



HAL
open science

Designing an architecture web service oriented for Collaborative Decision Making process facilitation support

Sid Ahmed Elandaloussi

► **To cite this version:**

Sid Ahmed Elandaloussi. Designing an architecture web service oriented for Collaborative Decision Making process facilitation support. Intelligence artificielle [cs.AI]. Université d'Oran, 2021. Français. NNT: . tel-03727482

HAL Id: tel-03727482

<https://hal.science/tel-03727482>

Submitted on 19 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FSEA2020/2021

THÈSE

En vue de l'obtention du diplôme
de Doctorat en sciences

Spécialité Informatique

Présentée par: **ELANDALOUSSI Sidahmed**

**Développement d'une architecture orientée services pour un
outil d'aide à la facilitation pour des processus de décision
collaborative**

Devant le jury compose de:

BELALEM Ghalem	Professeur, Université d'Oran1 Ahmed Ben Bella	President
SEKHRI Larbi	Professeur, Université d'Oran1 Ahmed Ben Bella	Examineur
BELKADI Khaled	Professeur, Université Mohamed Boudiaf, USTO	Examineur
BOUCHIHA Djelloul	Professeur, centre universitaire Sahli Ahmed, Naama	Examineur
TAGHEZOUT Noria	Professeure, Université d'Oran1 Ahmed Ben Bella	Directrice de these
ZARATE Pascale	Professeure, Université Toulouse1, Capitole	Codirectrice de these

Remerciement

Je présente toute ma gratitude au professeure TAGHEZOUT Noria, ma directrice de thèse, qui a été mon soutien indéfectible par ses conseils et ses orientations sur la conception de ce travail. Mes remerciements s'adresse également au professeure ZARATE Pascale ma codirectrice de thèse de l'accompagnement scientifique et humain qu'elle ma apporté et surtout le soutien moral pour aller de l'avant durant toute ma période de stage au sein du laboratoire IRIT. Quelles trouvent ici mon témoignage d'estime et de respect.

Je tiens à remercier tout particulièrement Mr BELALEM Ghalem, professeur à l'université d'Oran1 Ahmed Ben Bella, pour m'avoir fait l'honneur de présider ce jury. Aussi, je remercie vraiment Mr SEKHRI Larbi, Professeur à l'université d'Oran1 Ahmed Ben Bella, Mr BELKADI Khaled, professeur a l'université des sciences et de la technologie d'Oran, Mr BOUCHIHA Djelloul, Professeur au centre universitaire Sahli Ahmed de Naama pour avoir accepté de faire partie de mon jury de thèse autant que rapporteurs.

Je tiens également à exprimer toute ma reconnaissance a mes chers parents ainsi que mes sœurs et amis qui mont soutenu moralement durant toutes ces années de recherche pour la réalisation de ce travail.

Je ne pourrais pas terminer ces remerciements sans dire un petit mot à mon adorable neveu WAEL la source de joie dans notre famille, que Allah le protège.

Résumé

L'infrastructure informatique regroupe l'ensemble des équipements matériels (Serveurs, Commutateur, Routeur, poste travail, périphériques.....) et logiciels(Cloud, ERP, CRM, Messagerie.....) d'une entreprise. Ces éléments sont connectés entre eux pour fournir des services à un utilisateur final (employés, partenaires et / ou clients).En outre, l'interruption d'un telle service peut entraîner des pénalités financières importantes et / ou une perte de confiance des utilisateurs. En effet, il est primordial de réagir rapidement afin de rendre le délai d'interruption d'activité le plus court possible et limiter cet impact financier.

Par conséquent, pour s'assurer le bon fonctionnement, disponibilité et la sécurité de n'importe quelle infrastructure informatique nous devons être proactifs dans la détection des incidents qui perturbent la stabilité de l'entreprise.Cette mission implique également de prodiguer des conseils aux usagers, de répondre aux demande spécifiques et de s'adapter en permanence aux évolutions et besoins de l'entreprise.

L'objectif principal de notre travail de recherche est de proposer une plateforme de diagnostic efficace pour une Infrastructure informatique appelée (**MAITD**) afin d'aider les techniciens de l'entreprise à trouver une solution (s) en un temps réduit face à un problème de prise de décision.

Mot Clés : Systèmes multi critères, Systèmes multi agents, Infrastructure Informatique, maintenance corrective, Mesure de similarité, Extractions des termes, fouilles de textes, Fonction d'agrégation, Architecture orienté services, Ontologies.

Table des matières

Remerciement	
Résumé	
Liste des Figures	
Liste des Tableaux	
Introduction Générale	12
Chapitre1 : La maintenance de l'infrastructure Informatique	
I. Les techniques de la maintenance informatique	
1. Introduction	17
2. Concept et fonction de la maintenance	17
3. Niveau de la maintenance informatique	17
4. Type de la maintenance informatique	18
4.1 La maintenance logicielle	18
4.2 La maintenance matérielle	19
4.2.1 La maintenance Préventive	20
4.2.2 La maintenance Corrective	22
4.2.3 La maintenance a méliorative	24
II. Les outils logiciels pour la maintenance informatique	
1. Introduction	24
2. Les technologies des services WEB	25
2.1 Le protocole SOAP	25
2.2 Le langage de description WSDL	26
2.3 Le registre UDDI	27
2.4 Le langage XML	29
3. Le fonctionnement général d'un service WEB	30
III. Conclusion	31
Chapitre2 : Vers une approche collaborative multicritère de décision	
1. Introduction	33
2. Système multi agent	33
2.1 Introduction	33
2.2 Agent et système multi agent	33
2.3 Type d'agent	34
2.4 La conception des SMA	34
2.5 La communication entre agent	36
3. Aide à la décision multi critères	37
3.1. Types de problèmes de décision	38

3.2. La méthode d'analyse multicritère.....	39
3.3. Les bases méthodologiques.....	39
4. La décision collaborative.....	40
4.1. La notion d'aide à la décision.....	40
4.2. Définition de la collaboration.....	40
4.3. Processus de décision collaborative.....	40
4.4. Outils de collaboration.....	41
4.5. Caractéristique des systèmes d'aide a la décision.....	42
5. Opérateurs d'agrégations.....	43
5.1. Définition d'un opérateur d'agrégation.....	43
5.2. Types d'opérateurs d'agrégation.....	43
5.3. Méthode d'agrégation.....	45
6. Conclusion.....	46

Chapitre3 : Techniques de traitement des données

1. Introduction.....	48
2. Les méthodes d'extraction des termes.....	48
2.1 Les techniques d'extraction des termes.....	48
2.1.1 Approches statistiques.....	48
2.1.2 Approche linguistiques.....	49
2.1.3 Approche par regroupement.....	50
2.1.4 Approche à base de graphe.....	51
3. Mesure de similarité.....	52
3.1 Introduction.....	52
3.2 Type de similarité.....	53
3.2.1 Similarité syntaxique.....	53
3.2.2 Avantages et inconvénients.....	55
3.2.3 Similarité sémantique.....	55
3.2.1.1 Approche vectorielles.....	55
3.2.1.2 Approche Ontologique.....	56
3.2.1.3 Approche statistique.....	59
3.2.1.4 Avantages et inconvénients.....	61
4. Conclusion.....	61

Chapitre4 : Conception du MAITD

1. Introduction.....	63
2. Architecture Proposée.....	63
2.1 La couche présentation.....	63
2.2 La couche interprétation.....	65

2.3 La couche de donnée.....	69
3. Architecture MAITD a base de module.....	72
3.1 Prétraitement des données.....	73
3.2 Eextraction des termes.....	73
3.3 La similarité des données.....	74
3.4 La résolution du problème	74
4. Extraction des Termes.....	74
4.1 La Pertinence dans le domaine	74
4.2 La fréquence des termes TF-IPF	75
5. La mesure de similarité textuelle	75
5.1 La mésure de similarité Lexical.....	75
5.2 La mésure de similarité sémantique	76
6. Agrégation des Solutions.....	77
7. Conclusion	79

Chapitre5 : Expérimentation et Validation

1. Introduction.....	81
2. La description d'un problème	81
3. Prétraitement de donnée	81
4. La pertinence dans le domaine	82
5. La Composition des termes	83
6. Mesure de similarité lexicale.....	83
7. Mesure de similarité sémantique	84
7.1 Approche a base de Corpus	84
7.2 Construction de notre Ontologie du Domaine	85
7.3 Enrichissement de notre ontologie	89
7.4 Détermination du poids de concept dans une ontologie.....	90
8. Agrégation des solutions	91
9. Préparation d'une session de collaboration	91
10. Supervision de diagnostic du problème.....	92
11. Validation et evaluation des solutions	92
11.1 La validation F-measure.....	92
11.2 La méthodologie de validation.....	93
12. Conclusion	95
Conclusion Générale et perspectives.....	96
Référence Bibliographique	98

Liste des Figures

1. Type de la maintenance logiciel	19
2. Type de la maintenance matérielle	19
3. Structure d'un message SOAP	25
4. Structure d'un document WSDL.....	27
5. Interaction de l'annuaire UDDI	27
6. Schéma Document XML	29
7. Fonctionnement générale d'un service WEB	30
8. L'approche Voyelle d'un SMA	36
9. Communication par partage d'information	37
10. Communication par envoie de message	37
11. Processus de prise de décision.....	41
12. Structure interne de l'agent participant	64
13. La structure interne de l'agent expert	65
14. La structure interne de l'agent Analyseur	66
15. Diagramme d'UML d'activité	66
16. Diagramme de Séquence de MAITD(Scénario1).....	68
17. Diagramme de Séquence de MAITD (Scénario2)	69
18. La structure générale d'un WSDL	70
19. Architecture multi couche à base d'agent	71
20. l'architecture MAITD2 a base de module	73
21. Identification du Probleme	81
22. Prétraitement de donnée (Couche1)	82

23. Pertinence dans le domaine	82
24. La Fréquence des Termes TF-IPF	83
25. Mesure de similarité lexicale	84
26. Mesure de similarité sémantique Termes à Termes	84
27. Définition des classe de notre ontologie initiale.....	87
28. Les individus de notre ontologie initiale.....	88
29. Propriété de donnée de notre ontologie.....	88
30. Exemple d'ontologie de domaine pour une infrastructure informatique.....	89
31. Agrégation des solutions avec l'algorithme WAS-PAS.....	91
32. Préparation Session Collaborative.....	91
33. Supervision de diagnostic du problème.....	92
34. Validation du MAITD	95

Liste des Tables

1. Exemple des paramètres mesurés pour les serveurs Cisco Blade (UCS)	21
2. Opération de la maintenance corrective.....	23
3. Identification de type de problématique	34
4. Comparaison entre Agent cognitif et Réactif	38
5. La description des balises d'un service Web	70
6. Description des flux d'informations de l'architecture MAITD.....	72
7. Exemple d'une matrice de similarité a base de chaine	76
8. Exemple d'une matrice de similarité sémantique à base de Corpus	77
9. Récapitulatif des concepts de diagnostic IT	86
10. Pourcentage de satisfaction des experts	94



Introduction Générale

La maintenance de l'infrastructure informatique comprend toutes les tâches et actions nécessaires à leur bon fonctionnement. Il existe deux grandes catégories de maintenance informatique: matérielle (Serveur, Poste travail, Routeur, Commutateur, Concentrateur, Périphériques...etc.) et logicielle (ERP, CRM, Cloud, Messagerie...etc.) ;ces dernières sont liées à plusieurs domaines tels que application, virtualisation, stockage, serveur, réseau, sécurité, systèmes, Support,... etc.

Un problème peut avoir lieu et traité en allant du plus simple au plus complexe. En effet, trouver la meilleure solution requiert plus de temps et un niveau de compétence élevé de l'utilisateur.Cette intervention est matérialisée par un nombre de services ; ces services peuvent être fournis sur site ou à distance rentrant dans le cadre d'une maintenance corrective [**M.D. Kent et al, 2017**] ou par des méthodes préventives à temps fixe [**X. Mehmeti et al, 2018**].Ainsi, selon ce contexte au sein de la maintenance des infrastructures informatiques, plusieurs logiciels propriétaires ont apparus comme: Nagios [**Nagios, 2007**], Manage Engine [**Manage Engine, 2007**] et Zenoss [**Zenoss, 2018**]...etc.

Cependant, toutes ces solutions d'infrastructure Informatique sont préventives et limitées lorsque le problème est déjà survenu. Récemment, le but de la tâche de maintenance est de calculer les besoins de maintenance avant que la panne de l'équipement se produise, c'est-à-dire une surveillance continue [**P. Callewaerta, 2017**] dans le but d'améliorer les processus de contrôles, et de prévenir ou détecter les fraudes dans une plate-forme informatique IT.

De plus, des systèmes d'aide à la décision ont été développés pour aider les décideurs à résoudre leurs problèmes de manière préventive et corrective grâce à des méthodes supervisées [**WWW4, 2018**] ou non supervisées [**H.J Liao et al, 2013**] [**J.Campos et al, 2007**].Ces méthodes doit se traduire par une amélioration de la qualité de prise de décision plutôt que par une amélioration de son efficacité. L'évolution des systèmes d'aide à la décision oriente de plus en plus vers des architectures avec services Web. Les services web ont émergé comme un mécanisme puissant assurant l'interaction entre des fournisseurs de services et des consommateurs sur différentes plates-formes technologiques sur le web. Par conséquent, Les entreprises peuvent installer et exploiter ces services avec un minimum de programmation.De plus, leurs flexibilités permettent à l'entreprise et à d'autres applications de les découvrir et les appeler une fois qu'ils ont déployé.

Notre étude se focalise sur le développement d'un système d'aide à la décision corrective (DSS) a base de service web avec une approche semi-supervisée pour réduire le temps de résolution des problèmes. L'objectif principal de ce travail est de fournir une assistance automatisée à l'utilisateur afin d'assurer l'ajout des nouvelles informations au système.

Problématique

Introduction Générale

La description des problèmes n'a jamais été considérée comme parfaite et ceci est dû principalement aux éléments suivants :

- ✓ Niveau d'expertise de l'utilisateur et,
- ✓ La structure de la base de données qui sera interrogée (mots-clés, langage de requête structuré, dictionnaire... etc).

Néanmoins, ce fait nous amène à trouver des solutions insatisfaisantes, par conséquent, nous devons traiter les données saisies par l'utilisateur différemment avant de chercher la solution.

Cette dernière dépend souvent de :

- ✓ La qualité de la description du problème et,
- ✓ Le niveau de connaissance des utilisateurs.
- ✓ La manière de procéder afin de trouver la solution (Collaboration, négociation...)

Dans certaines situations, la description du problème peut être interprétée de différentes manières. Dans une autre situation, les termes pertinents n'apparaissent pas dans le problème défini. Par conséquent, ces faits nous conduisent à des solutions insatisfaisantes. Pour remédier à tout cela, nous avons établi une stratégie dans notre travail de recherche dans laquelle nous essayons répondre aux questions suivantes :

- ✓ Comment filtrer et traiter efficacement les problèmes entrants afin d'en extraire les termes pertinents qui devraient couvrir l'ensemble de la description du problème ?
- ✓ Comment trouver d'une façon non supervisée des solutions similaires à notre cas de diagnostic ?
- ✓ Comment agréger toutes les solutions candidates avec l'attribution des poids automatiquement pour chaque critère qui rentre dans le choix de la solution ?
- ✓ Comment bénéficier des avantages des architectures orientés service dans l'élaboration ou l'implémentation de la solution ?

Contribution

Les Systèmes Interactifs d'Aide à la Décision (SIAD) sont développés afin d'aider au mieux les décideurs dans les organisations. Cette aide doit se traduire par une amélioration de la qualité de prise de décision plutôt que par une amélioration de son efficacité. Le processus de décision assisté par des SIAD est alors envisagé comme une succession d'aller / retour entre le décideur (utilisateur) et la machine. En effet, l'automatisation complète ou partielle de tels processus constitue un défi pour nous durant la réalisation de ce travail.

Dans cette thèse, nous proposons un système d'aide à la décision collaboratif en incluant l'exploration de texte afin d'explorer et analyser toutes les données non structurées dans les systèmes décisionnels en se basant sur des agents pour la maintenance de l'infrastructure informatique(II).

Un tel système multi-agents est capable de coordonner et de coopérer son action afin d'aboutir à l'élaboration d'une méthode de décision correctrice pour la maintenance de l'infrastructure informatique. De plus, les méthodes de fouille de texte (Text-Mining) sont capables d'éliminer l'apparition des termes manifestement incorrects ou non pertinents pour un problème défini. De telles sortes, qu'ils ne conservent que les termes pertinents pour réduire le temps de réponse lors d'une tâche de dépannage. De ce fait, nous nous permettons de traiter les données générées (termes pertinents) comme un problème multicritères, qui est mieux adapté à ce type de situation avec l'attribution des poids automatiquement pour chaque critère en utilisant notre ontologie de domaine déjà créer et qui est définit comme un service web. De plus, la modularité et le parallélisme des agents permettent aussi d'accélérer l'action de trouver la bonne solution qui sera jugé dans notre étape de validation par des experts dans le domaine étudié.

Plan de Mémoire

Le mémoire est regroupé en deux grandes parties, la première est composée de trois chapitres qui sont dédiés à l'état de l'art des travaux existants qui se rapportent à notre problématique dans les domaines de la maintenance, de la décision collaborative et des techniques de traitement de données. La deuxième partie est consacrée à la conception et la réalisation de notre contribution. Elle est composée des chapitres 4 et 5. La structuration de cette thèse est comme suit:

Le chapitre1 est intitulé La maintenance de l'infrastructure informatique donne et présente un état de l'art sur le paradigme maintenance ainsi que leur application dans le domaine des infrastructures informatiques.

Le chapitre2 est intitulé Vers une approche collaborative de la décision et met l'accent sur les paradigmes agent et les systèmes multi-agents ainsi que leur application dans notre domaine d'étude. Nous discutons aussi la notion de collaboration et le traitement de la donnée multi

critères face à un problème décisionnel. Les opérateurs d'agrégation ont été présentés aussi dans ce chapitre. Un ensemble d'algorithmes dédié à ce genre de problèmes est présenté à cet effet.

Le chapitre 3 est intitulé Techniques du traitement des données et décrit les principales approches d'extraction des termes, ainsi que les différents techniques de calcul des similarités qui sont utilisées pour répondre à la contrainte de la mauvaise identification du problème.

Le chapitre 4 constitue le cœur de notre approche et décrit notre contribution théorique. La conception et la modélisation de notre système d'aide à la décision (**MAITD**) seront données en détail.

Le dernier chapitre présente le cadre applicatif de nos travaux ainsi que les expérimentations mises en place pour évaluer nos propositions.

Le présent manuscrit se termine par une conclusion générale qui conclut nos travaux et énumère quelques perspectives de recherches futures.

Des annexes seront présentées à la fin de ce document pour rappeler quelques notions citées tout au long du travail.

Chapitre1 :

La maintenance de l'infrastructure Informatique

Plan du Chapitre1

- I. Les techniques de la maintenance informatique
 1. Introduction
 2. Concept et fonction de la maintenance
 3. Niveau de la maintenance informatique
 4. Type de la maintenance informatique
 - 4.1 La maintenance logicielle
 - 4.2 La maintenance matérielle
 - 4.2.1 La maintenance Préventive
 - 4.2.2 La maintenance Corrective
 - 4.2.3 La maintenance a améliorative
- II. Les outils logiciels pour la maintenance informatique
 1. Introduction
 2. Les technologies des services WEB
 - 1.1 Le protocole SOAP
 - 1.2 Le langage de description WSDL
 - 1.3 Le registre UDDI
 - 1.4 Le langage XML
 3. Le fonctionnement général d'un service WEB.
- III. Conclusion

I. Les techniques de la maintenance informatique

1. Introduction

Les processus technologiques de ces dernières années touchent de nombreux domaines d'activités, notamment les infrastructures informatiques. De plus, leur maintenance est la tâche la plus critique et nécessite un effort au préalable. Elle garantit le bon fonctionnement des infrastructures et assure leurs fiabilités afin qu'elles soient performantes. Toute les entreprises (petite /grande) d'aujourd'hui possède un parc informatique contenant au moins un serveur physique ou virtuel et un système applicatif qui va avec.

Les objectifs associés à la maintenance sont divers, mais peuvent être regroupés en deux grands axes : l'assurance du bon fonctionnement de l'infrastructure et le traitement des changements et des améliorations.

Ce chapitre présente les concepts de base de la maintenance informatique ainsi que les différents outils logiciels pour la mise en place de ce type de maintenance.

2. Concept et fonction de la maintenance

La maintenance garantit le bon fonctionnement de notre infrastructure et nous permet ainsi un bon accomplissement du travail que nous devons effectuer. Cela génère alors un gain de temps énorme. De plus, la méthode utilisée pour chercher la panne entraîne une perte de temps et prise de risque si l'on agit de manière désordonnée ou si l'on pose un mauvais diagnostic.

3. Les Niveaux de la maintenance informatique

Une meilleure connaissance des différents niveaux de maintenance corrective et préventive vous permet d'améliorer la gestion des interventions sur l'infrastructure.

