

4.1.10 Diagramme de classes

Définition

Le diagramme de classes permet de représenter les relations entre les éléments de modélisation statique composant les différentes entités du domaine d'application ainsi que les associations entre eux. L'établissement du diagramme de classes est une étape essentielle dans le processus de conception de la base de données.

À partir de l'étude du système existant et des différents diagrammes de cas d'utilisation, nous avons pu identifier les principales classes illustrées dans la figure ci-dessous pour avoir une vue plus claire du système étudié. Ce diagramme de classes contient des attributs et des méthodes. À la page suivante, nous montrons le diagramme de classes pour notre système.

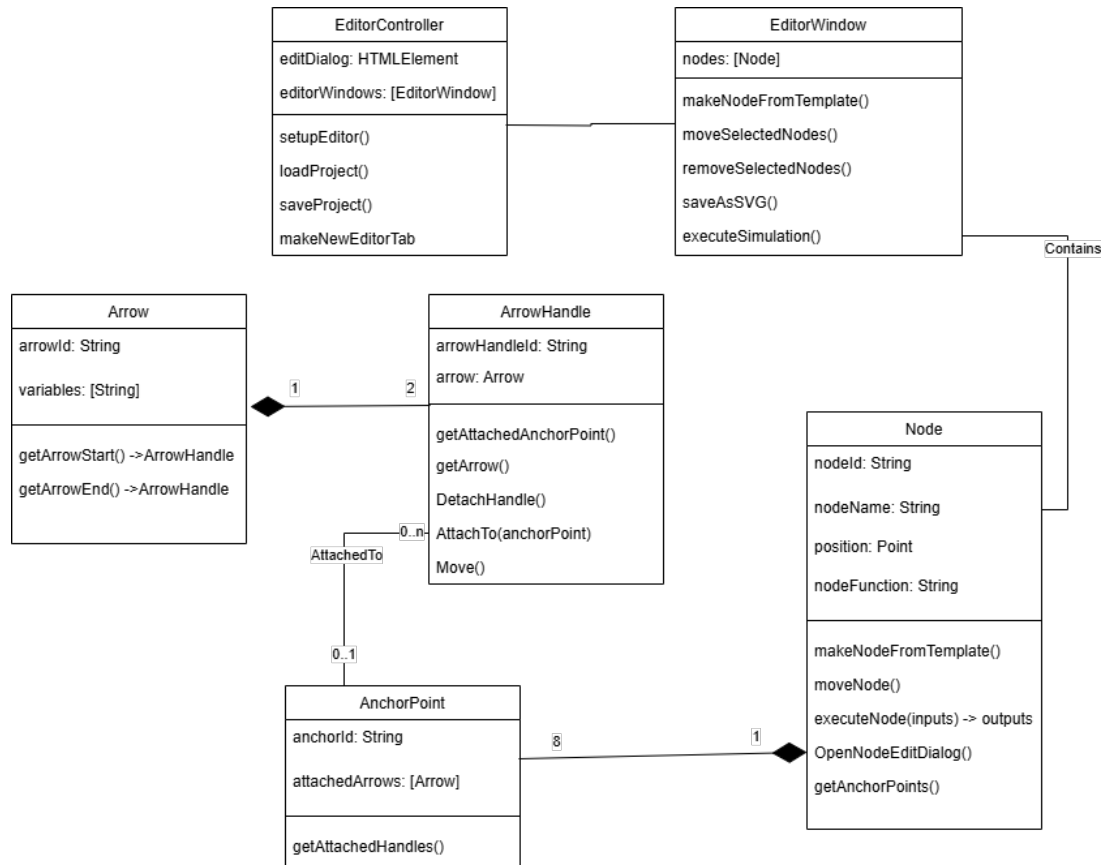


Figure 4.10 – Diagramme de classes.

4.2 Conclusion

Dans ce chapitre, nous avons présenté la modélisation de notre système. Quatre types de diagrammes ont été utilisés : diagramme de cas d'utilisation, diagramme d'activité, diagramme de séquence et diagramme de classes.

Le diagramme de cas d'utilisation et le diagramme d'activité nous ont permis d'établir les fonctionnalités du système, et le diagramme de classes nous a permis de poser les bases de l'implémentation du système en code.

Cette base théorique nous guidera dans la prochaine étape de développement, qui sera décrite dans le prochain chapitre.

Chapter 5

Réalisation du réseau de décision proposé

5.1 Introduction

Les réseaux de décision sont particulièrement précieux dans des domaines comme l'économie, la stratégie d'entreprise et les sciences sociales, où la prise de décision implique souvent de multiples variables et résultats potentiels.

Les réseaux de décision dynamiques (DDN) étendent les réseaux de décision traditionnels en incorporant des aspects temporels, permettant la modélisation de la prise de décision séquentielle dans le temps. Cela rend les DDN particulièrement adaptés pour simuler les interactions économiques et sociales, où les décisions à un moment donné peuvent influencer les choix et les résultats futurs. En permettant l'analyse de scénarios dynamiques et évolutifs, les DDN offrent un outil puissant pour comprendre et optimiser les processus de prise de décision dans des systèmes complexes.