Selon la norme française **AFNORX60-010**, chacun de ces 5 niveaux définit ci-dessous correspond au degré de complexité de l'opération de maintenance à effectuer, ainsi qu'au niveau d'expertise requis. Les cinq niveaux de maintenance présentés dans [R .Yende, 2018] ont été résumés dans ce qui suit :

3.1 Niveau 1 [R .Yende, 2018]

Le niveau1 correspond aux réglages simples prévus par le constructeur au moyen d'éléments accessibles sans aucun démontage ou ouverture de l'équipement, ou échanges d'éléments consommables accessibles en toute sécurité, tels que voyants ou certains fusibles, etc. Ce type d'intervention peut être effectué par l'exploitant du bien, sur place, sans outillage et à l'aide des instructions d'utilisation. Le stock de pièces consommables nécessaires est très faible.

3.2 Niveau 2 [R .Yende, 2018]

Ce niveau correspond aux dépannages par échange standard des éléments prévus à cet effet et opérations mineures de maintenance préventive, tels que graissage ou contrôle de bon fonctionnement. Ce type d'intervention peut être effectué par un technicien habilité de qualification moyenne, sur place, avec l'outillage portable défini par les instructions de

maintenance, et à l'aide de ces mêmes instructions. On peut se procurer les pièces de rechange transportables nécessaires sans délai et à proximité immédiate du lieu d'exploitation.

3.3 Niveau 3 [R .Yende, 2018]

Ce niveau est lié à l'identification et le diagnostic des pannes, les réparations par échange de composants ou d'éléments fonctionnels, les réparations mécaniques mineures, et toutes opérations courantes de maintenance préventive tels que réglage général ou réaligement des appareils de mesure. Ce type d'intervention peut être effectué par un technicien spécialisé, sur place ou dans le local de maintenance, à l'aide de l'outillage prévu dans les instructions de maintenance ainsi que des appareils de mesure et de réglage, et éventuellement des bancs d'essais et de contrôle des équipements et en utilisant l'ensemble de la documentation nécessaire à la maintenance du bien ainsi que les pièces approvisionnées par le magasin.

3.4 Niveau 4 [R .Yende, 2018]

À ce niveau tous les travaux importants de maintenance corrective ou préventive à l'exception de la rénovation et de la reconstruction sont prévus. Ce niveau comprend aussi le réglage des appareils de mesure utilisés pour la maintenance, et éventuellement la vérification des étalons de travail par les organismes spécialisés. Ce type d'intervention peut être effectué par une équipe comprenant un encadrement technique très spécialisé, dans un atelier spécialisé doté d'un outillage général (moyens mécaniques, de câblage, de nettoyage, etc.) et éventuellement des bancs de mesure et des étalons de travail nécessaires, à l'aide de toutes documentations générales ou particulières.

3.5 Niveau 5 [R .Yende, 2018]

Ce niveau est lié à la rénovation, reconstruction ou l'exécution des réparations importantes confiées à un atelier central ou à une unité extérieure. Ce type de travail est donc effectué par le constructeur, ou par le reconstruteur, avec des moyens définis par le constructeur et donc proches de la fabrication.

4. Les types de la maintenance informatique

La maintenance informatique est une démarche cruciale, quelle que soit la taille de l'infrastructure. En effet, lorsqu'elle est programmée de manière régulière, elle garantit le bon fonctionnement des terminaux et des périphériques utilisés. On distinguera alors 2 grandes catégories de maintenance informatique :

4.1 La maintenance logicielle

La **maintenance logicielle** vise à rétablir effectivement les composants logiciels (systèmes informatiques) d'un équipement informatique dans un état spécifié pour un service déterminé. Selon **SWEBOK (SOFTWARE ENGINEERING BODY OF KNOWLEDGE)** « C'est la totalité des activités qui sont requises afin de procurer un support au meilleur coût possible d'un logiciel, certaines activités avant la livraison du logiciel, donc pendant sa conception initiale mais la majorité des activités ont lieu après la livraison finale. La maintenance logicielle se subdivise en trois types : **la maintenance corrective**, **la maintenance adaptative** et **la maintenance évolutive**.

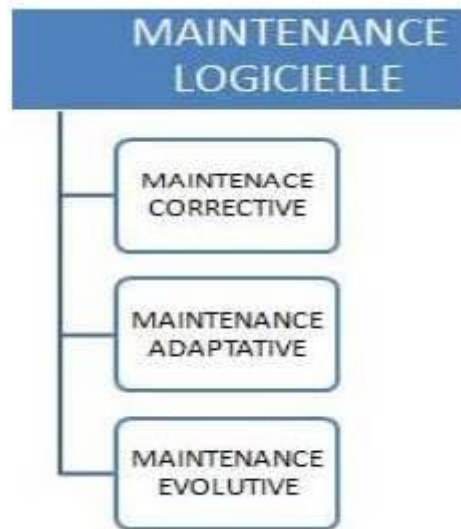


Figure1 : Type de la maintenance logiciel.

- ✓ **La maintenance corrective** : consiste à corriger les défauts de fonctionnement ou les non-conformités d'un logiciel.
- ✓ **La maintenance adaptative** : consiste à adapter l'application sans changer la fonctionnalité du logiciel afin que celle-ci continue de fonctionner sur des versions plus récentes des logiciels de base, voire à faire migrer l'application sur de nouveaux logiciels de base.
- ✓ **La maintenance évolutive** consiste à modifier progressivement l'application logicielle en l'enrichissant de fonctions ou de modules supplémentaires, ou en remplaçant une fonction existante par une autre, voire en proposant une approche différente.

4.2 La maintenance matérielle

La maintenance matérielle consiste à rétablir physiquement les différents composants matériels d'un équipement informatique (Ordinateur, Serveur, équipement...). En effet elle permet d'éviter les pannes ou les dysfonctionnements liés à l'usure d'une pièce. Ce type de maintenance est divisé en trois grands axes comme le montre la Figure2.

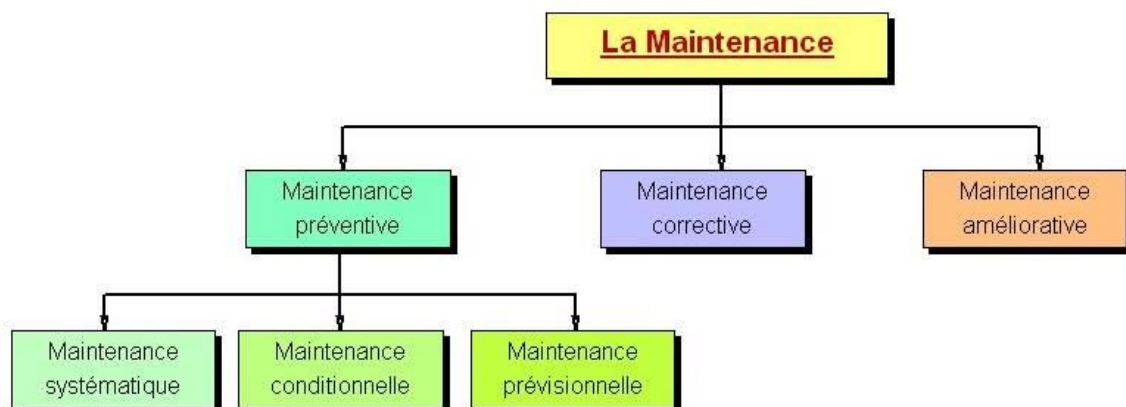


Figure2 : Types de la maintenance matérielle.

4.2.1 La maintenance Préventive

4.2.1.1 Définition et Objectifs

La maintenance préventive implique l'entretien et la révision des actifs organisationnels afin de les maintenir dans des conditions de fonctionnement optimales. Elle consiste à empêcher tout incident technique au moyen de la prévention dont l'objectif est de réduire la probabilité de défaillance ou la dégradation d'un service en cours d'utilisation, aussi afin de :

- Diminuer le budget de maintenance.
- Diminuer les temps d'arrêt en cas de révision ou de panne.
- Améliorer les conditions de travail du personnel de production.
- Augmenter la durée de vie des matériels.
- Prévenir et aussi prévoir les interventions coûteuses de maintenance corrective.
- Supprimer les causes d'accidents graves.

Selon **AFNOR**, la maintenance préventive [X. Mehmeti et al, 2018] est définie comme l'ensemble des opérations techniques exécutées à des intervalles prédéterminés ou selon des critères prescrits et destinés à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien. Plusieurs travaux ont élaboré ce type de maintenance dans des contextes différents, par exemple on cite le travail de [V.Jaramillo et al, 2020] qui ont développé une solution de maintenance préventive basée sur un modèle d'intelligence artificielle utilisant des données de supervision en temps réel dans l'industrie des transports maritimes.

4.2.1.2 Type de la maintenance préventive

Comme le montre la Figure 2 la maintenance préventive est divisée en trois autres types de maintenance : systématique, conditionnelle et prévisionnelle.

- ✓ **La maintenance préventive systématique** : Ce type de maintenance s'effectue selon un calendrier fixe établi par rapport à plusieurs facteurs (temps, nombre d'unités d'exploitation, nombre d'heures de fonctionnement..).
- ✓ **La maintenance préventive conditionnelle** : Elle repose sur la vérification et l'observation chronique de certains paramètres. Ce qui implique la définir comme étant une maintenance subordonnée à l'apparition d'un type d'événement prédéterminé (auto diagnostic, information d'un capteur, mesure d'une usure, etc.). D'un autre sens c'est une maintenance dépendante de l'expérience et faisant intervenir des informations recueillies en temps réel par la mise en évidence des points faibles. Suivant le cas, il est souhaitable de les mettre sous surveillance et, à partir de là, de décider d'une intervention lorsqu'un certain seuil est atteint.

Cette maintenance préventive conditionnelle se fait par des mesures pertinentes sur le matériel en fonctionnement. Par exemple, dans une infrastructure informatique les paramètres mesurés d'un serveur Cisco Blade peuvent porter sur (Voir Tableau 1).

Type deSupervision	Produits	Quoi Superviser?
Serveurs	Châssis(UCS)	Power/Temp/FAN
	Blade(UCS)	(1),HDD(Statut RAID/capacité)
	F.interconnect(UCS)	Networking (trafic flow Uplink + SAN Link/ Power /FAN
	F.Extender	Networking (Link status entre F.ext et F.interconnect

Tableau1: Exemple des paramètres mesurés pour les serveurs Cisco Blade (UCS)

4.2.1.3 Les opérations de la maintenance préventive

- a) **Les inspections** : Activités de surveillance consistant à relever périodiquement des anomalies et exécuter des réglages simples ne nécessitant pas d'outillage spécifique, ni d'arrêt de l'outil de production ou des équipements.
- b) **Les visites** : Ce type d'interventions correspondent à une liste d'opérations définies préalablement qui peuvent entraîner des démontages d'organes et une immobilisation du matériel. Une visite peut entraîner une action de maintenance corrective.
- c) **Les contrôles** : vérifications de conformité par rapport à des données préétablies suivies d'un jugement. Le contrôle peut :
 - Comporter une activité d'information.
 - Inclure une décision : acceptation, rejet, ajournement
 - Déboucher comme les visites sur des opérations de maintenance corrective
- d) **Les opérations de surveillance** (contrôles, visites, inspections) sont nécessaires pour maîtriser l'évolution de l'état réel du bien. Elles sont effectuées de manière continue ou à des intervalles prédéterminés ou non, calculés sur le temps, le nombre d'unités d'usage ou bien le nombre d'heures de fonctionnement.

4.2.1.4 Avantages et Inconvénients de la maintenance préventive

Parmi les avantages de ce type de maintenance,nous citons les plus importants comme suit :

- La facilité de gestion en prenant des intervalles d'intervention fixes,
- L'élimination de tout type de détériorations graves, et
- La diminution des risques imprévus.

Contrairement à cela, l'inconvénient majeur d'une telle technique est qu'elle repose sur l'hypothèse d'un temps moyen de bon fonctionnement fixe, ce qui implique un taux de défaillances constant, alors qu'en réalité un équipement vieillit, ce qui se traduit par une diminution du temps moyen de son bon fonctionnement.

4.2.2 Maintenance Corrective

Comme l'indique son nom, la maintenance corrective [M.D. Kent et al, 2017] vise à corriger les défauts de fonctionnement de tout équipement ou ligne de production dès leur apparition si la panne qui perturbe le processus de production n'a pas été anticipée par d'autres types de maintenance, comme la maintenance préventive.

4.2.2.1 Définition et Objectifs

C'est une maintenance qui est effectuée après une défaillance vis-à-vis l'élimination d'une avarie ou d'une altération dans le fonctionnement d'un élément matériel, ce qui implique une indisponibilité immédiate ou à très brève échéance des matériels affectés.

4.2.2.2 Les types de la maintenance corrective

Suivant la nature des interventions, on distingue deux types de remise en état de fonctionnement :

- ✓ **La maintenance corrective palliative** : Ce type de maintenance intervient lorsqu'un problème important ou bloquant, a été identifié. Selon **AFNORFD X 60-000** qui décrit la maintenance corrective palliative comme une : « action de maintenance corrective destinée à permettre à un bien d'accomplir provisoirement tout ou partie d'une fonction requise appelée couramment dépannage ».
- ✓ **La maintenance corrective curative** : La maintenance curative vise à réparer de façon définitive les sources de pannes, elle est prioritaire car les arrêts de production provoquent des lourdes pertes pour les entreprises.
- ✓ **La maintenance corrective immédiate** : La maintenance corrective immédiate consiste à réparer totalement l'équipement défectueux dès la première intervention. Le technicien procédera d'abord à un dépannage, puis à un remplacement ou un recalibrage qui permettra à l'équipement en question de fonctionner totalement sur le long terme.
- ✓ **La maintenance corrective différée** : Le technicien procédera d'abord à un dépannage superficiel, qui permettra à l'ordinateur (par exemple) de fonctionner provisoirement. Puis, le technicien devra intervenir de nouveau à une date ultérieure, afin de terminer sa maintenance corrective. L'appareil sera alors en total état de marche sur le long terme.

4.2.2.3 Opération de la maintenance corrective

Après l'apparition d'une défaillance, le technicien ou bien le responsable de la tâche de réparation doit mettre en œuvre un certain nombre d'opérations dont les définitions sont données ci-dessous. Ces opérations s'effectuent par étapes et dans l'ordre suivant :

Operations	Définitions
Le test	La comparaison des mesures avec la référence ou bien les valeurs types.
La détection	L'action de déceler l'apparition d'une défaillance.
La localisation	La localisation et la recherche d'une manière précisée éléments par lesquels la défaillance se manifeste.
Le diagnostic	L'identification et l'analyse des causes de défaillance.
Le dépannage	La réparation ou la remise en état (avec ou sans modification).
Le Contrôle	La vérification du bon fonctionnement du système après l'intervention, suivie nécessairement d'un jugement.
L'amélioration éventuelle	Dans le but d'éviter la réapparition de la panne dans le futur.
L'historique	Mettre à jour la base de connaissance pour une exploitation ultérieure.

Tableau2 : Opération de la maintenance corrective.

4.2.2.4 Avantages et Inconvénients de la maintenance corrective

La maintenance corrective donne de bons avantages pour un équipement défectueux. Elle élimine les avaries et les altérations qui empêchent le bon fonctionnement de quelques éléments grâce à la réparation, la restauration de son état antérieur, et le remplacement des éléments défectueux. Par exemple, pour l'industrie, la maintenance corrective est idéale afin de détecter le défaut. Cette intervention remettra le pont roulant d'une machine de manière normale. Par exemple, Une réparation faite dans l'immédiat réduira la probabilité des dépannages urgents. L'adaptation de ce type de maintenance est un bon moyen plus simple et plus économique pour optimiser le pont roulant. Parmi les diverses adaptations plus courantes, la maintenance prendra le relais concernant le remplacement de certains composants.

Même si la maintenance corrective est bien utile, elle représente quand même un petit inconvénient. Premièrement, la maintenance corrective ne concerne que les installations avec des coûts minimes. Deuxièmes, elle ne peut donc pas être élargie dans d'autres domaines et reste figée sur une seule option ce qui explique le fait que 80 % des réparations concernent la maintenance préventive et la maintenance corrective ne prend que les 20 % qui reste.

4.2.3 Maintenance Améliorative

La maintenance améliorative est un ensemble des mesures techniques, administratives et de gestion, destinées à améliorer la sûreté de fonctionnement d'un équipement informatique sans changer sa fonction requise. C'est ainsi que l'amélioration se rapporte à des modifications de la conception d'origine dans le but d'augmenter la durée de vie des composants, de les standardiser, de réduire la consommation d'énergie, et d'améliorer la maintenabilité,etc.

II. Les outils logiciels pour la maintenance informatique

1. Introduction

Le besoin de communication entre les applications informatiques n'a jamais cessé de croître avec la popularisation du réseau internet et du web. L'architecture client/serveur a permis d'offrir divers services répartis entre des clients et des serveurs d'applications réparties (exemple: serveurs d'informations, messagerie, etc.)

A ses débuts, la technologie impliquait une programmation longue et difficile d'approche en utilisant des mécanismes de bas niveau tels que les sockets. Entre autres, elle imposait au concepteur de logiciels de mettre au point des protocoles de communication relativement complexes. Ces protocoles devaient contrôler la transmission des données et effectuer les vérifications d'usage et le codage/décodage des données échangées. Le programmeur devait donc réserver une bonne part de son temps à la conception du protocole, affectant le temps consacré au processus de conception du logiciel lui-même.

Ensuite des technologies de plus haut niveau telles que RPC (**RemoteProcedureCall**) ont été développés, qui permet de séparer une application en deux modules (le serveur et le client) de fonctionnalités différentes et de les distribuer sur des machines différentes sous forme de procédures qui peuvent être exécutées à distance à travers un réseau. Cette technologie permet aux applications d'exécuter des procédures appartenant à des applications distantes et ce, de façon transparente. Avec cette technologie, le programmeur n'a plus qu'à développer des protocoles gérant les méthodes de transfert, de vérifications et de codage/décodage. Ces opérations sont entièrement prises en charge par RPC.

L'évolution des langages de programmation a amené de nouveaux outils aidant à la conception d'applications. Par exemple, la venue du langage orienté objet a facilité l'abstraction des problèmes à résoudre en fonction des données du problème lui-même (par l'utilisation de classes et d'instances d'objets). Elle a aussi permis l'essor de nouvelles technologies de distribution des applications. Les concepts d'héritage, d'encapsulation et de polymorphisme sont mis à profit avec langage orienté objet Java **RMI (RemoteMethod Invocation)** qui est un exemple de technologie orientée objet permettant de distribuer une application. Grâce à ce paradigme (de manière semblable à **RPC**), un client peut invoquer des méthodes d'un objet distant.

Les applications distribuées peuvent prendre plusieurs formes. Le système le plus simple est constitué d'un client et d'un serveur échangeant des informations. Le serveur attend et répond aux requêtes des clients, et les clients transmettent leurs requêtes. Les informations peuvent être échangées de façon bidirectionnelle entre les deux modules.

La venue du Web a permis de généraliser ces outils pour des applications hétérogènes en permettant de les modifier et de les incrémenter tout assurant l'accès aux ressources distribuées sur l'internet. Le web a été utilisé en tant que middleware permettant d'accéder à des données et à des applications. La disponibilité des serveurs web et des clients web (les navigateurs) a permis l'accessibilité de plus en plus grande à ces données et services.

La tendance qui a marqué cette évolution a poussé les concepteurs à utiliser le web pour stocker des services sous forme de composants accessibles à travers le réseau, appelées **Services web**. Ces derniers comportent plusieurs composants.

- ✓ Un protocole d'échange de messages de requêtes et de réponses appelé **SOAP (Simple Object Access Protocol)**.
- ✓ Un langage de description des contrats des services appelé **WSDL (Web Service Description Language)**.
- ✓ Un référentiel de description des services offerts appelé **UDDI (Universal Description Discovery and Integration)**.

2. Les technologies des services WEB

Les services web ont émergé comme un mécanisme puissant assurant l'interaction entre des fournisseurs de services et des consommateurs sur différentes plates-formes technologiques sur le web. Ils fournissent des mécanismes flexibles, extensibles basés sur le langage XML (pour la représentation des données et des messages) ainsi que les protocoles standard de l'internet tels que HTTP et FTP. Ces mécanismes facilitent le développement des systèmes en utilisant une architecture dite **Architecture Orientée Services (Service Oriented Architecture ou SOA)**.

L'évolution des logiciels applicatifs des architectures client-serveur s'orientée plus en plus vers des architectures avec services Web. L'architecture des services web comportent des éléments servant à décrire les services offerts, à découvrir ces services et à véhiculer des messages pour les invoquer. Ces éléments sont :

2.1 Le protocole SOAP (Simple Object Access Protocol)

Est un protocole servant à échanger des données XML , il définit les éléments qui permettent d'effectuer des appels de type RPC en se basant sur HTTP comme protocole de transport.

2.1.1 Structure d'un message SOAP

Un message SOAP est un document XML qui suit une structure définie dans un schéma XML. Il est composé principalement d'une enveloppe (**Envelope**) qui contient un en-tête (**Header**) optionnel et un corps (**Body**) obligatoire comme le montre la figure3 suivante :

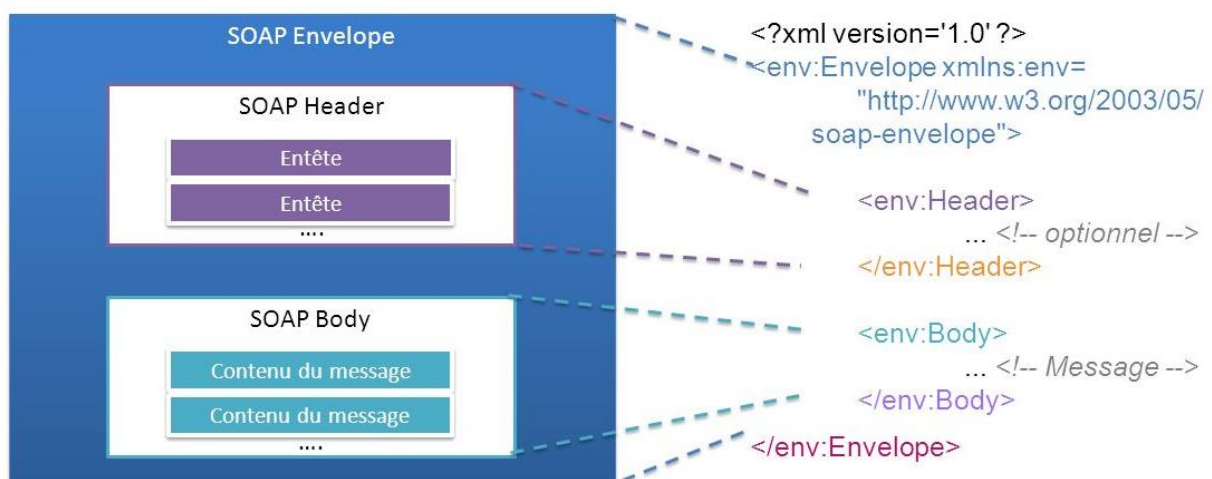


Figure3 : Structure d'un message SOAP.

L'élément Enveloppe, racine d'un message SOAP, contient un attribut obligatoire XML qui fait référence au schéma XML décrivant la structure d'un message SOAP. Il contient aussi un attribut optionnel encoding Style qui définit le style d'encodage du message SOAP.

L'élément Header permet de combiner d'autres technologies au protocole SOAP, telles que l'ajout des entrées pour le routage des messages SOAP, des informations relatives à la sécurité, ainsi que d'autres types d'extensions. L'élément Body contient la requête SOAP sous format XML. Les messages SOAP sont généralement utilisés pour acheminer des requêtes SOAP d'une application à une autre, et les réponses le cas échéant.

2.1.2 Patrons d'échange de messages SOAP

Le protocole SOAP supporte plusieurs patrons d'échange de messages SOAP:

- ✓ **Message de type requête-Réponse:** Le client envoie une requête SOAP au serveur et attend une réponse. Il est possible que le client reçoive plusieurs réponses pour une seule requête.
- ✓ **Message de sens unique (One-way) :** Le client envoie une requête SOAP au serveur et n'attend pas de réponse.
- ✓ **Message de type notification:** Le serveur envoie un message SOAP au client pour notification.
- ✓ **Message sollicité (serveur/Client) :** Le serveur envoie un message SOAP au client et reçoit une réponse de ce dernier.

2.2 Le langage de description WSDL (Web Services Description Language)

Est un vocabulaire pour définir ce que le service web fait, où il réside et comment l'invoquer. Il s'agit d'un langage JDL (**Job Description Language**). Celui-ci permet aux fournisseurs d'un service de décrire les données et les commandes que ce service accepte. Ces descriptions sont stockées dans un registre que les clients potentiels peuvent consulter pour découvrir ces services et ces descriptions.

2.2.1 Structure d'un document WSDL

Un document WSDL [**Web Service, 2021**] est un fichier XML, qui se conforme à un schéma XML. L'élément définitions, racine de ce document, englobe un ensemble d'éléments regroupés principalement en trois parties (Voir Figure4):

- ✓ Les éléments types, message, portType et opération définissent les opérations offertes par un service Web et les entrées-sorties de chacune de ces opérations.
- ✓ L'élément binding définit les protocoles de communication et de transport internet utilisés pour invoquer les opérations définies dans l'élément portType.
- ✓ Les éléments service et port définissent les points d'accès au service.

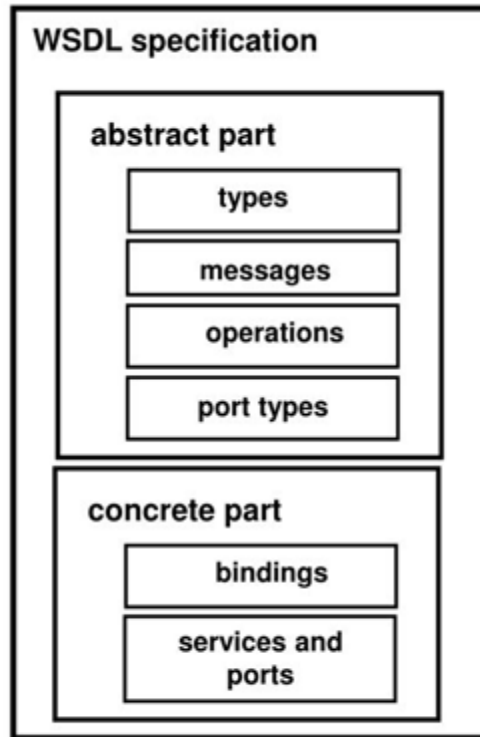


Figure4 : Structure d'un document WSDL[Web Service, 2021].

2.3 Le registre UDDI (UniversalDescription, Discovery and Integration)

Est un registre auquel on peut accéder pour trouver les descriptions des services ainsi quelques caractéristiques d'accès. Lorsqu'un fournisseur veut rendre un service accessible, il publie dans UDDI avec une commande de type *publish*. Les clients peuvent alors effectuer des recherches de services (avec une commande de type *find*) pour les utiliser dans leurs applications (avec une commande de type *bind*). L'accès à UDDI se fait également au moyen du protocole SOAP.

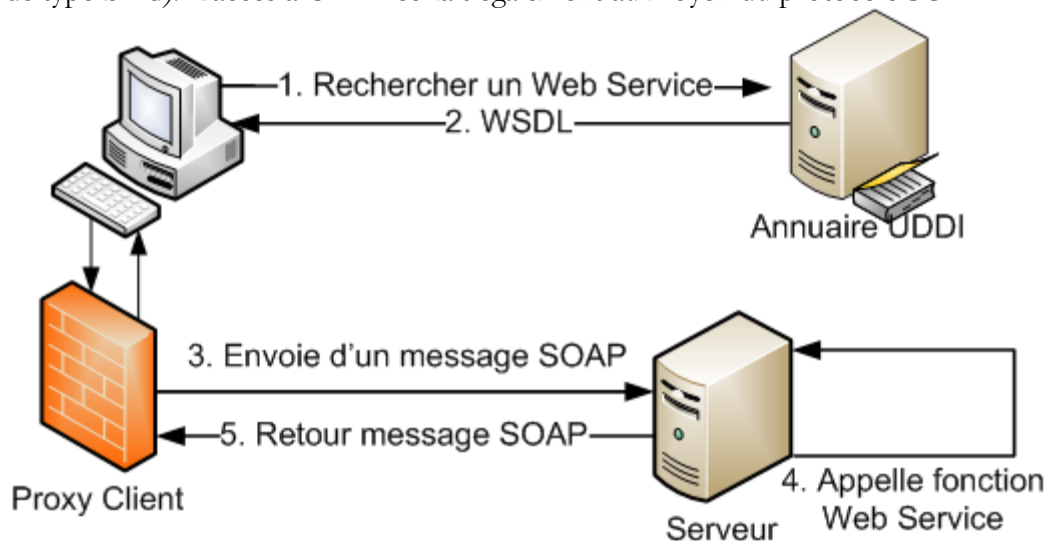


Figure5 : Interaction de l'annuaire UDDI.

2.3.1 Publication d'un service Web dans un registre UDDI

Pour illustrer quelques modèles de requêtes de publication, un scénario qui décrit la publication des informations d'une entreprise et d'un service qu'elle offre est donnée dans cet sous section. Voici les étapes de publication d'un service Web dans un UDDI.

- ✓ Une requête de publication nécessite, une clé d'authentification. Cette clé peut être récupérée par une requête de type **get_authToken**.
- ✓ Si le client détient les privilèges de publication sur le registre UDDI, il reçoit une entité de type **authInfo** qui contient une clé d'authentification qui va servir aux requêtes de publication.
- ✓ Une fois que la clé d'authentification est reçue, le client prépare une requête de type **save_business** qui sert à enregistrer les informations de l'entreprise et l'envoie au registre UDDI.
- ✓ Si tout va bien, le client reçoit, en réponse, une entité de type **businessDetail** qui contient les informations de l'entreprise enregistrée et une clé d'identification unique de l'entreprise. Cette clé servira par la suite pour enregistrer les services Web offerts par cette entreprise.
- ✓ Une fois que les détails de l'entreprise sont enregistrés, on passe à l'étape de publication de service Web qu'elle offre. Alors, le client prépare une requête de type **save_service** qui contient les détails d'un service Web.
- ✓ Le client reçoit, en réponse, une entité de type **serviceDetail** qui contient les informations du service enregistré, et une clé d'identification unique du service Web.
- ✓ Le client reçoit, en réponse, une entité de type **tModelDetail**, qui contient les informations du tModel enregistré, et une clé d'identification unique pour le tModel. L'enregistrement d'un tModel n'est pas nécessaire dans le cas où les détails techniques du service Web font référence à un tModel existant.
- ✓ Une fois que le tModel est créé, le client envoie une requête de type **savebinding** au registre. Cette requête enregistre l'adresse du service Web et une référence vers le tModel qui pointe sur l'adresse du document WSDL du service.
- ✓ Il reçoit, en réponse, une entité de type **bindingDetail** qui contient les informations techniques du service Web.

2.3.2 Recherche d'un service Web dans un registre UDDI

La spécification UDDI définit une dizaine de requêtes de recherche sur un registre. Ces requêtes permettent aux utilisateurs de chercher les informations d'une entreprise ou d'un service Web selon plusieurs critères et sur plusieurs niveaux.

- ✓ Le client envoie une requête **FindBusiness** qui cherche toutes les entreprises ayant le nom X. Il est possible de trouver plus d'une entreprise ayant le même nom sur un registre UDDI.
- ✓ Il reçoit en réponse une entité de type **BusinessList** qui contient la liste des entreprises ayant le nom X et la liste des services Web qu'elles offrent.
- ✓ Le client choisit le service Web qu'il veut consulter. il envoie une requête de type **GetServiceDetail** qui contient une clé unique. Chaque service Web enregistré au niveau du registre UDDI possède une clé unique.

- ✓ Il reçoit en réponse une entité de type **ServiceDetail** qui contient en détail des informations sur le service Web, comment y accéder et comment l'utiliser.
- ✓ Le document WSDL de ce service Web peut être récupéré en envoyant une requête de type **GetModelDetail**. Cette requête doit contenir une clé unique du **tModel** qui fait référence au document WSDL de ce service Web.
- ✓ Le client reçoit en réponse une entité de type **tModelDetail** qui contient une référence vers l'URL du document WSDL de ce service

2.4 Le langage XML (eXtensible Markup Language)

Est un langage de balisage extensible qui définit un ensemble de règles permettant de structurer des données. Contrairement au langage **HTML**, XML est un langage ouvert. il permet de créer ses propres balises selon le besoin. Ces balises n'ont pas de signification pour le langage XML, mais elles ont un sens pour les applications qui les utilisent.

2.4.1 Schéma d'un fichier XML

La structure de base d'un fichier XML est assez simple, et très précise. Rappelons-nous que le XML est une évolution du HTML, il est donc logique d'y retrouver le même genre d'éléments. Autrement, un schéma XML est un standard qui décrit la structure d'un document XML. Il définit les éléments et les attributs qui peuvent apparaître dans un document XML ainsi que l'ordre dans lesquels ils doivent apparaître. Contrairement à un DTD, un schéma XML est facile à lire. Il utilise le langage XML pour définir des contraintes de typage et les relations entre les éléments d'un document XML (Voir Figure6).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://schemas/temperature"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas/temperature"
  elementFormDefault="qualified" >

  <xsd:element name="temperature" type="Temperature"/>

  <xsd:complexType name="Temperature">
    <xsd:sequence>
      <xsd:element name="min" type="DegreeRange"/>
      <xsd:element name="max" type="DegreeRange"/>
      <xsd:element name="forecast-low" type="DegreeRange"/>
      <xsd:element name="forecast-high" type="DegreeRange"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="DegreeRange" >
    <xsd:restriction base="xsd:decimal">
      <xsd:minInclusive value="-70"/>
      <xsd:maxInclusive value="80"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

Figure6 : Schéma Document XML.

3. Fonctionnement général d'un service WEB

Tous les éléments de l'architecture des services web utilisent XML pour représenter les descriptions avec WSDL, les stocker dans l'UDDI et échanger les messages de requêtes et réponses avec SOAP. Le fournisseur doit d'abord réaliser ses services Web et produire leurs descriptions sous format WSDL comme indiqué dans les étapes ci-dessous :

1. Le fournisseur publie ses services dans l'annuaire UDDI en envoyant les messages encapsulés dans une enveloppe SOAP au registre UDDI via un protocole de transport (ex. HTTP). Les informations fournies par le fournisseur contiennent les méthodes d'invocation, les paramètres, la localisation du service ainsi le format de la réponse.
2. Le client recherche les informations sur les services et dans le registre UDDI.
3. L'annuaire UDDI retourne la réponse (qui est un message WSDL) au client.
4. Dès que le client reçoit la réponse, il interroge le fournisseur pour pouvoir accéder au service. La demande se fait via un message de SOAP.
5. Le fournisseur répond en donnant le format que le client doit utiliser. Pour effectuer l'appel.
6. Le client envoie sa requête d'appel sous forme d'un message SOAP au fournisseur pour obtenir le service demandé.
7. Le serveur envoie la réponse à cet appel.

En vue de faciliter l'interopérabilité et l'accès aux services web, ceux-ci se basent sur des standards et des normes bien définies : langages de description, protocoles de transport, registre, etc. Ces protocoles et ces langages sont basés sur XML. La figure3 illustre le modèle de fonctionnement des services Web. Ce modèle présente les grandes phases de la mise en œuvre d'un service Web(Voir Figure7).

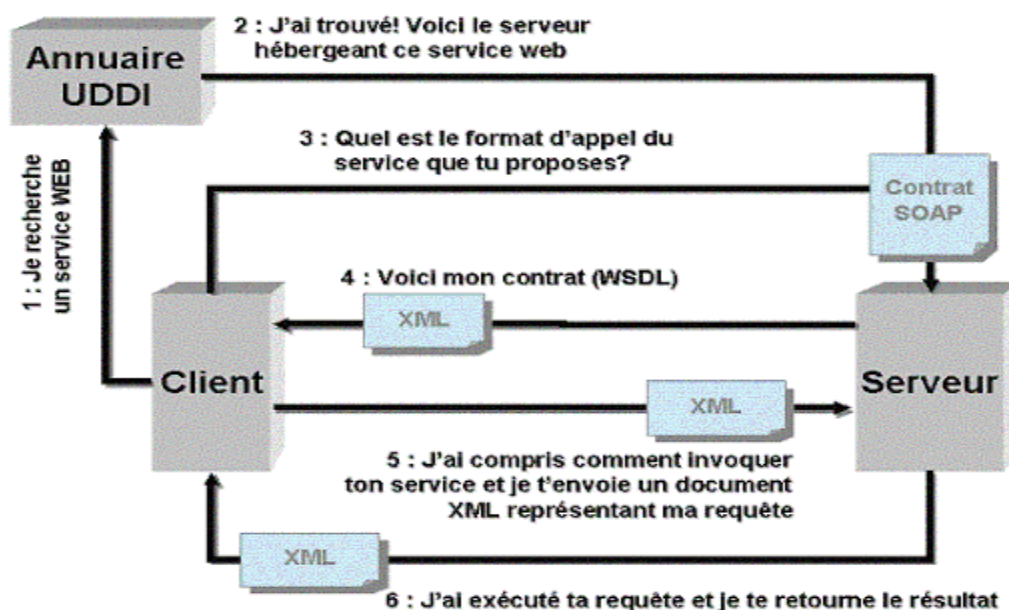


Figure7 : Fonctionnement général d'un service WEB.

III. Conclusion

La tâche de maintenance informatique est une fonction complexe qui, selon le type de processus, peut être déterminante pour la réussite d'une entreprise. Dans la première partie du chapitre, Nous avons essayé de donner un aperçu global du système de maintenance informatique en le décomposant en sous-fonctions et en précisant leur signification. Enfin, il est bon de souligner que le cycle de vie de la maintenance ne devrait pas débiter après la défaillance d'un service ou d'un matériel mais en réalité, bien avant par une participation active des responsables de la maintenance.

De plus, l'élément clé d'une architecture SOA est son service web. Ces services ne sont qu'un moyen de mettre en place une architecture orientée services et d'atteindre ses objectifs. Ils sont considérés comme étant l'évolution naturelle du web. En effet, Les Services Web ont apporté une solution aux problèmes d'échanges de données et d'intégration d'applications entre entreprises. Encore faut-il que les entreprises puissent identifier ou sélectionner des partenaires d'affaires ou bien elle créer leur propre cellule de création de service WEB pour un besoin particulier.

Dans le chapitre suivant, nous poursuivons notre état de l'art en mettant en évidence les différentes méthodes collaboratives de prise de décision.

Chapitre2 :

Vers une approche collaborative multicritère de décision

Plan du Chapitre2

1. Introduction
2. Système multi agent
 - 2.1 Introduction
 - 2.2 Agent et système multi agent
 - 2.3 Type d'agent
 - 2.4 La conception des SMA
 - 2.5 La communication entre agent
3. Aide à la décision multi critères
 - 3.1. Types de problèmes de décision
 - 3.2. La méthode d'analyse multicritère
 - 3.3. Les bases méthodologiques
4. La décision collaborative
 - 4.1. La notion d'aide à la décision
 - 4.2. Définition de la collaboration
 - 4.3. Processus de décision collaborative
 - 4.4. Outils de collaboration
 - 4.5. Caractéristique des systèmes d'aide a la décision
5. Opérateurs d'agrégations
 - 5.1. Définition d'un opérateur d'agrégation
 - 5.2. Types d'opérateurs d'agrégation
 - 5.3. Méthode d'agrégation
6. Conclusion

1. Introduction

L'aide à la décision est un processus qui utilise un ensemble d'informations disponibles à un instant donné, afin de formuler un problème et aboutir à une décision sur un objet précis. A nos jours, les décisions prises par les organisations sont complexe. Elle demande des points de vue et des compétences divers qui ne peuvent pas être maîtrisé par un seul individu car les modèles de décisions traditionnels adaptés au cas d'un décideur unique ne reflètent pas fidèlement la réalité.

Dans ce chapitre, nous commençons par présenter le principe des systèmes multi agents. Dans un seconde temps nous abordons la notion d'aide à la décision multicritères. Ensuite nous citons les différentes notions liées à la décision collaborative et à la fin de ce chapitre, les opérateurs d'agrégations ont été discutés.

2. Les systèmes multi agents

2.1 Introduction

Le paradigme multi-agents a été largement utilisé pour modéliser et concevoir des systèmes de maintenance. Leurs caractéristiques, que nous allons détailler par la suite, font qu'ils sont appropriés pour la modélisation de tels systèmes complexes. Les systèmes multi-agents procèdent donc à une décomposition de l'application en plusieurs entités autonomes et coopératives. Cette nouvelle décomposition permet d'accélérer les retours de l'application à l'utilisateur.

2.2 Agent et systèmes multi agent

Depuis trois décennies, les systèmes multi-agents constituent une extension et un prolongement de la notion d'objet. En effet, la modélisation basée sur agents permet de modéliser des systèmes complexes, distribués, ouverts, adaptables et autonomes. Les systèmes multi-agents ont été largement utilisés par les chercheurs et les industriels. Il est nécessaire alors d'introduire les définitions et les notions générales d'agent. Cependant les définitions diffèrent dans la littérature selon les auteurs et leur discipline. Nous utiliserons par la suite, la définition la plus répandue dans la communauté française.

Le principal intérêt des SMA (Système Multi Agent) est qu'ils permettent de distribuer des agents, entités communicantes, autonomes, réactives et dotées de compétences. Pour réaliser un SMA selon ces critères, il faut doter chaque agent, dit « cognitif », des trois propriétés suivantes : indépendance, communication et intelligence (expertise, savoir-faire). Pour modéliser de tels agents il nous faut définir leur architecture (fonctions et interactions cognitives), ainsi que la structuration des connaissances nécessaires pour leurs différentes activités.

Plusieurs définitions apparaissent autour de la définition des agents et le système multi agent qui les représente, Selon [J.Farber, 1995] un agent est une entité informatique, réelle ou abstraite, indépendante à durée de vie limitée, capable d'agir sur elle-même et son environnement.

Selon [H.Bond et L.Gasser, 1988] « une société d'agent est constituée de trois éléments, un ensemble d'agents, un ensemble de tache à réaliser et un ensemble d'objets associé à l'environnement ». Un agent peut prendre la responsabilité d'effectuer une tache s'il en a la capacité. Il prend alors un rôle dans le groupe. La réalisation d'une tache suppose la manipulation d'objet de l'environnement.