Dans ce chapitre, nous réaliserons le réseau de décision proposé selon les sections suivantes :

L'environnement de développement

L'environnement logiciel

Technologies utilisées

Interfaces graphiques

Conclusion

5.2 L'environnement de développement

5.2.1 L'environnement Machine

- DELL Latitude E5440
 - Type de système d'exploitation : Windows 10 Pro 64 bits
 - Processeur : Intel(R) Core (TM) i5-4300U CPU @ 1,90 GHz, 2501 Mhz, 2 cœurs, 4 processeurs logiques
 - RAM : 8 Go.
 - Disque dur : 500 Go HDD.
- HP
 - Type de système d'exploitation : Windows 10 Pro 64 bits
 - Processeur : Intel(R) Core (TM) i5-4210U CPU @ 2,9 GHz, 3100 Mhz, 2 cœurs, 4 processeurs logiques
 - RAM : 12 Go.
 - Disque dur : 240 Go SSD, 500 Go HDD.

5.2.2 L'environnement logiciel

Draw.io

Pour créer les diagrammes UML, nous avons utilisé une plateforme en ligne appelée draw.io.

Draw.io ou Diagrams.net est une plateforme en ligne open source sortie en 2013, où les gens peuvent créer des diagrammes. Ce site web prend en charge la collaboration en temps réel lorsqu'il est connecté à un compte Google. Il n'y a pas d'option pour un compte payant, mais vous pouvez payer pour des serveurs. draw.io est entièrement en ligne et construit autour de Google Drive, ce qui vous permet de créer des organigrammes, des UML, des relations entités, des diagrammes de réseau, des maquettes et plus encore [5].

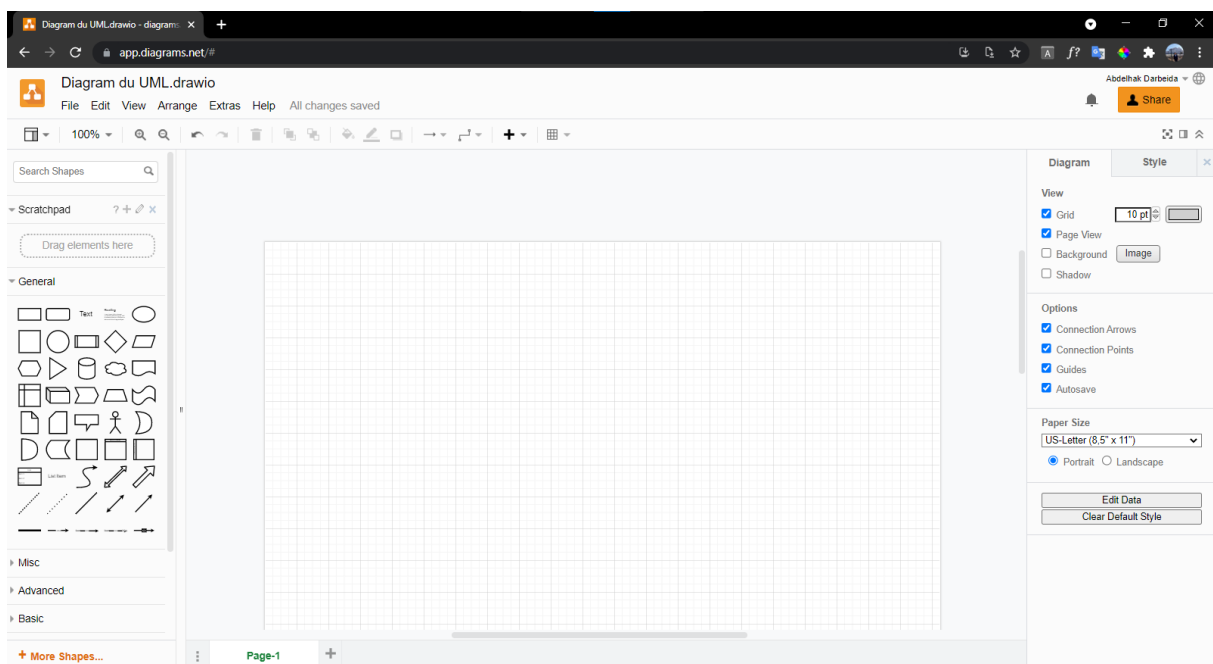


Figure 5.1 – Interface de la plateforme Draw.io.

Visual Studio Code

Visual Studio Code est un éditeur de code multiplateforme publié par Microsoft. Cet outil destiné aux développeurs prend en charge plusieurs dizaines de langages de programmation tels que HTML, C++, PHP, Javascript, CSS, etc. Visual Studio Code intègre plusieurs outils qui facilitent la saisie de code pour les développeurs, tels que la coloration syntaxique et le système d'auto-complétion IntelliSense. De plus, l'outil permet aux développeurs de corriger leur code et de gérer les différentes versions de leurs fichiers de travail, puisqu'un module de débogage est également inclus. [11].