Les systèmes multi agent sont généralement classés en deux principales familles : les systèmes cognitifs et les systèmes réactifs. Les systèmes cognitifs se rapproche le plus du modèle de société d'expert.

2.3 Type d'agent

Nous distinguons deux familles d'agents dans la littérature : cognitif et réactif. Un agent cognitif a une représentation explicite de croyances, de buts, de modèles des autres agents, etc., contrairement à un agent réactif.

2.3.1 Agent Cognitifs

Les agents cognitifs sont généralement intentionnels. En effet, ils disposent de plans explicites, ainsi que d'une base de connaissances avec plusieurs informations liées à la gestion des interactions avec les autres agents et leurs domaines d'expertise, ce qui leur permet d'atteindre les buts qui leur ont été fixés. Dans ce contexte, les notions de coordination et de négociation pour résoudre les conflits émergent.

2.3.2 Agent Réactifs

Contrairement aux agents cognitifs, les agents réactifs ne sont pas "intelligents" si nous les considérons individuellement. Ils ne sont pas capables de réagir à des changements de leur environnement. Leur comportement est alors régi par leur relation à leur entourage. Cependant, les agents réactifs ont la capacité de résoudre des problèmes complexes en s'intéressant à la modélisation d'un système d'agent plutôt qu'à la modélisation de l'agent. Nous résumons les différences entre les agents cognitifs et réactifs dans le **Tableau 3** :

Agents cognitifs	Agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire locale
Agents complexes	Fonctionnement stimulus/action
Nombre d'agents réduit	Nombre d'agents élevé

Tableau3 : Comparaison entre Agent cognitif et Réactif.

2.3.3 Agent Hybrides

En générale, la déférence entre agent réactif et cognitif peut être expliquée par le compromis efficacité / complexité. La complexité des systèmes réactifs exige le développement des nouvelles théories dans le domaine de la coopération, de la communication et de la compréhension de nouveau phénomènes tel que l'émergence. Toutefois il est maintenant possible de concevoir des systèmes hétérogènes comportant les deux types de comportements (cognitif et réactif) : on parlera alors des agents hybrides.

2.4 La conception de SMA : décomposition Voyelle (A E I O)

Les méthodes de conception de systèmes ont pour objectif de définir un ensemble d'étapes et un ensemble de règles nécessaires pour rendre le processus de construction des systèmes plus compréhensible. Le cycle de vie d'un système est constitué de deux grandes phases de développement : l'analyse et la conception.

- ✓ **La phase d'analyse** : concerne l'élaboration d'une solution détaillée mais indépendante des moyens de réalisation. Elle comprend la description de tous les processus composant le fonctionnement du système, la définition des informations utilisées, et la spécification des tâches à effectuer.
- ✓ **La phase de conception** : a pour but de choisir, dans le domaine opérationnel, les modèles pour la réalisation. Elle définit les spécifications opérationnelles pour le fonctionnement du système et les moyens choisis pour la solution retenue.

Il existe de nombreuses méthodes de conception pour les systèmes multi-agents, dont la plupart sont centrées sur l'analyse des tâches des différents agents.

Dans ce travail, nous adopterons la décomposition **AEIO** (*Voyelle*) qui privilégie une description explicite des interactions et de l'environnement, ce qui est plus adapté pour un problème comme le nôtre.

2.4.1 Les agents(A)

L'axe A de l'analyse AEIO permet donc de regrouper les éléments nécessaires à la construction des agents. On définit ici les modèles des agents qui cohabitent dans le système. Il faut définir les

Capacités et les compétences attribuées à chaque agent.

2.4.2 L'environnement (E)

Dans cet axe, on définit l'environnement où les agents agissent et évoluent. L'environnement dépend toujours du domaine d'application du problème mais sont généralement spatialisés, c'est-à-dire dotés d'une métrique

2.4.3 Les interactions (I)

La communication est le principal mécanisme d'interactions d'un agent avec la communauté des agents en conformité avec la relation triangulaire (**Communication, Coopération, Compétition**). Nous insistons sur le fait que pour qualifier un agent d'intelligence il est essentiel de prouver ses capacités à communiquer dans un but individuel ou collectif. Pour communiquer entre eux (information ou dialogue), les agents expriment leurs intentions selon le langage KQML, dérivé de la théorie des actes de langages.

Dans cet axe, on définit le mode de communication entre deux ou plusieurs agents à travers un ensemble d'actions réciproques. La conséquence de telles actions exerce en retour une influence sur le comportement futur des agents. On distingue deux types de communication :

- ✓ **indirecte**: par le biais de l'environnement, où les agents effectuent les émissions d'information sous forme d'actions et les réceptions d'information sous forme de perception. Ce mécanisme utilise l'environnement comme véhicule de transmission et peut être employé par des agents réactifs ou hybrides.
- ✓ **directe**: l'émission et la réception d'informations sont faites directement par les agents cognitifs. La conversation est établie entre des agents qui prennent tour à tour le rôle d'émetteur et de récepteur.

2.4.4 L'organisation (O)

Le concept d'organisation est assez vaste. On cite ici la définition de [T.W.Malone, 1988] : « une organisation peut être définie comme une structure de coordination (ou de prise de décision) et de communication comprenant un ensemble d'acteurs afin de réaliser un but

commun ». Dans le cadre de l'analyse AEIO, une organisation est soit **statique** soit **dynamique**. Dans le cas où la structure est conçue par le concepteur du système et qu'elle n'évolue pas, l'organisation est dite statique ; mais si le système est ouvert, autrement dit, si il reçoit et élimine des agents, le type d'organisation doit être dynamique pour permettre la réorganisation de la société.

L'axe O de la méthode comprend les éléments permettant d'affecter les agents à des ensembles (Des structures organisationnelles) par la détermination des rôles des agents, des dépendances entre les agents ou de leur statut dans la société.

La Figure8 illustre la vision d'un SMA décomposée selon l'approche **Voyelle** décrite ci-dessus.

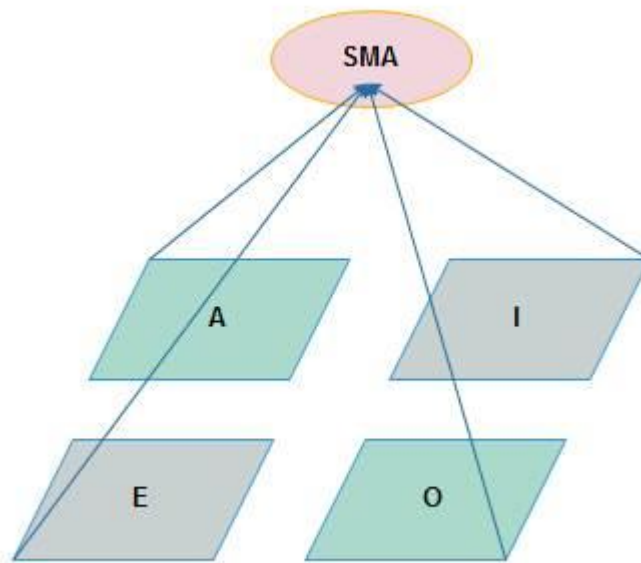


Figure8 : L'approche Voyelle d'un SMA

2.5 La communication entre agents

La communication désigne l'ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation d'un émetteur avec un ou plusieurs récepteurs, dans l'intention d'atteindre certains objectifs. Elle est considérée comme une forme d'action particulière, qui au lieu de s'appliquer à la transformation de l'environnement, agit sur les représentations mentales des agents (but, croyance...).

- a) **Communication par partage d'information** : Dans ce type de communication, la mémoire partagée est vue comme un tableau dont lequel les agents écrivent et trouvent des réponses partielles et des informations. C'est le premier modèle de communication qui est apparu au début des années 60 [Newell, 62]. Un dispositif de contrôle gère les conflits d'accès au tableau, les agents faisant les demandes d'accès de manière autonome. (Voir Figure9).

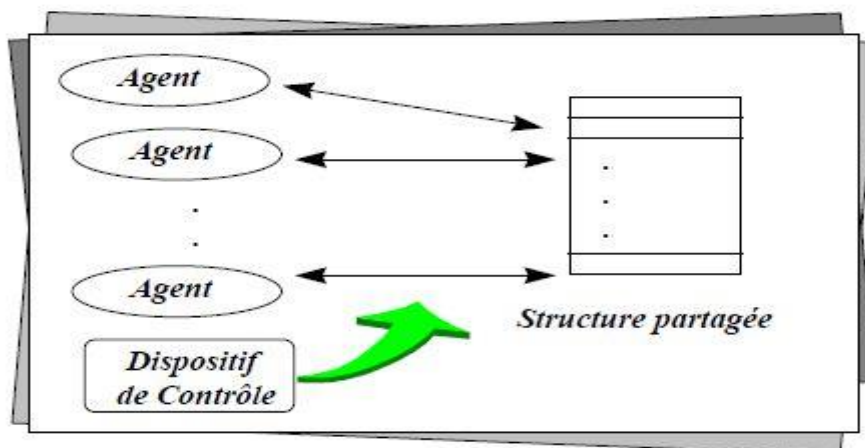


Figure 9 : Communication par partage d'information.

- b) **Communication par envoi de messages** : La communication par envoi de message se rapproche du premier modèle d'acteur proposé par [C.Hewitt, 77], qui définit un acteur comme une entité active et autonome qui a une vue partielle de l'univers. Cet acteur était décrit au moyen de deux éléments : une **accointance** et un **comportement**, la première correspondant à l'acteur connu d'un autre acteur et le deuxième est décrit par un script qui contient les différentes méthodes que l'acteur peut accomplir en réponse aux messages qu'il reçoit (Voir Figure10).

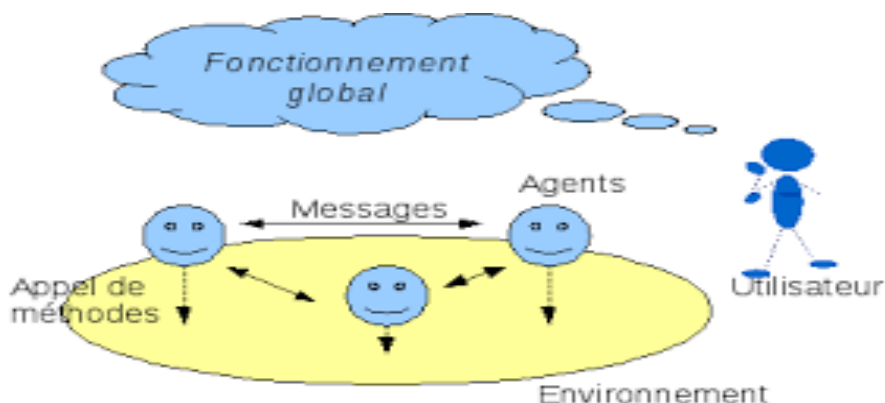


Figure10 : Communication par envoi de message.

Protocole de communication : Pour communiquer, les agents doivent utiliser un protocole qui leur permet de structurer et d'assurer la continuité des communications et des échanges entre un début et une fin. La nature de ces messages dépendant évidemment des capacités des agents à émettre des informations et à les traiter. Chacun émet des messages commençant par un mot de début et finissant de même, ces messages étant des ordres ou des informations, des requêtes, des promesses ou des menaces. Les réponses sont des messages porteurs des d'informations.

3 Aide à la décision multi critères

Les méthodes d'aide à la décision multicritères (MCDM) furent introduites dans les années 1970 en réponse à un besoin d'aide dans la prise de décision complexe. Plusieurs écoles ont émergées dans le domaine et elles pourraient se diviser en deux groupes : les méthodes d'aide à la décision multi-objectifs (MODM) et les méthodes d'aide à la décision multi attributs (MADM) [Pohekar et Ramachandran, 2004]. Chacune de ces catégories regroupent plusieurs types de

méthodologies qui ont chacune leurs avantages et leurs inconvénients en fonction du domaine d'application. Les points communs entre toutes ces méthodes sont qu'elles présentent des critères en conflit, des unités incomparables et de la difficulté dans le choix des alternatives.

L'aide à la décision multicritère se présente comme une alternative aux méthodes d'optimisation classiques basées sur la définition d'une fonction unique, souvent exprimée en terme économique (monétaire) et qui reflète la prise en compte de plusieurs critères. L'intérêt des méthodes multicritères est de considérer un ensemble de critères de différentes nature (exprimés en unité différentes), sans nécessairement les transformer en critères économiques, ni en une fonction unique. Il ne s'agit pas de rechercher un optimum, mais une solution compromis qui peut prendre diverses formes : choix, affectation ou classement.

3.1 Types de problèmes de décision

Dans le cadre de la décision multicritère, l'objet de la décision est formé par un ensemble d'actions ou alternatives. Pour [B.Roy, 1996] les problèmes réels peuvent être formulés à l'aide des méthodes d'analyse multicritère, selon trois formulations de bases : problématique de choix, notée P_α , la problématique de tri ou d'affectation notée P_β et la problématique de rangement noté P_γ (Voir Tableau4).

Problématique	Objectif	Résultat
P_α	Eclairer la décision par le choix d'un sous-ensemble aussi restreint que possible en vue d'un choix final d'une seule action. (optimums et satisfecums)	Un choix ou une procédure de sélection.
P_β	Eclairer la décision par un tri résultant d'une affectation de chaque action à une catégorie, les catégories étant définies a priori en fonction des normes ayant trait à la suite à donner aux actions qu'elles sont destinées à recevoir	Un tri ou une procédure d'affectation
P_γ	Eclairer la décision par un rangement obtenu en regroupant tout ou partie (les « plus satisfaisantes ») des actions en classes d'équivalence, ces classes étant ordonnées, de façon complète ou partielle, conformément aux préférences	Un rangement ou procédure de classement
P_δ	Eclairer la décision par une description, dans un langage approprié, des actions et de leurs conséquences	Une description ou une procédure cognitive

Tableau4 : Identification de type de problématique [B.Roy, 1996].

3.2 Les méthodes d'analyse multicritères

Les méthodes multicritères sont généralement classées comme discrète ou continue, selon le domaine des alternatives. La première traite un nombre, généralement limité, et pré-spécifié d'alternatives. La dernière traite des valeurs de décision variables déterminées dans un domaine continu ou entier d'un nombre de choix très grand ou infini. Différentes méthodes d'analyse multicritères sont disponibles dans la littérature [Figueira et al, 2005] [Mousseau, 2009].

- ✓ **Méthodes discrètes** : La condition majeure de la plupart des méthodes discrètes est l'élaboration d'une table de performance contenant l'évaluation ou le score des critères d'un ensemble d'alternatives sur la base d'un ensemble de critères. Par la suite une procédure d'agrégation est nécessaire qui prend en compte les préférences du décideur sous forme de poids en générale.
- ✓ **Méthodes continue** : Le point de départ de la plupart des procédés continus est un ensemble de fonctions contraintes qui contiennent les inégalités qui reflètent les restrictions naturelles ou artificielles sur les valeurs d'entrées. Dans cette méthode, les préférences du décideur prennent généralement la forme de poids qui sont affectés à différentes fonctions objectives. Le décideur doit aussi indiquer pour chaque fonction objective sa direction d'optimisation.

3.3 Les bases méthodologiques

En toute généralité, lorsqu'on pose un problème multicritère, il s'agit d'en trouver la solution la plus adéquate, compte tenu d'un certain ensemble de critères, cette solution pouvant prendre diverses formes (choix, affectation, classement). On peut alors opérer en 4 grands étapes [M.Abdelhadi,2019] :

- a) **Dresser la liste des actions potentielles** : D'après [B.Roy 1985], « une action 'a' est la représentation d'une éventuelle contribution à la décision globale, susceptible, eu égard à l'état d'avancement du processus de décision, d'être envisagée de façon autonome et de servir de point d'application à l'aide à la décision ».
- b) **Dresser la liste des critères à prendre en considération** : La construction des critères est une étape délicate qui nécessite une compréhension du problème posé et une interaction avec les acteurs impliqués dans la prise de décision. Dans cette étape il s'agit d'identifier les enjeux et la nature des conséquences possibles sur l'objet de la décision, c'est à dire les actions considérées. Elle permet de comprendre chaque action comme un vecteur à n dimensions dans l'espace des critères.
- c) **Établir le tableau des performances** : Ce tableau est constitué, en lignes, des actions de A, et en colonnes, des critères de F. Cette représentation permet de comprendre chaque action comme un vecteur à n dimensions dans l'espace des critères. Les valeurs qui remplissent ce tableau peuvent être des rangs, donc dépourvus de toute signification cardinale, d'où l'appellation de performance. Pour

bien faire, chaque colonne contiendra aussi, comme information complémentaire, les fonctions seuils et les éventuels “poids”.

- d) **Agréger les performances** : Il s’agit ici d’établir un modèle des préférences globales, c’est-à-dire une représentation formalisée de telles préférences relativement à un ensemble A d’actions potentielles, que l’homme d’étude juge appropriée au problème d’aide à la décision.

4. La décision collaborative

4.1 La notion d’aide à la décision

D’après [B.ROY, 1985]: « **L’aide à la décision** est l’activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans le processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, ou simplement à favoriser un comportement de nature à accroître la cohérence entre l’évolution du processus d’une part, les objectifs et le système de valeurs au service des quels cet intervenant se trouve placé d’autre part».

D’après [A.Nafi , C.Werey ,2010] L’aide à la décision est donc un processus qui utilise un ensemble d’informations disponibles à un instant donné, afin de formuler un problème et aboutir à une décision sur un objet précis.

4.2 Définition de la collaboration

Un environnement collaboratif est vu comme un environnement construit à partir d’une combinaison de différentes technologies, dans lequel deux ou plusieurs participants peuvent communiquer, coordonner et collaborer pour réaliser un objectif commun. Autrement dit, la collaboration ne relève pas d’une répartition a priori des rôles. La collaboration s’entend en fait par une situation de travail collectif où les tâches et les buts sont communs.

D’après [T.Kvan, 2000] :« la collaboration est la résolution de problèmes communs, elle signifie travailler avec d’autres participants ayant des buts partagés. L’équipe tente de trouver des solutions satisfaisant chacun des points de vue des participants ».

D’après [Kerhoas, 2006] : « A la fin de la réalisation d’un travail collaboratif, la distinction entre les tâches accomplies par l’un ou l’autre des collaborateurs est quasiment impossible ».

4.3 Processus de décision collaborative

La notion de décision collaborative n’est pas nouvelle en soi mais a pris une nouvelle dimension avec l’essor des outils informatiques et internet qui offre aux organisations des possibilités nouvelles en matière d’organisation de travail. Cette notion est donc aujourd’hui étroitement liée aux technologies de l’information et de la communication (TIC).

Le processus de décision collaborative est très important afin de prendre la décision la plus adaptée, il faut procéder par étapes. En général, il existe quatre phases à suivre :

- ✓ **Phase de détermination de la situation réelle** : Cette étape consiste en la prise de conscience de la situation. L'apparition d'une situation particulière requiert une réaction de la part du décideur. Si un risque se présente, une solution doit être trouvée pour résoudre le problème. En revanche, une grande réflexion est nécessaire s'il s'agit plutôt d'une opportunité.
- ✓ **Phase d'instruction** : Cette étape concerne l'étude et l'analyse de cas. Il s'agit d'étudier des cas antérieurs (base des cas) et des solutions efficaces lors d'une situation similaire. Dans cette partie, les décideurs collectent également l'avis des collaborateurs et des experts pour faciliter la prise de décision. La collaboration se fait aussi via des entités autonomes appelées agent dans le but d'automatiser cette tâche.
- ✓ **Phase de sélection** : Dans cette partie, la première chose à faire est de réunir toutes les alternatives. Ensuite, il s'agit de déterminer les contraintes et les avantages de chaque solution. À ce stade, une concertation doit être réalisée pour peser le pour et le contre et choisir la solution idéale. L'automatisation de cette tâche est réalisée par l'incorporation des méthodes spécifiques appelée méthode d'agrégation qui sont décrit en détail dans la sous section 5 de ce chapitre.
- ✓ **Phase de mise en œuvre** : Logiquement, le processus se termine par le passage à l'action. Il s'agit de désigner les acteurs responsables et de lancer le dispositif adopté



Figure11 : Processus de prise de décision.

4.4 Outils de collaboration

A travers le paradigme de l'aide à la décision collaborative, Plusieurs expérimentations et outils permettent également d'explicitier ces nouveaux modes de prise de décision. D'après [N.Sad, 2017] ces outils sont regroupés en quatre catégories :

- ✓ **Outils de communication** : les emails, les chats, les messageries instantanées, visioconférence, etc.
- ✓ **Outils de partage d'application et de ressources** : Englobe les outils de partage d'applications, forum de discussion, etc.
- ✓ **Outils d'accès et de partage d'informations et de contenus** : Regroupe les Wikis, les Weblogs, les listes de diffusion, les cartographies de compétences, Office en ligne etc.

- ✓ **Outils de coordination et de synchronisation** : aident à la conduite de projets par exemple : agenda partagé, gestionnaire de planning, etc.