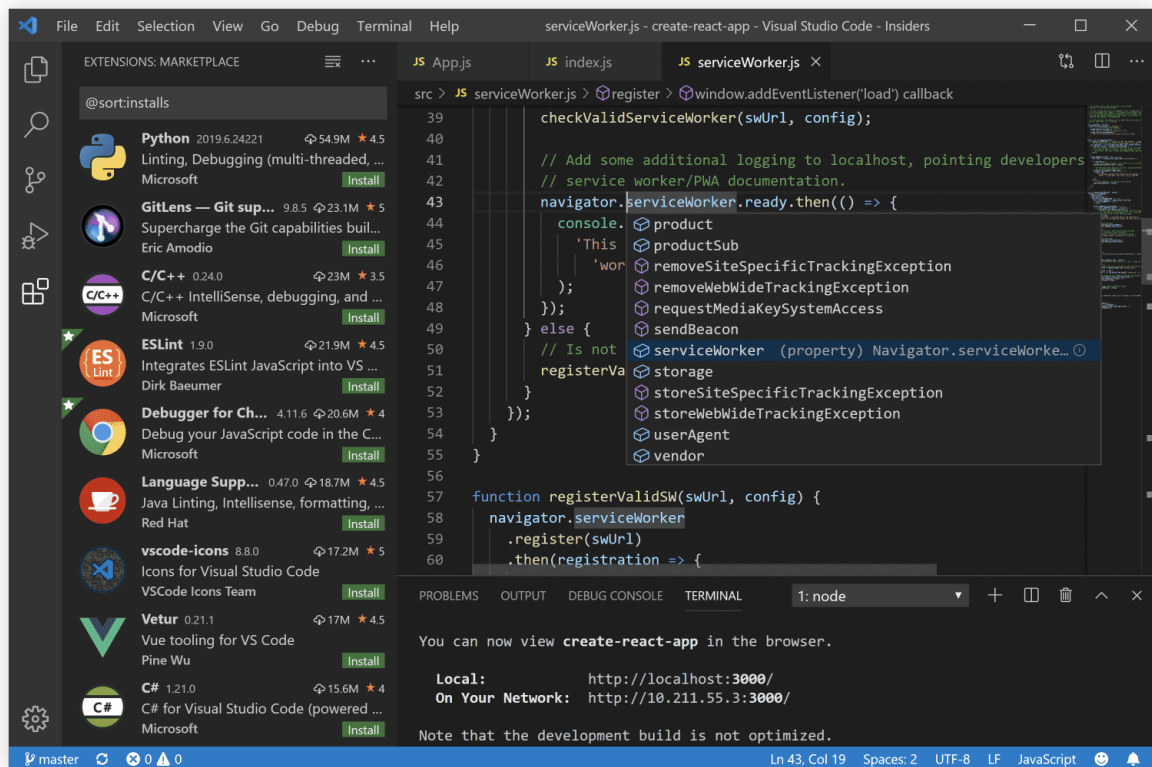


Figure 5.2 – Interface de Visual Studio Code.

IntelliJ IDEA Ultimate

IntelliJ IDEA Ultimate est un IDE polyvalent et puissant conçu pour le développement Java, bien qu'il prenne en charge une large gamme de langages et de frameworks. Il offre une complétion de code avancée, des outils de refactoring robustes, un excellent débogueur et une intégration transparente avec les systèmes de contrôle de version. De plus, il comprend des outils de base de données intégrés, un support de test et un vaste écosystème de plugins pour améliorer la productivité. En se familiarisant avec ses fonctionnalités clés, ses raccourcis et ses options de personnalisation, il peut considérablement améliorer l'efficacité de la programmation et la gestion de projet.[7]

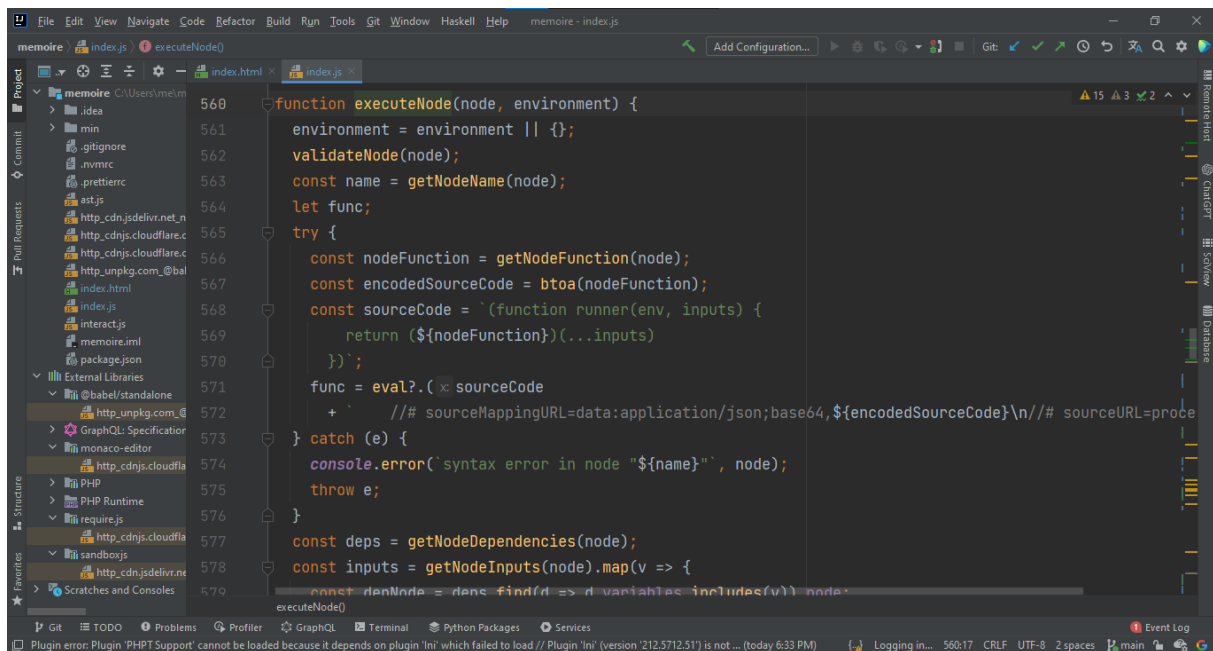


Figure 5.3 – Interface de IntelliJ IDEA Ultimate

XAMPP

XAMPP est une solution de pile technologique de serveur web gratuite et open-source qui inclut Apache, MariaDB, PHP et Perl, facilitant ainsi la configuration d'un environnement de serveur local pour le développement web. Il dispose d'un panneau de contrôle convivial, prend en charge plusieurs plates-formes et regroupe tous les composants nécessaires pour une installation et une utilisation rapides. Idéal pour le développement, les tests et l'apprentissage locaux, XAMPP permet aux développeurs de gérer efficacement les serveurs web et les bases de données sur leurs machines locales. [13].

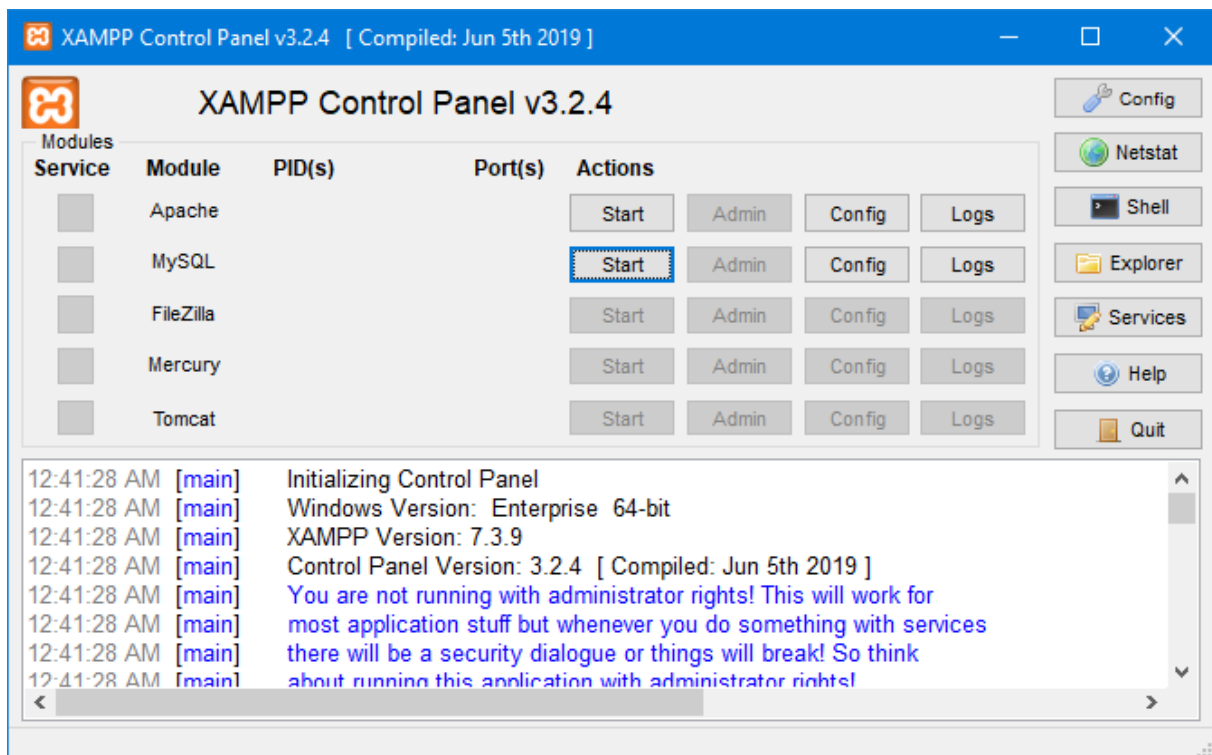


Figure 5.4 – Interface de XAMPP.

5.3 Technologies Utilisées

Base de données MySQL

MySQL est un système de gestion de base de données relationnelle open-source largement utilisé, connu pour sa fiabilité, sa facilité d'utilisation et ses performances. Il prend en charge SQL (Structured Query Language) pour la gestion et la manipulation des bases de données, ce qui en fait un choix populaire pour les applications web, l'entreposage de données et les plateformes de commerce électronique. MySQL offre des fonctionnalités telles qu'une sécurité des données robuste, un support transactionnel et une évolutivité, répondant aux besoins des petites et grandes applications. Il s'intègre parfaitement avec divers langages de programmation et plates-formes, ce qui en fait une solution polyvalente pour les développeurs et les entreprises pour gérer efficacement leurs données.



Figure 5.5 – Logo MySQL.

TailWind CSS

Tailwind CSS est un framework CSS utilitaire qui fournit un ensemble de classes prédéfinies pour styliser rapidement les applications web sans écrire de CSS personnalisé. Il met l'accent sur l'utilisation de petites classes utilitaires réutilisables pour créer des designs personnalisés, favorisant ainsi l'efficacité et la maintenabilité. Hautement personnalisable, Tailwind permet aux développeurs de configurer le système de design, y compris les couleurs, les espacements et la typographie. Il inclut également des utilitaires de conception responsive et supprime automatiquement le CSS inutilisé dans les builds de production pour optimiser les performances. Tailwind CSS est idéal pour les développeurs recherchant une approche simplifiée et flexible du design web.