Les outils collaboratifs en entreprises se multiplient. Ils facilitent les échanges entre les collaborateurs, ils centralisent l'information et améliorent la productivité. Il en existe aujourd'hui de toutes les sortes à tel point qu'il devient parfois difficile de s'y retrouver, chaque plateforme ayant ses spécificités.

D'un point de vue entreprise, la majorité d'entre eux les classent en trois grands axes :

- ✓ **Les suites d'outils et logiciels de productivité en ligne** : de type Cloud Computing par exemple : Office 365, G Suite, etc.
- ✓ **Les réseaux sociaux d'entreprise, appelés aussi RSE** : qui permettent de centraliser l'information et qui traitent souvent de tous les sujets.
- ✓ **Les solutions collaboratives de type ChatOps** : Ils permettent de rassembler en une interface, les tâches à réaliser, les tâches effectuées, et les tâches en cours par exemple : Slack, GitHub, Javelo, etc.

4.5 Les caractéristiques des systèmes d'aide à la décision

Selon le contexte étudié, trois caractéristiques sont générés comme suit :

- ✓ **Système passif** : La plupart des systèmes d'aide à la décision fonctionnent en mode passif. L'utilisateur ou l'initiateur doit explicitement faire appel au système. Il décrit la situation par-rapport au contexte étudié et il attend un retour du système.
- ✓ **Système semi-actif** : Dans un système semi-actif, l'appel au système d'aide à la décision est effectué automatiquement. L'objectif est de fournir des informations et des règles de comportement et de connaissances générales acceptées par tous. Le système joue au minimum le rôle de garde-fou ou de "chien de garde" (watch-dog system).
- ✓ **Système actif** : Un système actif est un système qui, se déclenchant automatiquement, est capable de fournir un retour adapté au cas défini préalablement par l'utilisateur. Il peut prendre automatiquement des décisions, sans intervention de l'initiateur.

5. Opérateurs d'agrégations

5.1 Définition d'un opérateur d'agrégation

D'après [F.Soumana, 2018] un opérateur d'agrégation peut être considéré comme une fonction mathématique qui prend une entrée, appelée l'argument, et produit une sortie unique représentant le score global de l'entrée. Chaque entrée a une unique sortie associée à celui-ci. La fonction d'agrégation est généralement désignée par $y = f(x)$, où x est l'argument et y est la valeur agrégée.

L'argument x peut être un vecteur de taille n : $x = (x_1, x_2, \dots, x_n)$, et x_1, x_2, \dots, x_n sont appelés composants de x .

La fonction f peut être représentée de plusieurs façons :

- ✓ Comme une forme algébrique
- ✓ Comme un graphe.

5.2 Types d'opérateurs d'agrégation

De nombreuses méthodes et algorithmes ont été proposés pour décrire la notion d'agrégations. Nous donnons ci-après les définitions principales de quelques opérateurs et des définitions complémentaires sont disponibles en Annexe A.

- a) **La somme pondérée** : La somme pondérée est un opérateur d'agrégation souvent utilisé pour sa simplicité de mise en œuvre. Elle demande d'attribuer un poids à chaque critère en fonction de son importance dans un problème de décision et multipliant chacun d'eux par leur poids et en les additionnant. Elle est définie par la formule1 :

$$\sum_{i=1}^n w_i p_i \dots \dots \dots (1)$$

Avec w_i est le poids des critères tel que :

$$0 \leq w_i \leq 1, \sum_{i=1}^n w_i = 1$$

- b) **La somme pondérée ordonnée** : La somme pondérée ordonnée (**OrderedWeightedAveraging - OWA**) [R. YAGER,1988] est une classe d'opérateurs d'agrégation qui détermine le poids du critère en fonction de son degré de satisfaction à la différence de la somme pondérée où le poids dépend de la nature du critère. L'idée derrière ces opérateurs est de récupérer pour chaque vecteur d'alternatives le vecteurs d'utilité de ces alternatives, puis de le trier en ordre décroissant. Ensuite, on fait une somme pondérée de ces nouveaux vecteurs par des coefficients prédéfinis. Elle définit avec la même fonction de la précédente méthode.
- c) **L'intégrale de Choquet** : Les opérateurs d'agrégation tels que la somme pondérée ou OWA ne sont pas capables de modéliser les interactions entre les critères. Hors, il est très rare qu'il n'y ait pas de synergie entre les critères. Afin d'obtenir une représentation plus proche de la réalité, ces interactions doivent être prises en compte et pour cela, l'utilisation d'un vecteur poids n'est pas appropriée. Remplacer ce vecteur poids par une fonction non-

additive permet en plus de définir un poids pour chaque critère, de définir aussi un poids pour chaque sous-ensemble de critères.

- d) **L'intégrale de Sugeno :** Contrairement à l'intégrale de Choquet qui utilise des évaluations quantitatives, l'intégrale de Sugeno est utilisée pour des évaluations qualitatives. L'intégrale de Sugeno a été introduite dans [M. SUGENO,1974].
- e) **Weighted aggregated sum product assessment (WASPAS) :** WASPAS est l'une des méthodes MCDM les plus récemment développées. C'est une combinaison de deux approches MCDM bien connues, à savoir le modèle de somme pondérée (WSM) et le modèle de produit pondéré (WPM). Dans notre travail nous nous intéressons à cette méthode. La matrice suivante est utilisée pour démarrer la procédure de solution:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{t1} & x_{t2} & \dots & x_{tn} \end{bmatrix}$$

L'algorithme WASPAS est un algorithme de recherche optimal basé sur deux critères d'optimisation. Le premier critère est un critère de succès moyen pondéré similaire à la méthode WSM et le second critère est basé sur la méthode WPM. Sur la base des méthodes WSM et WPM, l'importance relative totale de la k-ième alternative est calculée comme suit (Voir formule2 et Formule3):

$$Q_k^{(1)} = \sum_{j=1}^n x_{kj} W_j \dots \dots \dots (2)$$

$$Q_k^{(2)} = \prod_{j=1}^n (x_{kj})^{w_j} \dots \dots \dots (3)$$

Un critère général d'agrégation pondérée des méthodes additives et multiplicatives peut alors être proposé comme suit(Formule4):

$$Q_k = 0.5Q_k^{(1)} + 0.5Q_k^{(2)} = 0.5 \sum_{j=1}^n x_{kj} W_j + 0.5 \prod_{j=1}^n (x_{kj})^{w_j} \dots \dots \dots (4)$$

Afin d'augmenter la précision du classement et l'efficacité du processus de prise de décision, une équation plus généralisée pour déterminer l'importance relative totale de la k-ième alternative peut être utilisée (Voir Formule5) :

$$Q_k = \lambda Q_k^{(1)} + (1 - \lambda) Q_k^{(2)} = \lambda \sum_{j=1}^n x_{kj} w_j + (1 - \lambda) \prod_{j=1}^n (x_{kj})^{w_j} \dots \dots \dots (5)$$

Enfin, les alternatives candidates peuvent être classées en fonction des valeurs de Q , c'est-à-dire que la meilleure alternative serait celle ayant la valeur la plus élevée. Lorsque la valeur de λ est 0, la méthode WASPAS coïncide avec WPM, tandis que pour $\lambda=1$, WASPAS correspond à WSM.

Dans notre travail de recherche nous s'intéressant à la méthode d'agrégation **WASPAS** la faite quelle est simple a implémenter et elle correspond mieux a notre cas d'agrégation.

D'autres méthodes d'agrégations sont apparues dans la littérature. Par exemple, Multi-objective optimization based on ratio analysis, Complex proportional assessment of alternatives, Technique for Order Preference by Similarity to Ideal Solution, Evaluation based on Distance from Average Solution.

5.3 Méthodes d'agrégations

Les trois premières étapes décrites au paragraphe précédent sont communes à toutes les méthodes et ne présentent que de faibles variations. La quatrième présente, une diversité nettement supérieure. Si les méthodes d'agrégation sont si nombreuses, c'est parce qu'il est dans la nature des choses qu'aucune méthode ne respecte la totalité des exigences qu'un utilisateur pourrait trouver "normales" dans l'idée du multicritère [Schärlig, 1985]. Il faut donc décider sur quelle exigence on va céder. Face à ce choix, on peut distinguer trois attitudes ou approches opérationnelles [B.Roy, 1985].

- ✓ **Méthodes d'agrégation complète** : L'énorme avantage de ce genre de méthode, il y a un classement clair des résultats (solutions), de plus mathématiquement parfaitement reproductibles et souvent définies avant, ce genre de méthodes bien que rassurantes, on produit un classement des solutions, ne sont pas toujours bien adaptées. La première attitude serait d'inclure toutes les performances dans ce qu'on appellerait en mathématique une fonction d'utilité ou d'agrégation [B.Roy, 1985], en leur attribuant d'éventuels poids. Cela suppose que tous les jugements sont commensurables alors qu'une des justifications de l'approche multicritère est le non commensurabilité de ces jugements. Par exemple : Somme pondérée, Maut, Uta, Ahp, ...
- ✓ **Méthode d'agrégation partielle** : une méthode qui permet de donner une direction un ensemble de solution sans pour autant se bloquer sur des questions annexes comme le poids d'un critère. Avec ce type de solution, vous perdrez en clarté du résultat, souvent une brochette de solutions et pas une solution qui est meilleure. Ici l'idée est de trouver les (la) solutions les (la) moins mauvaises. C'est souvent ce type de méthodes que vous utiliserez si vous devez trouver un compromis. Par exemple : Electre, Oreste, Qualiflex, ...)
- ✓ **Méthodes d'agrégation locale** : on part d'une solution qui pourrait être valable et on cherche par tâtonnement « autour ». C'est un peu long, beaucoup de dialogues avec le

(les) décideur, mais par cycles successifs, une solution se dessine. Parmi ces méthodes on cite : Plm, Stem, Cônes , ...

6 Conclusion

Les outils d'aides à la décision fournissent une méthode d'analyse de scénario multiple, ils aident à rendre la prise de décision robuste, consistante et reproductible. De plus les méthodes d'analyse multicritère ou, plus exactement, les méthodes d'aide multicritère à la décision sont des techniques assez récentes et en plein développement. Par leur manière d'intégrer tout type de critères, ces procédures semblent mieux permettre de se diriger vers un judicieux compromis plutôt qu'un optimum souvent désuet.

Dans ce chapitre, nous avons abordé les systèmes d'aide multicritère à la décision. Ensuite, un panorama des différents opérateurs d'agrégation qui auront été proposés au fil du temps a été présenté dans ce chapitre. Un bref aperçu des systèmes multi agents a été donnée qui reflète leur utilisation dans le domaine qui nous intéresse.

Notre objectif, par la présente étude, et de proposer un système multicritère d'aide a la décision et un ensemble d'agents dans le but d'automatiser la tâche préconiser.

A cet effet, nous aborderons dans le chapitre suivant les différentes stratégies et méthodes de traitement de données. Nous présenterons leurs intérêts, leur fonctionnements et on mettant l'accent sur les méthodes choisis pour élaborer notre plateforme.

Chapitre3

Techniques de traitement des données

Plan du chapitre3

1. Introduction
2. Les méthodes d'extraction des termes
 - 2.1** Les techniques d'extraction des termes
 - 2.1.1** Approches statistiques
 - 2.1.2** Approche linguistiques
 - 2.1.3** Approche par regroupement
 - 2.1.4** Approche à base de graphe
3. Mesure de similarité
 - 3.1 Introduction
 - 3.2 Type de similarité
 - 3.2.1 Similarité syntaxique
 - 3.2.2 Avantages et inconvénients
 - 3.2.3 Similarité sémantique
 - 3.2.3.1 Approche vectorielles
 - 3.2.3.2 Approche Ontologique
 - 3.2.3.3 Approche statistique
 - 3.2.3.4 Avantages et inconvénients
4. Conclusion

1. Introduction

Les techniques de traitement des données ont été profondément modifiées par l'analyse des données qui interviennent dans une phase préliminaire afin d'apprécier la qualité de l'information, et orienter la suite de traitement.

Ce chapitre commence par donner une vue d'ensemble sur les différentes méthodes d'extraction des termes ensuite présenter d'une manière détaillée les principales approches qui existe dans la littérature suivi par les techniques de mesure de similarité. Pour terminer avec un aperçu sur les travaux connexes.

2. Les méthodes d'extraction des termes

L'extraction de mots-clés (également appelée analyse des mots-clés) est une méthode d'analyse d'un texte qui consiste à extraire automatiquement les mots les plus importants. Cela permet de résumer le contenu d'un texte et d'identifier les principaux domaines qui sont abordés.

Dans notre contexte, quelle que soit la façon d'associer les termes-clés à un problème donnée, ils ont pour rôle de synthétiser au mieux le contenu de la requête d'entrée dans notre travail pour aider l'utilisateur à choisir parmi un ensemble de problèmes similaires ceux qui répondent le mieux à leur besoin.

2.1 Les techniques d'extraction des termes

L'extraction automatique de termes c'est une tâche d'identification des mots simples ou bien composés qui peut s'appuyer sur des approches de traitement du langage naturel qui est basé sur des aspects non supervisés ou bien supervisés. La première a la particularité de s'abstraire de la spécificité des données traitées. Parmi ces méthodes nous pouvons citer les méthodes linguistiques [J. Antonio et al, 2014], méthodes statistique [Van Eck et al, 2010], méthodes par regroupement [Z.Liu et al, 2009] ou bien à base de graphe [X.WAN et al. 2008]. Pour ce qui concerne l'aspect supervisé [Sarkaret al, 2010], ce genre de méthodes utilisent des modèles de décision prise à partir des techniques statistiques et linguistiques ou bien calculés sur les termes-clés d'un corpus d'apprentissage [Z.Liu et al, 2011]. Le choix de la méthode d'extraction de termes dépend de la stratégie de collecte des données, du type de variable, et de la précision souhaitée.

2.1.1 Les Approches statistiques

Les techniques statistiques sont généralement utilisées pour faire de la comparaison de corpus [T .Chung, 2003], c'est à dire pour comparer un corpus d'analyse à un corpus de référence afin d'identité des termes propres au corpus d'analyse. Ces calculs peuvent être basés sur la présence d'un mot dans un texte, sa fréquence dans le texte ou bien sa fréquence dans un texte en comparaison avec sa fréquence d'une collection. Dans ce qui suit une liste des méthodes statistiques qui sont apparus dans la littérature :

a) La méthode TF-IDF (Term Frequency Inverse Document Frequency)

D'après [K. Jones, 1972] est une méthode statistique très utilisée pour expliquer les approches statistiques, elle s'exprime par le fait qu'un terme à une forte importance vis-à-vis du document/texte analysé s'il y est très présent, alors qu'il ne l'est pas dans le reste de la collection

de telle sorte que plus le score **TF-IDF** d'un terme candidat est élevé plus celui-ci est important dans le document/texte analysé. Elle est définie comme suit (**Formule 6**) :

$$TF - IDF(Terme) \times \log\left(\frac{N}{DF(Terme)}\right) \dots \dots \dots (6)$$

TF :représente le nombre d'occurrences d'un terme dans le document analysé,

DF :représente le nombre de documents dans lequel il est présent,

N :le nombre total de documents.

D'un point de vue général, cette approche semble être non performante pour l'extraction des termes qui apparaît rarement dans le document ou bien dans le texte.

b) **La méthode TFC(Term Frequency based Cosinus) :**

C'est une autre mesure de pondération, appelée TFC semblable à $TF \times IDF$ et qui corrige la longueur du texte par une normalisation du cosinus, pour éviter de donner plus de crédit aux plus longs documents. Elle est décrite avec la fonction donnée dans la formule 7 suivante :

$$TFC(t_k, d_j) = \frac{TF \times IDF(t_k, d_j)}{\sqrt{\sum_{k=1}^{|T|} (TF \times IDF(t_k, d_j))^2}} \dots \dots \dots (7)$$

D_j : Document j t_k : Terme K

D'autres approches statistiques connues sur le même principe que TF-IDF sont apparues, nous citons ici la méthode **Likey** [Paukkeri et Honkela, 2010] et la méthode **Okapi** [Robertson *et al.*, 1999].

2.1.2 Les Approche linguistiques

Ces méthodes [Gaizauskas *et al.*, 2000], [Krauthammer et Nenadic, 2004] reposent sur un degré de figement syntaxique des termes, ou sur certains aspects de leur structure interne (mécanismes de formation, composition morphosyntaxique.....etc). Généralement, les textes utilisés en entrée sont des textes en langue naturelle. Pour réussir un traitement juste de ces textes et extraire des connaissances à partir de ceux-ci, il faut qu'ils passent par des étapes appelées généralement prétraitement. L'étape de prétraitement des textes appartient au domaine du traitement automatique de la langue naturelle. Elle comporte en général les phases suivantes:

- ✓ **Détection de la langue du texte:** Pour commencer le traitement, il faut d'abord savoir dans quelle langue chaque texte est écrit et comment cette langue est encodée. Il est important de détecter avec précision la langue dans laquelle le texte est rédigé, car une erreur à ce niveau voue à l'échec les étapes suivantes. Il existe deux familles d'approches dans l'identification de la langue: linguistique et statistique.
- ✓ **Nettoyage des données :** Habituellement, le nettoyage consiste à éliminer les mots vides (stop-words). Ces mots vides sont des mots ne jouant qu'un rôle syntaxique, contribuant peu au sens des documents.
- ✓ **Lemmatisation :** La lemmatisation est l'opération qui consiste à ramener les variantes (flexionnelles) d'un même mot à une forme canonique, le lemme. Elles'appuie sur une

analyse grammaticale des textes afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier.

2.1.3 Les approches par regroupement

L'objectif des approches par regroupement est de définir des groupes dont les unités textuelles partagent une ou plusieurs caractéristiques communes. Plusieurs travaux dans la littérature ont été apparus comme le travail de [Matsuo et Ishizuka ,2004] qui consiste à comparer les termes candidats du document analysé avec les groupes de termes fréquents en fonction de leur lien sémantique. Le travail de [Liu et al. 2009] utilisent aussi un regroupement sémantique de telle sorte que les mots qui sont les plus proches du centroïde sont sélectionnés comme des mots de référence. L'ensemble des mots de référence est ensuite utilisé pour filtrer les termes candidats. On distingue aussi d'autres approches de regroupement qui seront décrit ci-dessous tel que la méthode K-means, K-médoïds.....etc.

2.1.3.1 La méthode K-means :

L'algorithme K-means [J. A. Hartigan and M. A. Wong, 1979] est l'outil de classification le plus utilisé dans les applications scientifiques et industrielles. Cette méthode construit k classes à partir d'un ensemble de n individus, tout en minimisant la quantité selon la formule8 suivante :

$$\sum_{i=1}^k \sum_{x_j \in C_i} (x_j - y_i)^2 \dots\dots\dots(8)$$

Ou :

C_i est la classe numéro i x_j est un individu dans une classe y_i est le centre de classe C_i

La somme des carrés des erreurs peut être rationalisée comme le log vraisemblance pour les modèles de mélanges distribués normalement. Le but est de trouver un clustering avec une dispersion intra-cluster minimale. L'algorithme général de cette méthode est résumé comme suit :

Algorithme K-means

Variables: i,j,k ; # k est le nombre maximum de classes #

Debut

1. Choisir k individus au hasard (comme centre des classes initiales) ;
2. Affecter chaque individu au centre le plus proche ;
3. Recalculer le centre de chacune de ces classes ;
4. Répéter l'étape (2) et (3) jusqu'à stabilité des centres ;
5. Editer la partition obtenue ;

Fin

2.1.3.2 La méthode de K-médoïdes :

C'est une méthode itérative combinant la réaffectation des individus dans des classes avec une intervention des médoïdes et des autres individus. Elle minimise l'erreur quadratique moyenne qui est la distance entre les points de la classe et le point central (médoïde). Dans la méthode K médoïde une classe est représentée par un de ses points. C'est une méthode simple parce qu'elle couvre n'importe quel type de variables, quand des médoïdes sont choisis, des classes sont définies comme sous-ensembles des individus près des médoïdes les plus proches par rapport à une mesure de distance choisie.

Si en comparant cette méthode avec la précédente, dans k-moyennes, chaque cluster est représenté par son centre tandis que dans les k-médoïdes, chaque cluster est représenté par un des objets du cluster.

2.1.4 Les approches à base de graphe

Les approches à base de graphe sont actuellement très populaires et utilisées dans de nombreuses applications du TAL [Kozareva et al, 2013]. Elles consistent à représenter le contenu d'un document sous la forme d'un graphe. Les graphes ayant l'avantage de proposer une modélisation simple et intuitive du document notamment des textes. Nombreuses méthodes à base de graphe sont apparues comme :

a) TextRank

Une méthode d'ordonnement d'unités textuelles à partir d'un graphe pour le résumé automatique et l'extraction de mots-clés, elle est basée sur le calcul du score d'importance des sommets en utilisant le principe de vote ou de recommandation entre deux sommets [Mihalcea et Tarau, 2004]. TextRank est une adaptation de base de PageRank [Brin et Page, 1998], cette méthode utilise une représentation efficace d'un document/Texte, elle peut aussi être utilisée pour faire des résumés automatiques d'un document, elle repose sur les étapes suivantes : construction de graphe, calcul des scores pour les sommets et extractions des termes clés. Par conséquent, les nœuds du graphe représentent les mots du document/texte et les arêtes qui les connectent représentent leurs relations adjacentes.