Figure 5.6 – Logo TailWind CSS.

Javascript

JavaScript est un langage de programmation polyvalent de haut niveau couramment utilisé dans le développement web pour créer des pages web interactives et dynamiques. Il est essentiellement utilisé dans la pile technologique web, aux côtés de HTML et CSS. JavaScript permet aux développeurs d'implémenter des fonctionnalités complexes telles que la validation de formulaires, des cartes interactives, des graphiques animés, et plus encore. Il peut être exécuté côté client dans les navigateurs web ou côté serveur en utilisant des environnements comme Node.js. Avec ses vastes bibliothèques et frameworks, tels que React, Angular et Vue.js, JavaScript améliore l'expérience utilisateur et améliore la fonctionnalité des applications web.



Figure 5.7 – Logo de Javascript.

NodeJS

Node.js est un puissant environnement d'exécution open-source qui permet aux développeurs d'exécuter du code JavaScript en dehors d'un navigateur web. Construit sur le moteur JavaScript V8 de Chrome, Node.js est conçu pour la création d'applications réseau évolutives et efficaces, notamment des serveurs et des applications web. Il utilise un modèle d'entrées-sorties événementiel et non bloquant, ce qui le rend léger et efficace, idéal pour gérer de multiples connexions simultanées. Node.js est largement utilisé pour développer des applications en temps réel, telles que des applications de chat, des API RESTful et des applications intensives en données nécessitant un back-end rapide et réactif.



Figure 5.8 – Logo NodeJS

Vue.js

Vue.js est un framework JavaScript progressif utilisé pour créer des interfaces utilisateur et des applications à page unique. Il est conçu pour être adoptable de manière progressive, ce qui signifie que vous pouvez l'utiliser comme une bibliothèque pour améliorer des parties spécifiques de votre projet ou comme un framework complet pour des applications plus complexes. Vue.js propose un système de liaison de données réactif, une architecture basée sur les composants, et met l'accent sur la simplicité et les performances. Sa courbe d'apprentissage douce et sa documentation complète en font un choix populaire parmi les développeurs pour créer des interfaces web dynamiques et engageantes.



Figure 5.9 – Logo Vue.js.

NPM

npm (Node Package Manager) est le gestionnaire de paquets par défaut de Node.js, permettant aux développeurs d'installer, de partager et de gérer les dépendances de leurs projets JavaScript. Il offre un accès à un vaste référentiel de bibliothèques et d'outils open-source, facilitant l'intégration de modules tiers dans vos applications. Avec npm, les développeurs peuvent gérer efficacement la gestion des versions, des dépendances et l'automatisation des scripts, rationalisant ainsi le flux de travail de développement. Il s'agit d'un outil crucial pour le développement JavaScript moderne, facilitant le partage et la réutilisation de code au sein de la communauté des développeurs.



Figure 5.10 – Logo NPM.

5.4 Interfaces Graphiques

Dans la partie suivante, nous présenterons les principales interfaces de notre système.

5.4.1 Écran Principal de l'Éditeur

Cet écran contient une liste des objets que nous utilisons pour construire un réseau de nœuds et de flèches.

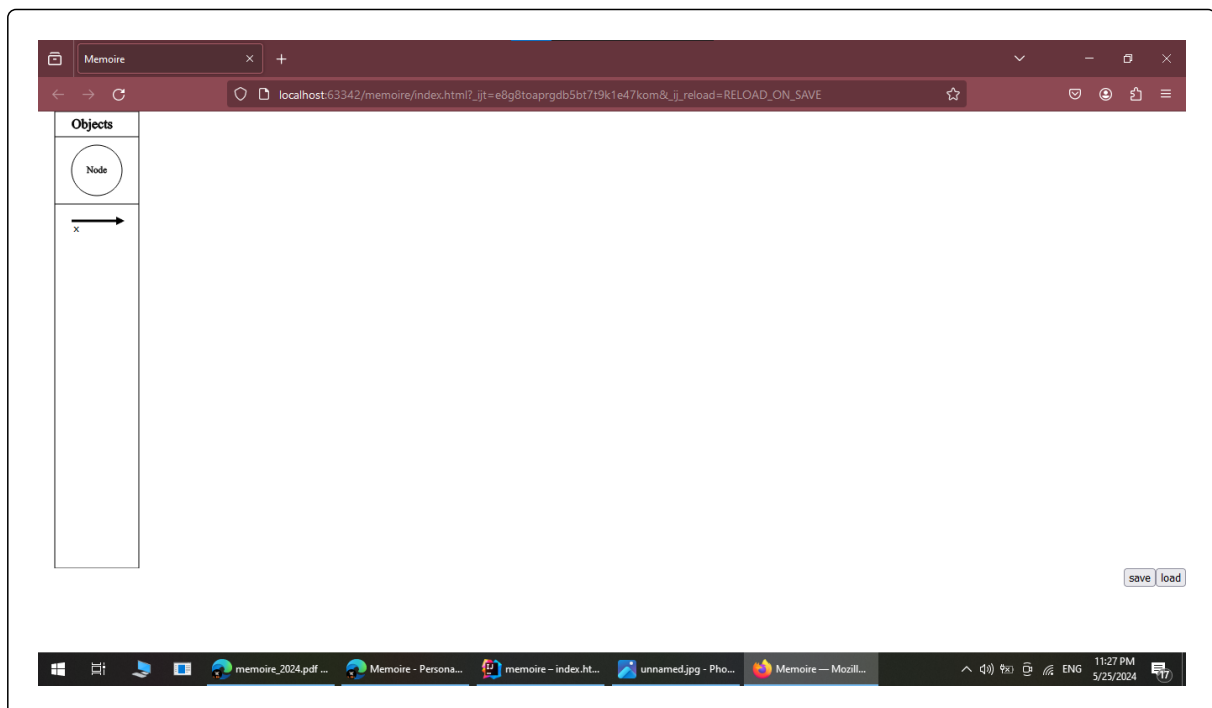


Figure 5.11 – Éditeur principal de l'application

5.4.2 Ajout de nœud

Nous faisons glisser le nœud de la liste et le plaçons dans l'interface pour créer notre réseau en spécifiant le nom du nœud.

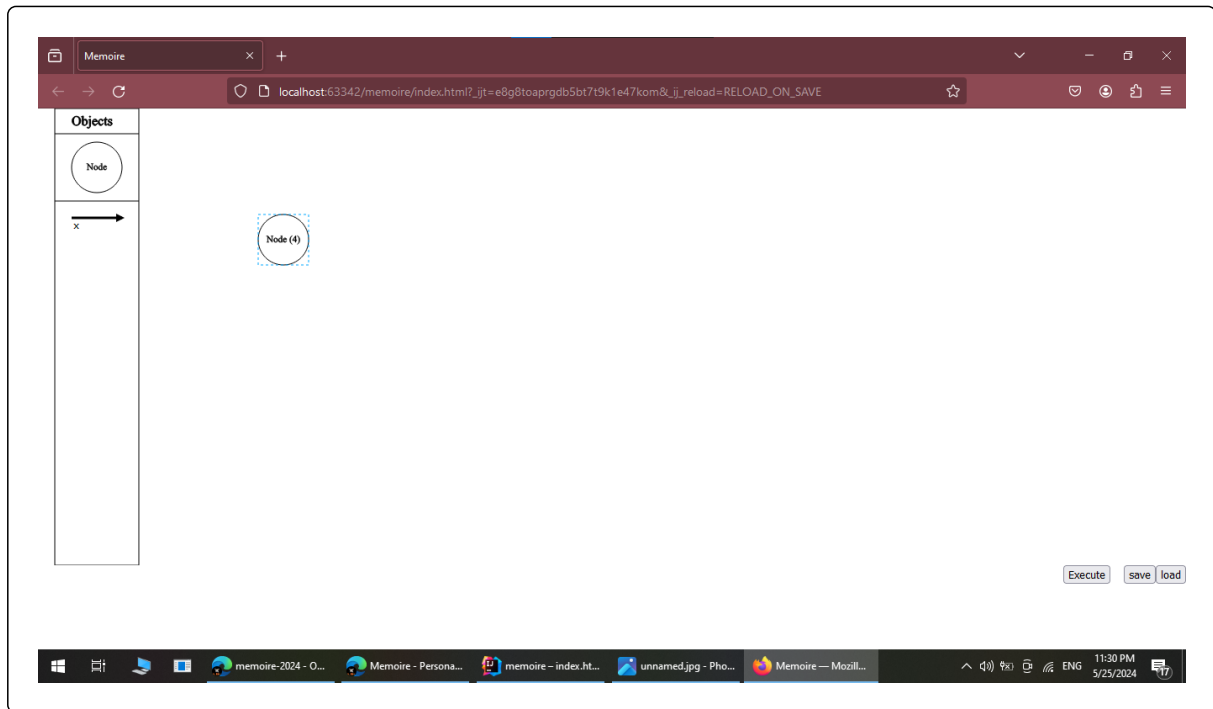


Figure 5.12 – Après l'ajout de 1 nœud

5.4.3 Modifier le nœud

Un nœud est une fonction. Pour changer la fonction du nœud, il suffit de cliquer sur le nœud et la fenêtre de modification s'ouvre, ce qui permet d'ajouter et de modifier les fonctions, les entrées et les sorties du nœud.