Soit le graphe d'occurrences de mots non orienté $G = (N, A)$, où les nœuds N représentent les mots du document, et où les arêtes A les connectent lorsqu'ils apparaissent ensemble dans le document. L'importance de chaque mot n_i est obtenue itérativement selon la formule TextRank suivante (Formule 9) :

$$S(n_i) = (1 - \lambda) + \lambda \times \sum_{n_j \in A(n_i)} \frac{S(n_j)}{|A(n_j)|} \dots \dots \dots (9)$$

Où $A(n_i)$ est l'ensemble des nœuds connectés au nœud n_i et où λ est un facteur d'atténuation. Nombre réel défini entre 0 et 1, ce dernier peut être considéré comme la probabilité pour que le nœud n_i soit important d'après le principe de la recommandation.

b) La methode TopicRank

La méthode **TopicRank**, fondée sur TextRank, est différente par rapport aux autres méthodes à base de graphe, parce qu'au lieu de faire une recherche des unités textuelles importantes du document/texte, elle cherche ses sujets importants. Par rapport à TextRank et SingleRank, elle présente les avantages suivants : suppression des problèmes de redondance dans les termes-clés extraits, construction d'un graphe plus compacte, renforcement des poids des arêtes dans le graphe et amélioration de la qualité d'ordonnement.

c) La méthode Kcore

Comme cela a été décrit par les auteurs dans [Rousseau et al, 2015], la construction de son graphe de mots est semblable à celle de TextRank. Elle utilise l'algorithme défini dans le travail [Batagelj et al, 2011] pour calculer les scores d'importance des sommets. Contrairement à la méthode TopicRank qui utilise le principe de vote ou de recommandation. Mais la seule contrainte est qu'elle dépend de la fenêtre d'appartenance des mots.

Plusieurs variations ont été proposées dans la littérature pour définir l'approche statistique d'extraction des termes. Le travail présenté dans [T.Hisamitsu al,2000] inclut la méthode TF-IDF pour mesurer l'importance d'un mot dans un document/texte donné par rapport à d'autres documents/texte.

D'autres travaux ont aussi abordé l'approche statistique comme le travail présenté dans [M.Paukkeri et al, 2010] et [M.Paukkeri, I.Nieminen et al, 2008] qui ont intégré deux aspects supervisés et non supervisés dans le même travail. Les auteurs dans [Z.Liu et al , 2011] ont utilisé un modèle d'alignement des mots (**Wordalignmentmodel**) et propose une plateforme pour l'extraction des phrases clés.

De plus, afin d'illustrer les méthodes de regroupement, les auteurs dans [Z.Liu, P.Li et al, 2009] ont proposés une approche non supervisé pour l'extraction des phrases. Premièrement, la méthode trouve les termes pertinents utilisant des techniques de regroupement afin de garantir que sémantiquement le document est couvert par ces termes clés. Ensuite les phrases clé sont extraites utilisant ces termes.

3. Les Mesures de similarité

3.1 Introduction

La notion de similarité consiste à comparer des mots, des phrases, des textes ou bien des documents entiers. Ces textes sont de petites tailles ou bien de grande taille selon le contexte de l'étude. Ce type de calculs pourront être réalisés localement ou bien à distance avec un seul défit c'est que le résultat soit pertinent et fiable.

La similarité entre documents textuels est une des problématiques qui a été implémenté dans plusieurs disciplines : l'analyse de données textuelles, la recherche d'information [O.Medelyan et I .Witten, 2008] où l'extraction de connaissances à partir de données textuelles (TextMining). Dans chacun de ces domaines, les similarités sont utilisées comme ceci:

- ✓ Dans l'analyse de données textuelles : les similarités sont utilisées pour la description et l'exploration de données. Ce type d'analyse a été utilisé dans notre cas d'étude pour le traitement des requêtes entré par l'utilisateur.
- ✓ Dans la recherche d'information : Les similarités entre documents et requêtes sont utilisées pour identifier les documents pertinents par rapport à des besoins d'information exprimés par les utilisateurs.

Nous présentons dans cette section certaines approches permettant de comparer des textes notamment les documents. Cette étude ne présente pas la liste exhaustive de toutes les méthodes existantes, mais c'est un aperçu pour les méthodes les plus utilisées dans le contexte de notre étude.

3.2 Types de similarité

L'identification de la similarité a été considérée comme un sujet de recherche fortement recommandé dans plusieurs domaines tel que l'intelligence artificielle, le Web sémantique, et la littérature linguistique. Les techniques mises en œuvre pour calculer ces similarités varient bien évidemment selon les disciplines. Par exemple, dans le domaine du Web sémantique les ontologies qui interviennent pour la modélisation des connaissances. Dans ce qui suit un résumé de principales similarités qui existe dans la littérature

3.2.1 Les similarités syntaxiques et lexicales

En mathématiques et en informatique, une mesure permettant de comparer des documents textuels, consiste à comparer des chaînes de caractères. C'est une métrique qui mesure la similarité ou la disimilarité entre deux chaînes de caractères. Par exemple, les chaînes de caractères "Sam" et "Samuel" peuvent être considérées comme similaires. Une telle mesure sur les chaînes de caractères fournit une valeur obtenue algorithmiquement.

Parmi de telles mesures de similarité, citons par exemple, la distance de Levenshtein (ou distance d'édition), le coefficient de Dice, Overlap Coefficient, Dice's Coefficient, l'indice de Jaccard, la distance euclidienne, LCS, Smith- Waterman , N-gram, Needleman-Wunsch Jaro, le cosinus, ...etc.

3.2.1.1 La similarité cosinus : [Baeza-Yates and Ribeiro-Neto, 1999] est fréquemment utilisée en tant que mesure de ressemblance entre deux documents d_1 et d_2 . Il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des documents à comparer. La similarité obtenue est : $sim_{cosinus}(d_1; d_2) \in [0; 1]$. (Voir Formule 10)

$$Sim_{cosinus}(d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{\|\vec{d_1}\| \cdot \|\vec{d_2}\|} \dots \dots \dots (10)$$

3.2.1.2 L'indice de Jaccard: [Jaccard, 1901] est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents d_1 et d_2 sont donc représentés, non

pas comme des vecteurs, mais comme des ensembles de termes. La similarité obtenue est : $sim_{jaccard}(d_1; d_2) \in [0; 1]$. (Voir Formule11).

$$sim_{jaccard}(d_1, d_2) = \left\| \frac{d_1 \cap d_2}{d_1 \cup d_2} \right\| \dots \dots \dots (11)$$

Il est aussi possible d'utiliser la représentation vectorielle suivante (Formule12):

$$sim_{jaccard}(d_1, d_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{\|\vec{d}_1\| \|\vec{d}_2\| - \vec{d}_1 \cdot \vec{d}_2} \dots \dots \dots (12)$$

3.2.1.3 La distance de Levenshtein : [Levenshtein, 1966] calcule la similarité entre les représentations sous forme de chaînes de caractères des documents d_1 et d_2 . Il s'agit du coût minimal, i.e. du nombre minimal d'opérations d'édition, pour transformer d_1 en d_2 . Les opérations sont les suivantes :

- ✓ substitution d'un caractère de d_1 en un caractère de d_2 ,
- ✓ ajout dans d_1 d'un caractère de d_2 ,
- ✓ suppression d'un caractère de d_1 .

Pour obtenir la distance de Levenshtein $sim_{levenshtein}(d_1; d_2)$ entre les documents d_1 et d_2 , il s'agit d'associer à chacune de ces opérations un coût. Le coût des opérations est toujours égal à 1, sauf dans le cas d'une substitution de caractères identiques. Notons que cette distance a été étendue pour prendre en compte la grammaire, la phonétique, ...etc.

3.2.1.4 L'indice de Dice : mesure la similarité entre deux documents d_1 et d_2 en se basant sur le nombre de termes communs à d_1 et d_2 . (Voir Formule13).

$$sim_{dice}(d_1, d_2) = \frac{2N_c}{N_1 + N_2} \dots \dots \dots (13)$$

Où

- N_c est le nombre de termes communs à d_1 et d_2 ,
- N_1 (resp. N_2) est le nombre de termes de d_1 (resp. d_2).

3.2.1.5 Coefficient de corrélation de Pearson : Le coefficient de corrélation de Pearson calcule la similarité entre deux documents d_1 et d_2 comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue est comme mentionné dans la Formule14.

$$Sim_{pearson}(d_1, d_2) = sim_{cosinus}(d_1 - \bar{d}_1, d_2 - \bar{d}_2) \dots \dots \dots (14)$$

Où

\bar{d}_1 resp. \bar{d}_2 représente la moyenne de d_1 resp. d_2 .

3.2.1.6 La distance euclidienne : calcule la similarité entre deux documents d_1 et d_2 comme la distance entre leurs représentations vectorielles ramenées à un seul point. (Voir Formule15).

$$Sim_{euclidienne}(d_1, d_2) = \|\vec{d}_1 - \vec{d}_2\| = \sqrt{\sum_{i=1}^n (d_{1_i} - d_{2_i})^2} \dots \dots \dots (15)$$

Où

n est le nombre total de termes représentés, i.e. la taille des vecteurs.

3.2.2 Avantages et inconvénients des approches syntaxiques et lexicales

Les techniques basées sur l'approche syntaxique ne laissent pas de place aux exceptions :

- ✓ Elles sont donc facilement automatisables.
- ✓ Par définition, les techniques basées sur l'approche syntaxique ne prennent pas en compte la sémantique.

Par conséquent, la prise en compte de la sémantique semble importante. En effet, les inconvénients de ces approches syntaxiques et lexicales sont comme suit :

- ✓ Les relations syntaxiques sont ignorées. Par exemple, aucune différence n'est faite entre "Processeur endommage la machine virtuelle" et "La machine virtuelle endommage processeur". Dans le contexte de notre étude, les relations syntaxiques peuvent influencer sur la pertinence.
- ✓ Les rôles sémantiques sont ignorés. Par exemple, "Cisco achète EMC" et "EMC a été achetée par Cisco", seule la forme verbale change. Cela peut engendrer des problèmes de pertinence. Une proposition serait peut-être d'analyser les classes verbales.
- ✓ Les problèmes liés aux négations (par exemple, "système d'exploitation endommagé" et "système d'exploitation n'est pas endommagé") semblent encore difficiles à pallier.

3.2.3 Similarité sémantique

Le défi des mesures de similarité sémantique consiste à trouver une méthode identique au raisonnement de l'être humain. L'utilisation d'ordinateurs pour quantifier et comparer les similitudes sémantiques est devenue un domaine important de la recherche dans divers domaines, y compris l'intelligence artificielle, la recherche d'information [O.Medelyan et .Witten,2008], traitement de langues naturelles,...etc.

Parmi les mesures de similarité sémantique on retrouve les mesures de similarité sémantique qui sont fondées sur la connaissance [Wu and Palmer, 1994] ou sur un corpus [Olivier Ferret,2010] Ces mesures ont été appliquées avec succès à des tâches de traitement du langage comme, par exemple, la désambiguïsation du sens des mots [Andon Tchechmedjiev,2012].

La définition de mesures sémantiques au niveau lexical a fait l'objet de nombreux travaux depuis plusieurs années. Les auteurs dans [Olivier Ferret,2010] focalisent leur travail plus particulièrement sur les approches à base de corpus par extraire des synonymes de noms à partir de corpus en s'appuyant sur l'hypothèse distributionnelle, ce qui nécessite en premier lieu de choisir une mesure de similarité sémantique adéquate. Parmi les approches sémantiques qui existent on cite :

3.2.3.1 Approches vectorielles

Cette approche se fonde sur la construction d'un modèle sémantique qui prenne en compte le contexte de mot et qui le projette dans un vecteur dans lequel certains liens sémantiques entre ces termes peuvent être observés et mesurés.

[Bisson and Hussain, 2008] ont proposé une approche sémantique vectorielle qui utilise la similarité entre les mots et la similarité entre les documents pour segmenter la matrice. Leur méthode est basée sur la recherche de similarité document-document et de similarité mot-mot, puis utilise les méthodes classiques de classification. Parmi les méthodes vectorielles on cite :

a) Bi-clustering

La classification double ou co-clustering ou bi-clustering est une technique d'exploration de données non supervisée permettant de segmenter simultanément les lignes et les colonnes d'une matrice. Étant donné un ensemble de r lignes à c colonnes (c'est-à-dire une matrice $r \times c$), l'algorithme de bi-clustering génère des bi-clusters (un sous ensemble de lignes qui présentent un comportement similaire sur un sous-ensemble de colonnes, ou vice versa).

Le bi-clustering est utilisé dans le domaine de la fouille de texte, où il est populairement connu en tant que co-clustering [Bisson et Hussain, 2008]. Les corpus de textes sont représentés sous une forme vectorielle : comme une matrice D dont les lignes sont les documents et les colonnes sont les mots du dictionnaire.

Les éléments D_{ij} de la matrice désignent l'occurrence du mot j dans le document i . Les algorithmes de bi-clustering sont ensuite appliqués pour découvrir des blocs dans D qui correspondent à un groupe de documents (lignes) caractérisé par un groupe de mots (colonnes).

b) Vecteurs sémantiques

L'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Une manière simple de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de similarité vectorielles.

Le plus difficile est d'obtenir de tels vecteurs. Il faut donc construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé. Les vecteurs sont définis dans un espace vectoriel orthogonal à n dimensions où chaque base se voit attribuer un mot de vocabulaire unique (donc chaque entrée du dictionnaire a une base dans l'espace vectoriel). Pour chaque mot du dictionnaire, on détermine un vecteur dans cet espace, où la composante du vecteur pour chaque base est le nombre d'occurrences du mot dans la base qui le représente où il apparaît dans le contexte du mot pour lequel un vecteur a été construit.

3.2.3.2 Approches Ontologique (knowledge-based)

Plusieurs efforts ont été faits pour définir la similarité sémantique à base d'ontologie. Les plus connus sont ceux qui utilisent une ontologie comme WordNet [Fellbaum, 1998]. L'objectif de telle mesure est d'évaluer la proximité sémantique entre les concepts (auxquels les termes des requêtes et documents sont rattachés). En recherche d'information, les mesures de similarité jouent un rôle important, en particulier dans le processus de désambiguïsation des concepts, la pondération des concepts et l'évaluation de la pertinence. De nombreuses approches ont été proposées pour évaluer la similarité sémantique entre deux concepts. Ces approches se divisent [T.Slimani et al, 2006] en trois catégories : les approches basées sur les chemins, les approches basées sur le contenu informationnel et les approches hybrides.

a) Définition Ontologie

Une ontologie est une représentation partagée et consensuelle dans le but est de définir un ensemble de connaissance dans un domaine donné. Elle explicite un vocabulaire en définissant les termes nécessaires pour partager la connaissance liée à ce domaine. Il existe plusieurs types d'ontologies et ses applications sont diverses.

- ✓ **Ontologie d'information** : Elle contribue à organiser et clarifier les idées des collaborateurs sur un projet en exposant le schéma global du système avec tous ses liens et ses raisonnements. C'est un schéma qui est fait à la main en groupe devant un tableau pour se mettre d'accord sur la définition des termes employés, avec les liens entre eux.
- ✓ **Ontologie de Domaine** : ce type d'ontologie est fonctionnel et orientée objet. Elle est utilisée pour représenter un domaine sous forme de base de connaissance. Par exemple, faire l'ontologie d'un site web est intéressant pour prendre sa structure. Elle présente les concepts clé, les attributs, les instances relatifs au domaine. Pour le réaliser, il existe des éditeurs de structure de base de connaissance. Le plus connu est le plus utilisé et PROTEGE (Open Source gratuit et qui implémenté à partir d'une plateforme J2EE).
- ✓ **Ontologie Informatique** : Les ontologies fournissent une spécification formelle d'une conceptualisation partagée. Elles sont exploitables par une machine et construisent à partir du consensus d'une communauté d'utilisateurs ou des experts de domaine. Les ontologies représentent une source de connaissance très fiable et structurée. Pour cette raison et grâce aux initiatives du Web sémantique, qui a apporté la création des milliers d'ontologies de domaine, les ontologies ont été largement exploitées dans les systèmes à base de connaissance, et plus précisément, pour le calcul de la similarité sémantique.

b) WordNet

Est une base de connaissances ou taxonomie qui organise plus de 100,000 concepts en anglais. Elle regroupe les concepts en ensembles de synonymes appelés synsets. Ces derniers sont des ensembles de concepts pour lesquels le mot peut correspondre. Cependant, une base similaire a été créée pour la langue française : **WOLF** (WordNet Libre du Français). Enfin, il faut noter que les concepts sont de plus en plus abstraits et généraux lorsqu'on va vers la racine et qu'ils sont plus spécifiques lorsqu'on va vers les feuilles.

c) Composant d'une ontologie

Les composants d'une ontologie sont :

- ✓ **Les concepts** : une description d'une tâche, d'une fonction, d'une action, d'une stratégie ou d'un processus de raisonnement...etc.
- ✓ **Les relations** : Représentent un type d'interaction entre les notions d'un domaine.
- ✓ **Les Axiomes** : Une ontologie est composée d'axiomes qui forment des contraintes sémantiques pour le raisonnement et donnent un acompte d'une conceptualisation.

d) Mesure de similarité Ontologique

Ces mesures peuvent être classées en deux grandes catégories : approche basé sur les chemins et approche basé sur le contenu de l'information.

1. **les mesures de similarité basée sur les chemins** : l'idée est le fait que la similarité entre deux concepts est en fonction de la longueur du chemin reliant les concepts et leurs positions dans la taxonomie. L'avantage principal des mesures basées sur les chemins est leur simplicité. Ils comptent seulement sur le modèle graphique d'une ontologie et dont l'évaluation exige un coût informatique bas (en comparaison avec les approches qui traitent des corpus de texte). [Rada et al., 1989] ont suggéré que la similarité dans un réseau sémantique peut être calculée en se basant sur les liens taxonomiques « is-a ». Plus généralement, le calcul de la similarité entre concepts peut être basé sur les liens hiérarchiques de spécialisation/généralisation. La relation entre concept est calculée comme ceci (Voir Formule16) :

$$Rel(c1, c2) = T - chemin - k \times d \dots \dots \dots (16)$$

Où

T et **K** sont des constantes, **chemin** est la longueur du chemin le plus court en nombre d'arcs et **d** est le nombre de changements de direction.

L'idée est que deux termes sont proches sémantiquement si leurs synsets sont connectés par un chemin qui n'est pas très long et qui ne change pas souvent de direction. S'il n'y a pas de chemin, le calcul est égal à zéro.

Plusieurs méthodes concerne l'approche topologique à base de chemin existe dans la littérature, on cite les plus important :

- ✓ **Wu et Palmer** : La métrique de similarité de Wu et Palmer [Wu and Palmer, 1994] mesure la profondeur de deux concepts donnés dans la taxonomie WordNet, et la profondeur de leur plus bas ancêtre commun (lowest common subsumer(LCS)) et les combine pour obtenir un score de similarité (Voir Formule17) :

$$Sim_{wvp}(c_1, c_2) = \frac{2 \times profondeur(LCS)}{profondeur(c_1) + profondeur(c_2)} \dots \dots \dots (17)$$

- ✓ **Edge-based** : L'approche basée sur les arcs est une manière naturelle et directe d'évaluer la similarité sémantique dans une taxonomie. Il s'agit d'estimer la distance (e.g., longueur des arcs) entre les noeuds correspondants aux concepts / classes à comparer. Compte tenu de l'espace multi dimensionnel des concepts, la distance conceptuelle peut facilement être mesurée par la distance géométrique entre les noeuds représentants les concepts. Évidemment, plus le chemin d'un noeud à l'autre est court, plus ils sont similaires.

2. **Contenu basé sur l'information (ou information content-based)** : Une approche basée sur les noeuds pour déterminer la similarité conceptuelle est appelée une approche information content-based [Resnik, 1995]. Étant donné un espace multi dimensionnel où un noeud représente un concept unique composé d'un certain nombre d'informations, et où un arc représente une association directe entre deux concepts, la similarité entre deux concepts est la mesure dans laquelle ils partagent des informations en commun. Compte tenu de cette notion de structure hiérarchique / espace de classes, ces informations communes peuvent être identifiées comme un noeud/concept spécifique qui englobe les deux dans la hiérarchie. Plus précisément, cette super-classe devrait être la première classe en haut de la hiérarchie qui englobe les deux classes. La valeur de similarité est définie comme la valeur du contenu de l'information de cette super-classe. La valeur du contenu de l'information d'une classe est ensuite obtenue en estimant la probabilité d'occurrence de cette classe dans un grand corpus de texte. Le contenu de l'information (IC) d'un concept / d'une classe c est donné par la formule 18 suivante :

$$IC(c) = -\log P(c) \dots \dots \dots (18)$$

Où

$P(c)$ est la probabilité de rencontrer une instance du concept c .