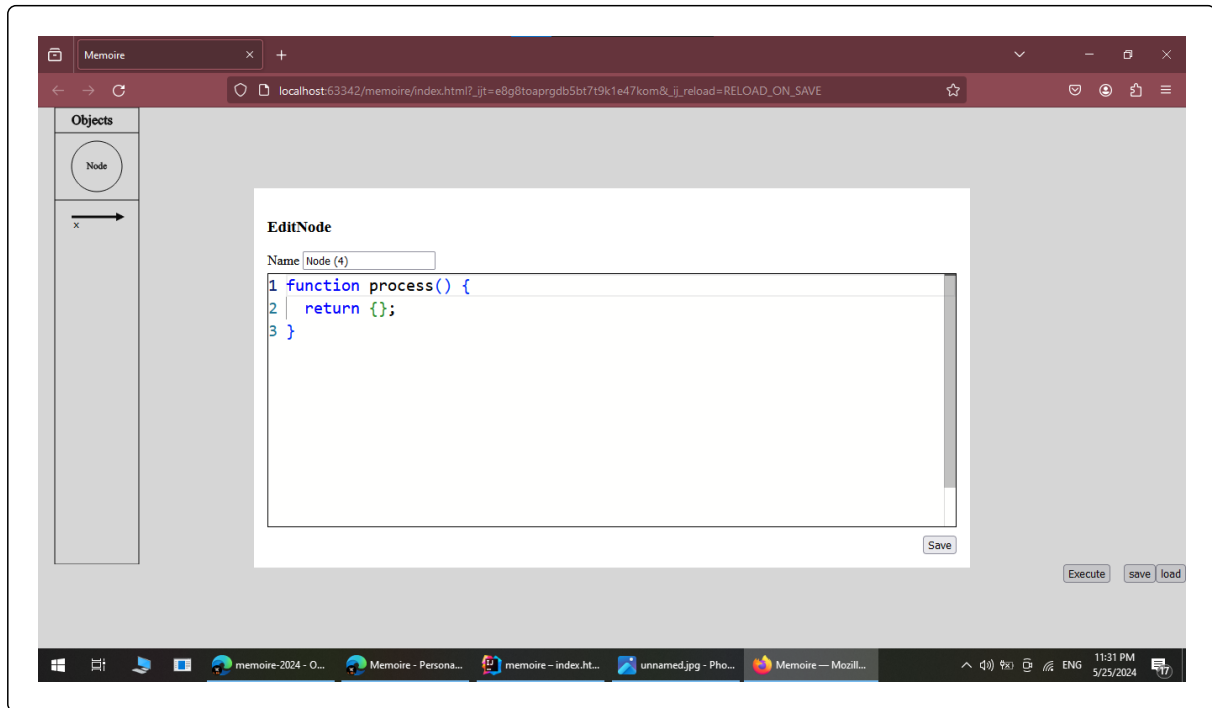


Figure 5.13 – Modifier le nœud

5.4.4 Enregistrer le nœud modifié

Pour enregistrer toutes les modifications des nœuds, nous devons cliquer sur le bouton Enregistrer et terminer le reste des modifications sur tous les nœuds selon les besoins.

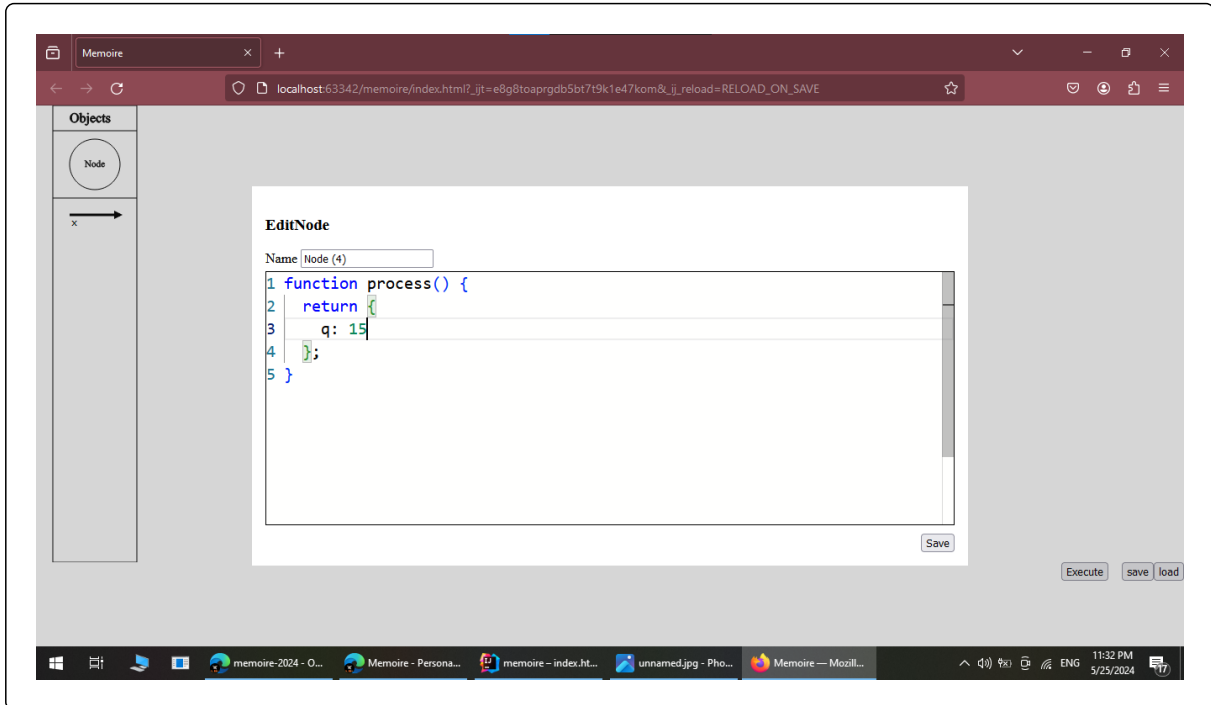


Figure 5.14 – Enregistrer le nœud modifié

5.4.5 Réseau final à exécuter

Nous créons plus de nœuds et configurons les connexions et les fonctions des nœuds. Nous pouvons également configurer des paramètres temporels si nécessaire.