Plusieurs méthodes pour les approches sont basées sur le contenu de l'information. Parmi lesquelles nous citons :

- ✓ **Lin** : La mesure proposée par [Lin, 1998] est une normalisation de celle de [Resnik, 1995]. La normalisation est faite en factorisant par le contenu de l'information (IC) des deux concepts.
- ✓ **Resnik** : La notion de contenu informationnel (CI) a été la première fois introduite par Resnik [Resnik, 1995]. Elle utilise conjointement l'ontologie et le corpus. Le contenu informationnel d'un concept traduit la pertinence d'un concept dans le corpus en tenant compte de la fréquence de son apparition dans le corpus ainsi que de la fréquence d'apparition des concepts qu'il subsume. On dit qu'un concept $C1$ subsume un concept $C2$ si $C2$ est plus spécifique que $C1$. Plus précisément le contenu de l'information (IC) du plus bas ancêtre commun (LCS) de deux concepts se calcule de la manière par la formule 19 suivante :

$$Sim_{rcs}(c_1, c_2) = IC(LCS(c_1, c_2)) = -\log P(LCS(c_1, c_2)) \dots \dots (19)$$

3.2.3.3 Approches statistiques (ou corpus-based)

Les mesures basées sur des corpus [M. Aminul et D. Inkpen, 2006] diffèrent des mesures présentées précédemment car elles ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte. Parmi de telles mesures de similarité sémantique, nous présentons l'analyse sémantique latente (LSA) [Deerwester et al., 1990] où les occurrences des termes dans un corpus sont capturées au moyen d'une réduction de dimension réalisée par une décomposition en valeurs singulières (SVD) sur la matrice termes/documents représentant le corpus ; l'analyse sémantique explicite (ESA) [Gabilovich and Markovitch, 2007] qui est une variation du modèle standard vectoriel où les dimensions du vecteur sont directement équivalentes à des concepts abstraits.

- a) **LSA / PLSA et LDA** :[Deerwester et al., 1990] propose l'analyse sémantique latente (LSA), qui peut être utilisée pour déterminer la distance entre des mots ou entre des ensembles de mots. Contrairement aux diverses approches décrites précédemment où une matrice d'occurrence mots/mots est créée, la première étape de LSA consiste à créer des matrices mots/paragraphes, mots/documents ou mots/passages, où un passage est un groupe de mots. Ensuite, la dimension de cette matrice est réduite par l'application d'une décomposition en valeurs singulières (SVD), une technique de décomposition de matrice standard. Ce plus petit ensemble de dimensions représente un résumé de concepts. Puis la matrice originale mot/passage est recrée, mais cette fois à partir des dimensions réduites.

Lorsque vous utilisez la modélisation latente, les documents d'une collection sont modélisés comme une combinaison pondérée des thèmes latents d'un ensemble $Z = \{z_1, \dots, z_n\}$. Dans l'analyse sémantique latente probabiliste (**PLSA**) [Hofmann,1999], chaque thème latent possède un modèle de langage probabiliste $P(W|Z)$ représentant la probabilité que le mot w puisse être généré par le thème z . Chaque document d_i de la collection de documents D est alors supposé avoir été généré par un mélange pondéré des modèles latents des thèmes. Si un document est modélisé par une collection de mots $C = \{c_1, \dots, c_n\}$, le modèle généralisé **PLSA** de C sachant d_i est donné par la formule 20 :

$$P(C||d_i) = \prod_{w \in V} \left(\sum_{z \in Z} P(W|Z)P(Z||d_i) \right)^{c_w} \dots \dots \dots (20)$$

- b) **ESA** : L'analyse sémantique explicite (**ESA**) [Gabrilovich et Markovitch, 2007] est une représentation vectorielle de texte (mots isolés ou documents) qui utilise Wikipédia comme une base de connaissances. Plus précisément, dans l'ESA, un mot est représenté par un vecteur utilisant l'algorithme TF-IDF du texte dans Wikipédia et un document (chaîne de mots) est représenté comme le barycentre des vecteurs représentant ses mots. LSA fait l'hypothèse que les articles de Wikipédia sont "orthogonaux". Toutefois, il a été démontré que l'ESA améliore également les performances des systèmes de recherche d'information quand elle est fondée non pas sur Wikipédia, mais sur un corpus, qui ne satisfait pas la propriété d'orthogonalité.

- c) **SOC-PMI (Second Order Co-occurrence PMI)**: Cette méthode utilise l'information mutuelles pour trier les listes de mots voisins importants des deux mots cibles. Ensuite, elle considère les mots qui sont communs dans les deux listes et agrège leurs valeurs PMI pour calculer la similarité sémantique relative. Les résultats de l'évaluation dans le travail de [M. Aminul, D. Inkpen, 2006] montrent que cette méthode donne satisfaction que plusieurs méthodes concurrentes basées sur des corpus.

Soit deux mots W_1 et W_2 pour lesquels nous devons déterminer la similarité sémantique et $C = \{c_1, c_2, \dots, c_m\}$ désigne un grand corpus de texte contenant m mots. Soit aussi $T = \{t_1, t_2, \dots, t_n\}$ l'ensemble de tous les mots (types) uniques qui apparaissent dans le corpus C . Contrairement au corpus C , qui est une liste ordonnée contenant de nombreuses occurrences des mêmes mots, T est un ensemble qui ne contient aucun mot répété et W pour désigner soit W_1 soit W_2 . La fonction de similarité sémantique PMI entre ces deux mots W_1 et W_2 est donnée par la formule 21 suivante :

$$sim(w_1, w_2) = \frac{f^B(w_1)}{B_1} + \frac{f^B(w_2)}{B_2} \dots \dots \dots (21)$$

La valeur de B est liée au nombre de fois où le mot W apparaît dans le corpus, c'est-à-dire la fréquence de W ainsi que le nombre de types dans le corpus. Nous définissons β comme avec la formule 22 suivante :

$$B_i = (\log(f^t(w_i)))^2 \frac{(\log_2(n))}{\delta}, i=1,2 \dots\dots\dots(22)$$

Où δ est une constante qui dépend de la taille du corpus.

Aussi la fonction de sommation β -PMI pour le mot w_1 et w_2 est définie comme suit dans la formule 23 et formule 24 respectivement :

$$f^B(w_1) = \sum_{i=1}^{B_1} (f^{pmi}(x_i, w_2))^\delta \dots\dots\dots(23)$$

$$f^B(w_2) = \sum_{i=1}^{B_2} (f^{pmi}(y_i, w_1))^\delta \dots\dots\dots(24)$$

Des méthodes hybrides qui utilisent des approches statiques (à base de corpus) et des mesures basées sur la connaissance sont apparues pour déterminer la similarité de texte comme les travaux dans [R. Mihalcea, 2006] et [L.Y. Mclean, 2006]. Notre approche est basée sur une méthode hybride par l'exploitation de notre ontologie du domaine afin d'illustrer la similarité sémantique et une approche statique (à base de corpus) utilisant l'algorithme SOC-PMI [M. Aminul, D. rerféInkpen, 2006].

3.2.2.5 Avantages et inconvénients des approches sémantiques

Comparativement aux approches syntaxiques, certains inconvénients ont été palliés, d'autres subsistent. En effet, les problèmes liés à la négation et l'antinomie, aux rôles sémantiques inverses et à l'inconsistance logique ne sont toujours pas réglés. De plus, il est à noter que les approches sémantiques basées sur les corpus ou la connaissance posent des problèmes de stockage et de complexité et souvent spécifiques à un domaine donné.

4. Conclusion

Ce chapitre a permis d'introduire quelques méthodes pour l'extraction automatique de mots-clés. Certaines se fondent uniquement sur des statistiques et d'autres les combinent avec des représentations plus complexes du document ou bien texte comme des groupes sémantiques ou des graphes de cooccurrences de mots.

Ensuite nous avons parlé des mesures de similarité existantes avec ses différentes approches afin de justifier notre choix sur les méthodes implémentées qui répondent au mieux à notre contexte d'étude.

Chapitre4

Conception du MAITD

Plan du chapitre4

1. Introduction
2. Architecture Proposée
 - 2.1 La coucheprésentation
 - 2.2 La coucheinterprétation
 - 2.3 La couche de donnée
3. Architecture MAITD a base de module
 - 3.1 Prétraitement des données
 - 3.2 Eextraction des termes
 - 3.3 La similarité des données
 - 3.4 La résolution du problème
4. Extraction des Termes
 - 4.1 La Pertinence dans le domaine
 - 4.2 La fréquence des termes TF-IPF
5. La mesure de similarité textuelle
 - 5.1 La mesure de similarité Lexical
 - 5.2 La mesure de similarité sémantique
6. Agrégation des Solutions
7. Conclusion

1. Introduction

Dans ce chapitre, nous présentons notre approche qui vise à faciliter la tâche de diagnostic dans les infrastructures informatique. Nous considérons un utilisateur ou bien un initiateur de requêtes proprement dit, et des fournisseurs d'informations, utilisant respectivement des Corpus d'information (Interventions archivées, Ontologie du domaine, Information via Web...) comme principaux acteurs.

Un des principaux challenges de la mise en place de ce système de diagnostic est d'être tolérant aux différentes pannes que nous pourrions rencontrer avec les infrastructures IT. De ce fait le système regroupe les différentes pannes considérées comme similaires, de telle sorte que les données d'un même groupe soient le plus similaires entre elles, et les données des groupes distincts présentent le moins de similarité possible. Le but étant de générer un diagnostic automatique (non-supervisée) pour cela notre choix est porté sur une approche hybride qui combine entre plusieurs méthodes: Méthodes statistiques à base de chaine pour éliminer les mauvaises descriptions des problèmes rencontrés et des méthodes sémantiques à base de corpus et une ontologie simultanément afin de prendre en compte le sens du mot dans la description.

Dans ce chapitre, nous commençons par la présentation générale de l'architecture proposée ainsi que les différents éléments qui les composent (Section 4.1). Ensuite, nous définissons une deuxième architecture modulaire listant les différents modules et paradigmes proposés par MAITD (Maintenance Architecture Information Technology Diagnostic). Les méthodes d'extraction et les approches de similarité implémentées dans notre travail ont été définies dans les sections 4 et 5 respectivement. Enfin la notion d'agrégation des alternatives a été présentée dans la section 6.

2. Architecture Proposée

Notre architecture **MAITD-2** [S.Elandaloussi, 2020] est une version améliorée de l'architecture MAITD [S.Elandaloussi, 2019]. C'est une architecture multi couches qui est organisée en trois couches: présentation, interprétation et la couche de données. Chaque couche incorpore plusieurs agents avec des composants ayant des fonctionnalités typiques. La première couche appelée couche de présentation est constituée de deux types d'agents: l'agent participant et l'agent expert. Ces agents agissent comme une interface avec le système pour traiter toutes les données d'entrée ou de sortie. De plus, le reste des agents (Agent Analyseur, Agent Recherche, Agent Réunion) est inclus dans la deuxième couche au milieu de notre architecture. Dans notre approche, cette dernière est considérée comme un noyau avec ses fonctions et ses méthodes qui sont modélisées dans un environnement Java. Fondamentalement, la couche de données est une couche physique qui rassemble toutes les sources de données qui sont demandées. Dans ce qui suit, nous avons décrit séparément les différentes couches qui sont définies auparavant en mettant en évidence les tâches principales de chaque agent qu'elles contiennent.

2.1 La couche présentation

La couche de présentation est chargé de faire le traitement et de la livraison des informations entrante au niveau des couches les plus basses. De plus, c'est la première couche applicative de notre système qui est considérée comme l'interface utilisateur dédiée à afficher des informations

généérées à l'utilisateur. Les principaux agents représentés dans cette couche sont définis ci-dessous:

2.1.1 Agent Participant

Comme le montre la figure 20, notamment à l'étape d'extraction des termes, cet agent décompose les problèmes en termes pertinents afin de traiter les informations entrées par l'utilisateur. De plus, il est responsable de la transmission de toutes les notifications système au participant correspondant. Le module le plus important de cet agent est nommé module de décomposition qui permet de sélectionner dans chaque problème les termes les plus pertinents nécessaires pour accélérer la recherche d'informations en appliquant un ensemble de fonctions et d'algorithmes qui sont présentés dans le chapitre précédent. La structure interne de cet important agent est donnée dans la figure 12 :

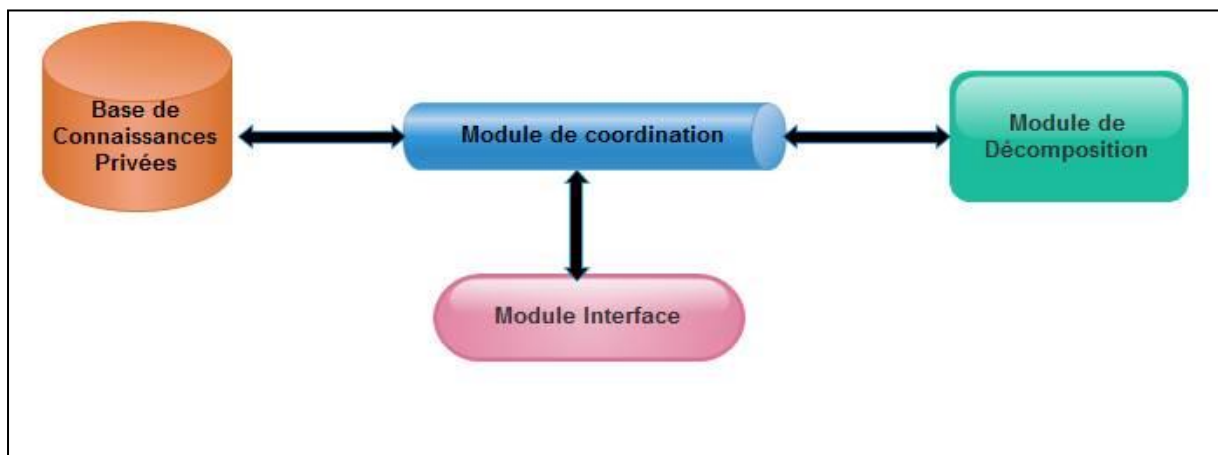


Figure 12 : Structure interne de l'agent participant.

2.1.2 Agent Expert

Cet agent gère les informations personnelles de l'expert, telles que son agenda, ses préférences et son profilage. Ainsi, il accepte et refuse toute invitation à une réunion. De plus, il peut intervenir lors de la construction et de l'enrichissement de l'ontologie du domaine (Voir chapitre 5) en validant la classification des termes dans leur contexte taxonomique correspondant pour préserver la qualité de l'information. Parmi les modules constituant la structure interne de cet agent, nous avons le module profil qui reprend le profil et la préférence d'expert global. Le module de planification qui prend place dans l'organisation de la collaboration et qui est construit en fonction des informations de disponibilité de l'expert. La Figure 13 résume la structure interne de cet agent.

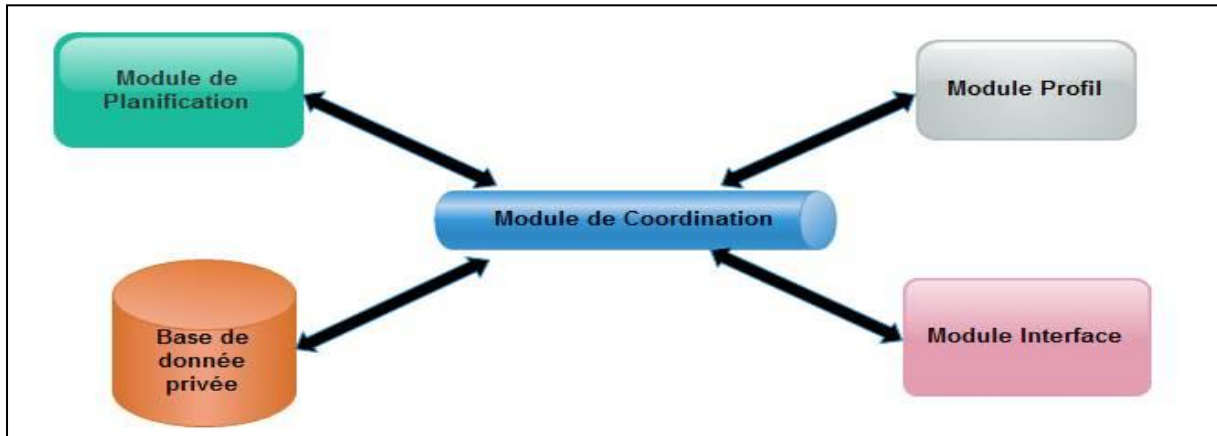


Figure13 : La structure interne de l'agent expert.

2.2 La Couche interpretation

C'est la couche noyau de notre approche avec tous ses fonctions et méthodes. L'ensemble des agents qui implémentent ses méthodes et qui visent à atteindre les objectifs généraux de notre architecture MAITD sont décrits comme suit :

2.2.1 Agent Analyseur

L'ensemble des problèmes similaires (problèmes candidat) à notre problème initial sont filtrés et sélectionnés par ce type d'agent à partir de plusieurs sources de données pour construire notre corpus de données. Les fonctionnalités de certains modules (présentées dans le Figure14) est comme suit:

- ✓ **Module de similarité (SM):** permet de générer la matrice de performance en appliquant des mesures de similarité statique et de la similarité basée sur la connaissance, qui sont définies dans la section5 de ce chapitre.
- ✓ **Module d'agrégation (AM):** est utilisé pour avoir un classement fiable entre toutes les solutions proposées par l'implémentation d'une fonction spécifique appelée l'algorithme **WASPAS** (Weighted Aggregated Sum Product Assessment) déjà introduit dans le chapitre précédent.

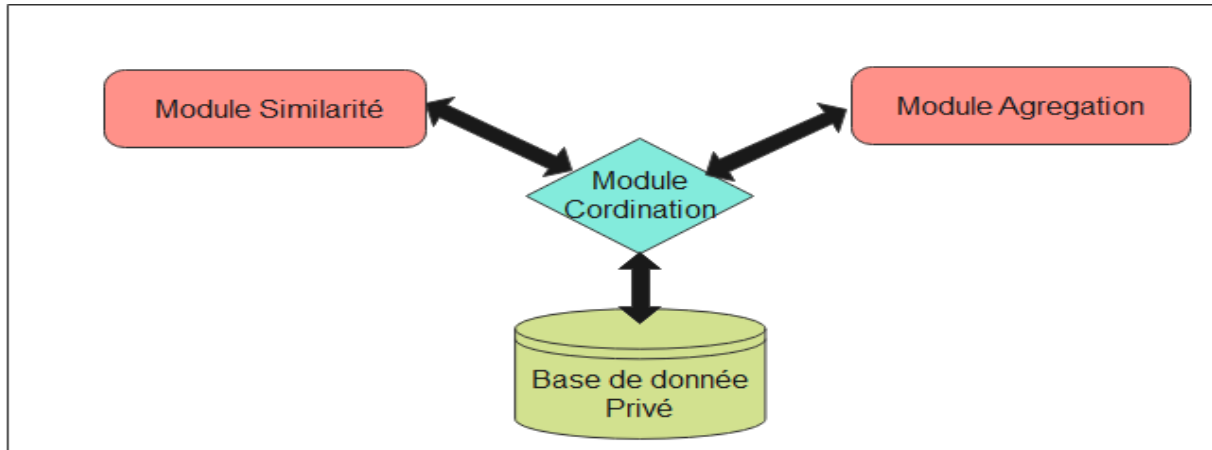


Figure14 : la structure interne de l'agent Analyseur.

Le diagramme d'UML d'activité de cet important agent dans notre architecture est le suivant :

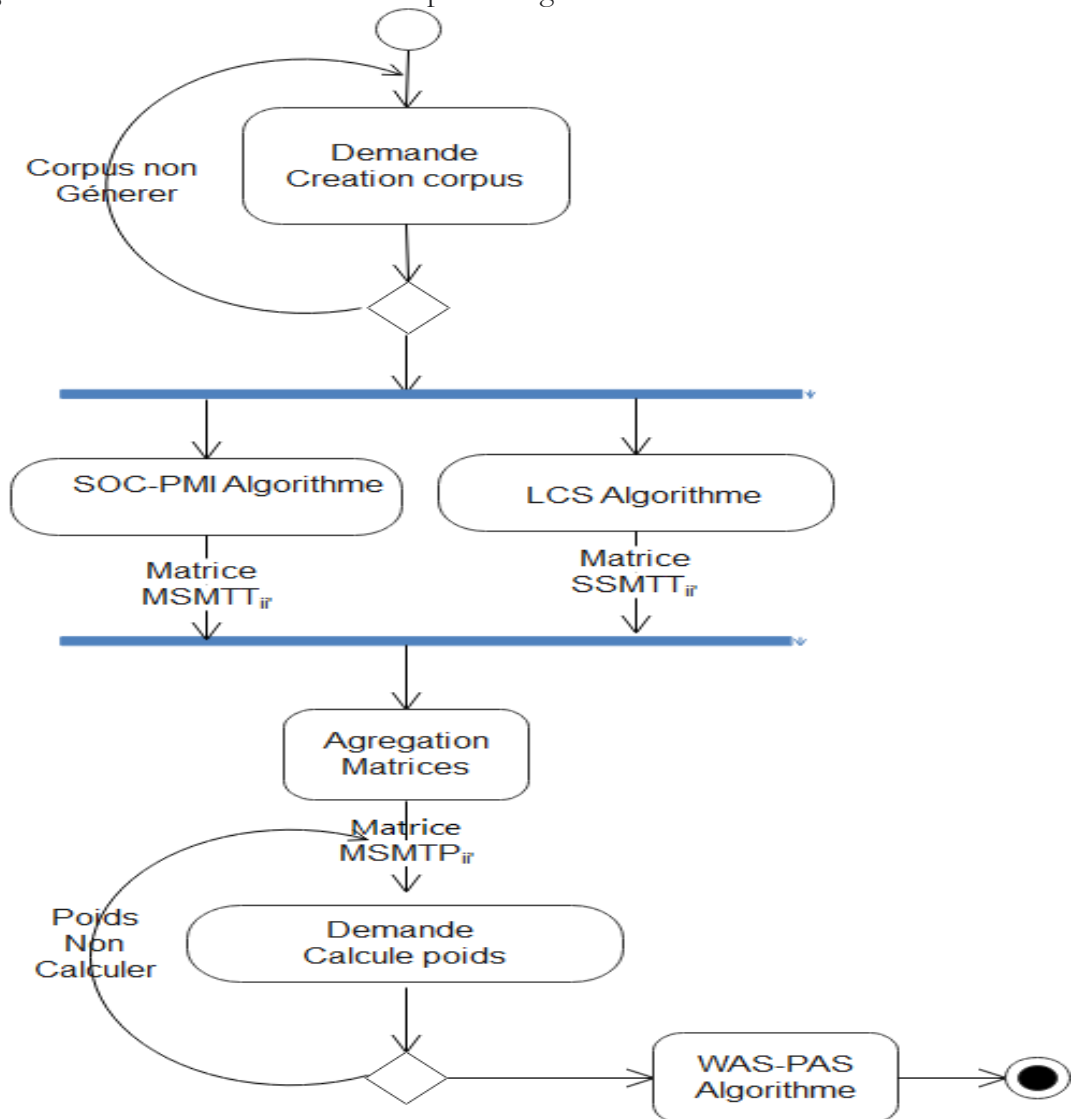


Figure15 : Diagramme d'UML d'activité.