Figure 5.15 – Réseau complet

5.4.6 Rapport d'exécution

Après l'exécution de la simulation 2 fois, nous obtiendrons ce rapport d'exécution. Ici, nous pouvons comparer les différentes sorties en fonction des différentes entrées, ce qui peut aider l'utilisateur à choisir la bonne décision.

	Inputs			Outputs		
		Sim1	Sim2		Sim1	Sim2
BasePrice				basePrice	100	150
Quantity				quantity	10	50
Tax	basePrice	100	150	taxRate	0.10	0.08
	quantity	10	50			
Discount	basePrice	100	150	discount	0	30
	quantity	10	50			
FinalPrice	Tax	0.10	0.08	finalPrice	1100	6480
	Discount	0	30			
	basePrice	100	150			
	quantity	10	50			

Figure 5.16 – Rapport d'exécution pour 2 exécutions de simulation

5.5 Conclusion Générale

L'objectif de cette thèse est de proposer et de concevoir un réseau de décision et un système de simulation de l'environnement économique et social basé sur le réseau proposé. Notre thèse se conclut en soulignant l'importance et l'utilité des Réseaux de Décision Dynamiques (DDN) dans la simulation des interactions économiques et sociales complexes. À travers la conception détaillée, la mise en œuvre et l'analyse fournies dans notre travail, il est évident que les DDN offrent un cadre robuste pour comprendre et optimiser les processus de prise de décision dans des environnements dynamiques.

Les principales contributions de notre thèse sont:

- **Modélisation Complète:** Nous démontrons avec succès l'utilisation de différents diagrammes UML pour modéliser le réseau de décision, posant ainsi une base solide pour l'implémentation. L'utilisation des diagrammes de cas d'utilisation, d'activité, de séquence et de classe offre une approche claire et structurée pour capturer les fonctionnalités et les interactions au sein du système.
- **Implémentation Technique:** Nous détaillons les aspects techniques de la mise en œuvre du réseau de décision, notamment la gestion des nœuds et des connexions, l'exécution de la logique de décision, et la gestion du comportement dynamique à travers la modélisation mathématique. Cela assure une compréhension claire du fonctionnement interne du système et facilite sa reproduction et son développement ultérieur.
- **Simulation et Analyse:** En incorporant des fonctionnalités telles que les sorties dépendantes du temps, la gestion des boucles et les sorties différées, nous montrons comment le système peut simuler des scénarios du monde réel et fournir une analyse perspicace à travers des rapports complets. Cela met en évidence les applications pratiques du réseau de décision dans divers domaines tels que l'économie, la stratégie d'entreprise et les sciences sociales.
- **Développement et Outils:** L'utilisation d'environnements de développement modernes et d'outils tels que Visual Studio Code et IntelliJ IDEA garantit que le système est construit sur une plateforme robuste et flexible. Ce choix d'outils soutient un développement, un débogage et une collaboration efficaces, améliorant ainsi la fiabilité et la scalabilité du système.

En conclusion, notre thèse propose une approche approfondie et bien structurée pour concevoir et implémenter un réseau de décision en utilisant les DDNs. Elle aborde les complexités des interactions économiques et sociales, offrant un outil puissant aux décideurs pour simuler, analyser et optimiser leurs stratégies dans un environnement dynamique et en évolution. Ce travail pose les bases pour la recherche et le développement futurs dans le domaine des systèmes d'aide à la décision, avec des applications potentielles dans divers domaines.

Bibliographie

- [3] Roger M. Cooke. In: *Journal of the American Statistical Association* 109.507 (2014).
- [10] J. Q. Smith. “Influence Diagrams for Statistical Modelling.” In: *The Annals of Statistics* 17.2 (1989), pp. 654–672.

Webographie

- [1] bayesserver.com. *Bayes Server Learning Center / Dynamic Bayesian networks - an introduction*. URL: <https://www.bayesserver.com/docs/introduction/dynamic-bayesian-networks/>.
- [2] Irad Ben-Gal. *Bayesian Networks*. URL: <https://doi.org/10.1002/9780470061572.eqr089> (visited on 01/16/2024).
- [4] le-dictionnaire.com. *Dictionnaire / Définition DÉCISION*. URL: <https://www.le-dictionnaire.com/definition/d%C3%A9cision> (visited on 01/16/2024).
- [5] Draw.io. URL: <https://app.diagrams.net/>.
- [6] edrawmax.com. *What is The Influence Diagram*. URL: <https://www.edrawmax.com/article/what-is-influence-diagram.html>.
- [7] IntelliJ IDEA Ultimate. URL: <https://www.jetbrains.com/idea/>.
- [8] Lassaad Mlk. *La Decision Dans L'entreprise*. URL: <https://www.scribd.com/document/547124447/LA-DECISION-DANS-L-ENTREPRISE?fbclid=IwAR0TGYi6wq8eWNM81Ne49wDC1JoNjX1alEq-UMG09MgDv5DrA> (visited on 01/14/2024).
- [9] scikit-learn.org. *Decision Trees*. URL: <https://scikit-learn.org/stable/modules/tree.html> (visited on 03/30/2024).
- [11] Visual Studio Code. URL: <https://code.visualstudio.com/>.
- [12] www.geeksforgeeks.org. *What are the advantages and disadvantages of UML?* URL: <https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-uml/>.
- [13] Xampp. URL: <https://www.apachefriends.org/>.