2.2.2 Agent Solution

Il participe avec l'agent expert dans les opérations de recherche et de sélection de toutes les solutions intéressantes via la consultation du registre **UDDI (Universal Description Discovery and Intégration)** ainsi que les différentes bases de connaissances pour trouver les nouvelles solutions publiées face à un problème précis. Le module recherche qui est le module le plus important de cet agent vise aussi à chercher les problèmes candidats avec leurs solutions pour la création de corpus.

2.2.3 Agent réunion

La tâche principale de cet agent est de préparer les sessions de collaboration dans le cas où aucune solution n'a été trouvée à partir de différentes sources de données également lorsque le problème est considéré comme nouvellement rencontré ou mal exprimé. Cet agent communique en permanence avec l'agent expert via son module réunion afin d'avoir une information réelle sur la disponibilité des experts et ceci dans le cas où aucune solution automatique n'a été sélectionnée.

Enfin, nous complétons la définition de nos agents en donnant les fonctionnalités de certains modules qui remplissent la structure globale de tous ces agents comme des composants communs par exemple la base de connaissance privée et le module de coordination :

- ✓ **Base de connaissances privée (PKB):** Elle contient toutes les connaissances concernant un agent particulier, par exemple les algorithmes, les paramètres de fonction et les index sur l'ontologie du domaine partagée entre tous les agents du système. Il n'est donc pas nécessaire que tous les agents partagent les mêmes informations.
- ✓ **Module de coordination (CM):** Ce module joue un rôle important pour contrôler l'ensemble du mécanisme de prise de décision en coordonnant tous les agents qui constituent les couches arborescentes de notre architecture afin d'atteindre l'objectif visé.

2.2.4 La communication entre les Agents

Deux diagrammes de séquence ont été présentés ci-dessous afin d'expliquer au mieux l'enchaînement des flux décrits dans le Tableau 6 et permettant ainsi la description de l'interaction des éléments du système. Nous considérons pour une meilleure illustration deux scénarios comme suit :

Scénario 1 : Il s'agit d'un problème diagnostiqué avec succès et la solution est trouvée automatiquement et jugée satisfaisante

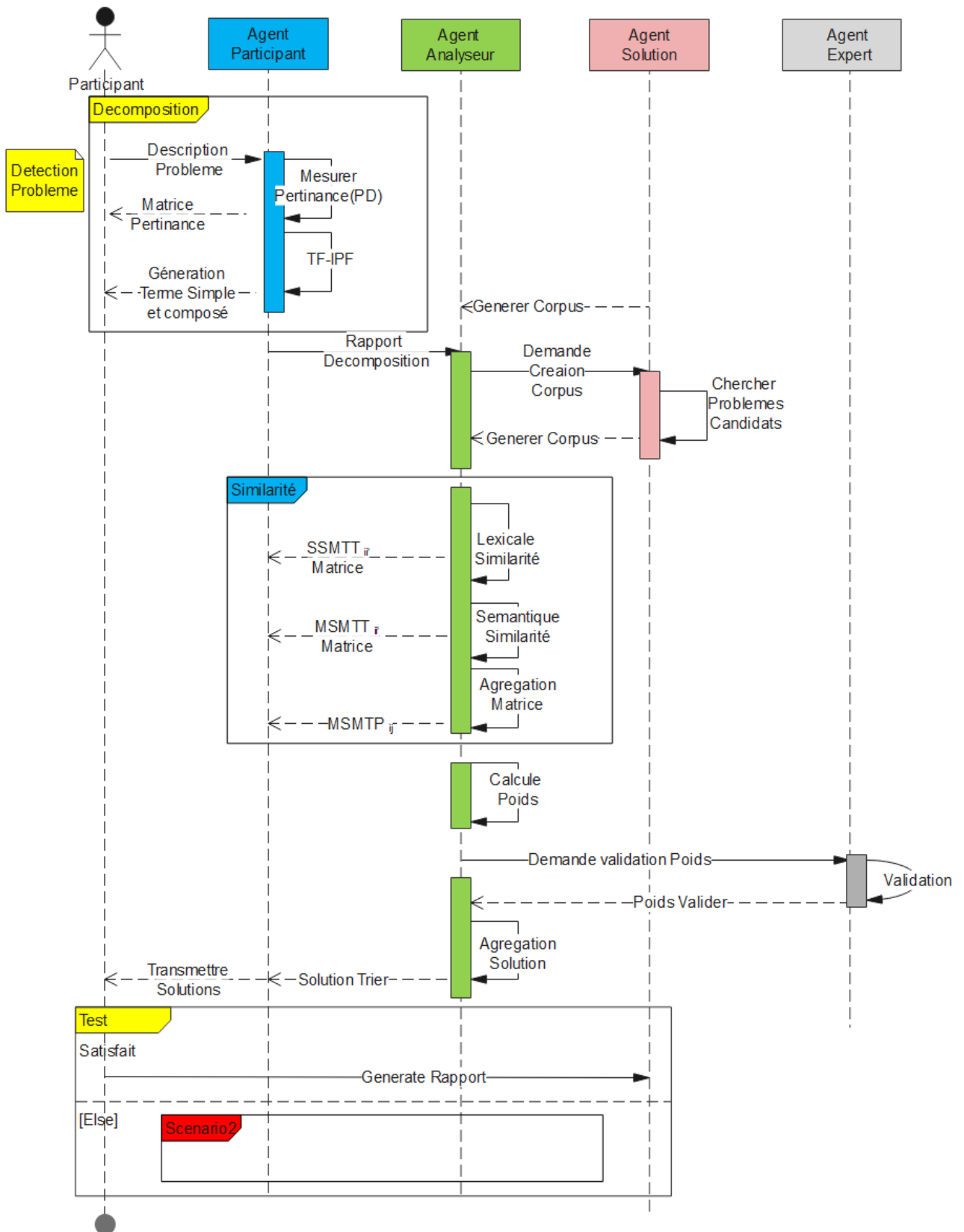


Figure16 : Diagramme de Séquence de MAITD(Scénario1).

Scénario2 : Solution non satisfaisante ou bien problème non diagnostiqué.

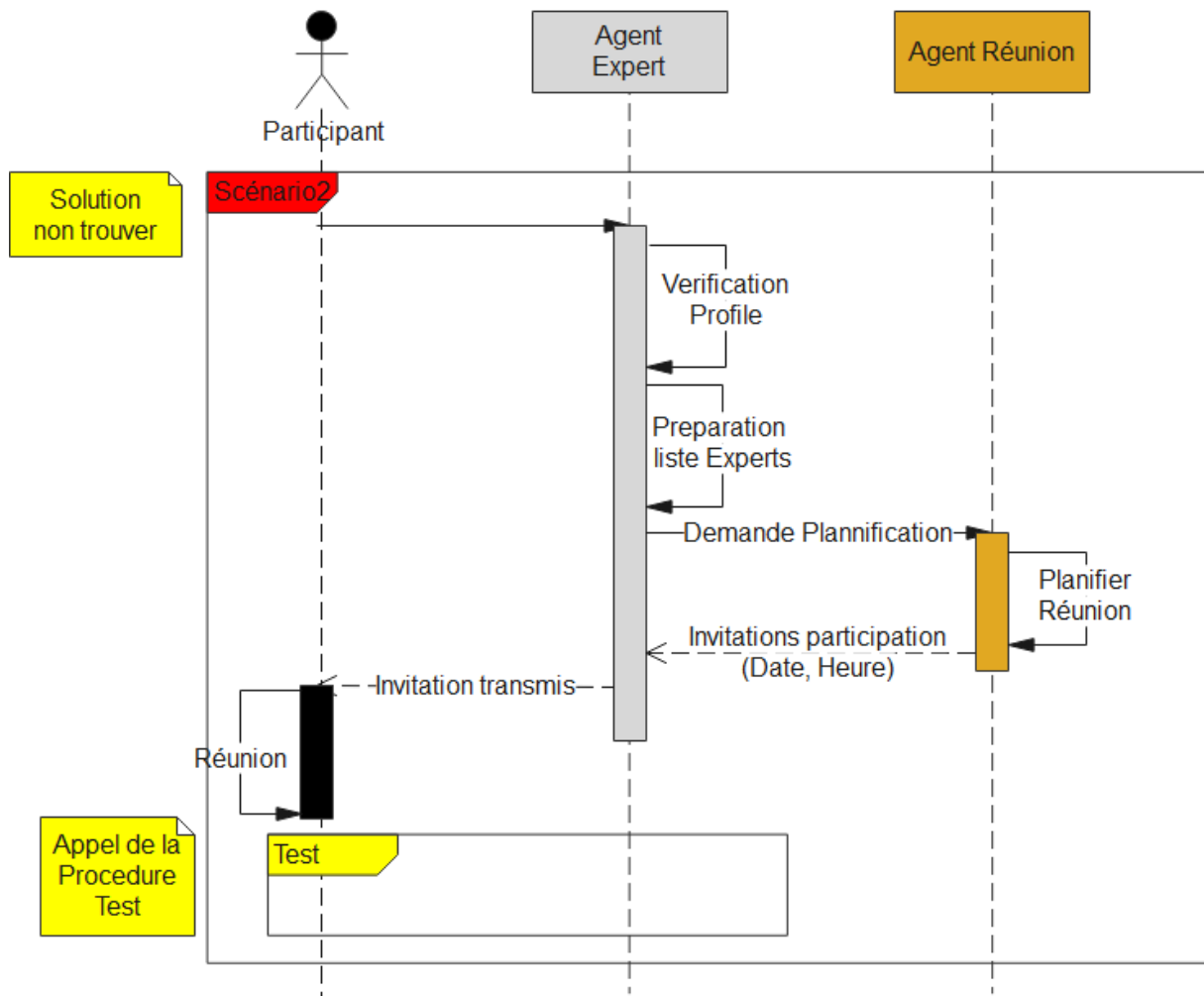


Figure17 : Diagramme de Séquence de MAITD (Scénario2).

2.3 La couche de donnée

Cette couche contient toutes les données que nous voulons traiter avec notre système selon des différents scénarios tels que l'ontologie de domaine, le registre UDDI, le dictionnaire Tree Tagger et la base de connaissances qui sont décrits dans ce qui suit:

2.3.1 Tree Tagger Dictionary: C'est un étiqueteur pour annoter du texte, il fournit la catégorie grammaticale et les informations de lemme pour regrouper les différents mots de la même famille. Dans notre approche, il est utilisé lors de la tâche de prétraitement des données.

2.3.2 Base de connaissances: Il s'agit d'un référentiel d'informations centralisé, qui comprend les différentes interventions notamment les scénarios nécessaires à la conception de notre corpus. En outre, il est utilisé comme source initiale des solutions. Il est considéré comme la base de données partagée avec tous les agents.

2.3.3 UDDI: la description, la découverte et l'intégration universelles est un service d'annuaire qui gère les informations sur les fournisseurs de services. Il s'agit d'un emplacement central où les clients/agents peuvent découvrir de manière dynamique les différentes solutions stockées et publiées par l'ensemble des organisations. De plus, chaque solution sur UDDI est définie par sa description WSDL (Web Service Description Language), celle-ci est basée sur la notation XML pour décrire les services Web correspondants afin qu'elle contienne toutes les informations nécessaires dont tous clients auraient besoin pour utiliser le service Web concerné. Comme cité dans [Web Service,2021], la structure générale d'un WSDL est décrite ci-dessous dans la Figure 18:

```

<!--WSDLStructure-->
<Definitions
Name = IT Service
TargetNameSpace = UDDI.org
  <!--AbstractDefinition-->
    <Types>
      <Message> ...
      <PortType> ...
  <!--ConcreteDefinition-->
  <Binding> ...
  <Service> ...
</Definitions>

```

Figure18 : La structure générale d'un WSDL[WSDL, 2021].

La définition des balises apparues dans cette structure de service web et résumée dans le Tableau5:

Mots clés	Descriptions
<Types>	Fournit des informations sur les données apparues dans le fichier WSDL(élémentsXML)
<Message>	Décrit le protocole spécifique et le format de données approprié des solutions et permet de définir un message échangé entre service Web.
<PortType>	Un ensemble d'opérations supportées par le fournisseur de solution. Il est utilisé pour combiner plusieurs messages dans une seule opération.
<Binding>	Décrit le protocolespécifique et format de donnée approprié des solutions. Il définit exactement comment chaque opération prendra place sur le réseau.
<Service>	Définit l'adresse IP et les ports correspondant du fournisseur. Il indique d'où les services sont accessibles.

Tableau5 : La description des balises d'un service Web [WSDL,2021]

2.3.4 Ontologie de domaine: Dans notre thèse, elle représente et reprend tous les concepts pertinents du domaine informatique. En conséquence, elle est développée pour générer un concept similaire à nos termes pertinents apparus dans un problème posé. Elle permet aussi de calculer les poids attribués pour chaque critère de notre matrice de performance décisionnel associé aux différentes solutions. Cette ontologie nous permet de générer un ensemble de termes multi-clés (MKT) et un terme clé spécifié (SKT) dans le but d'améliorer la qualité de l'extraction des termes tels que définis dans [M. Romero et al,2012]. Les auteurs de ce dernier travail combinent entre deux approches basées sur le thésaurus (Wikipédia) pour détecter les termes composés et un dictionnaire contrôlé afin de définir les termes simples. La Figure19 est une mise à jour de l'architecture décrite dans [s.elandaloussi et al, 2019] avec quelques flux numérotés pour illustrer le séquençage des actions et les relations entre tous les composants.

Notre architecture proposée [S.elandaloussi et al, 2021] permet une maintenance plus facile dans laquelle les rôles, la planification des interventions et la relation entre agents sont précisés au préalable. En effet, elle permet de suivre le cycle de vie du problème posé. Cependant, dans le cas où un problème est transporté dans toutes les couches décrites précédemment (couche présentation, couche interprétation et couche de donnée), il sera considéré comme un problème résolu. De plus, un maximum de fiabilité et d'autonomie sont données aux agents qui nous permettent de mettre à jour les composants d'une couche sans pouvoir impacter les autres couches.

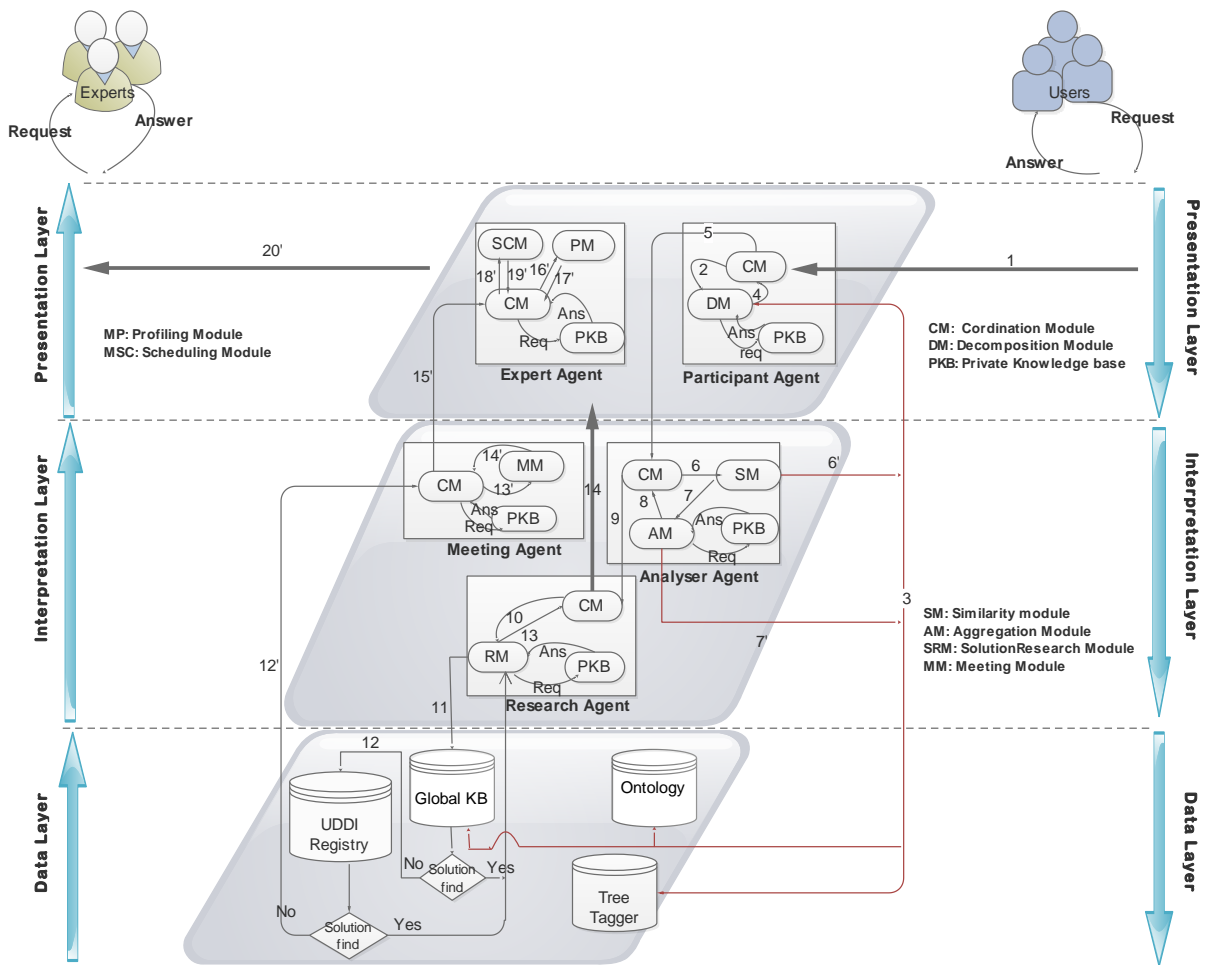


Figure19 : Architecture multi couche à base d'agent [s.elandaloussi et al, 2021].

Pour plus d'informations sur les flux numérotés apparus dans notre architecture, une brève description est donnée dans le tableau suivant (Tableau6):

Numéro de flux	Description	Numéro de flux	Description
1	Reception du problème	12	Interroger UDDI si aucune solution n'a été trouvée dans la base de connaissance globale.
2	Decomposition du problème	13	Rapport avant la coordination
3	Interroger les sources de donnée externe	14	Transmission du rapport au participant
4	Rapport de la décomposition	15	Informer l'agent Expert si aucune solution n'a été trouvée
5	Coordonner avec l'agent analyseur	16	Vérification des profils Experts
6	Mesure de la similarité	17	Vérification de la disponibilité des Experts
7	Solution d'aggregation	18	Generer une liste initiale des experts
8	Rapport d'aggregation	19	Generer une liste finale des expert
9	Coordonner l'agent recherche	20	Preparer une reunion
10	Chercher le corpus ou bien les solutions	20'	Informer l'expert et le participant
11	Interroger la base de connaissance globale		

Tableau6 : Description des flux d'informations de MAITD [S.elandaloussi et al, 2021].

3. Architecture à base de modules

L'architecture présentée à la Figure 21 est basée sur une suite de modules, elle est structurée en quatre étapes pour le traitement d'information, chacune d'elle inclut une liste des fonctions nécessaires pour atteindre l'objectif globale prédéfini du système. Comme le montre la Figure 20 d'où elle considère que le problème de dépannage comprend une suite d'étapes. Ci-dessous une brève description de ces étapes et qui seront présentées en détail dans les sections 5 et 6.

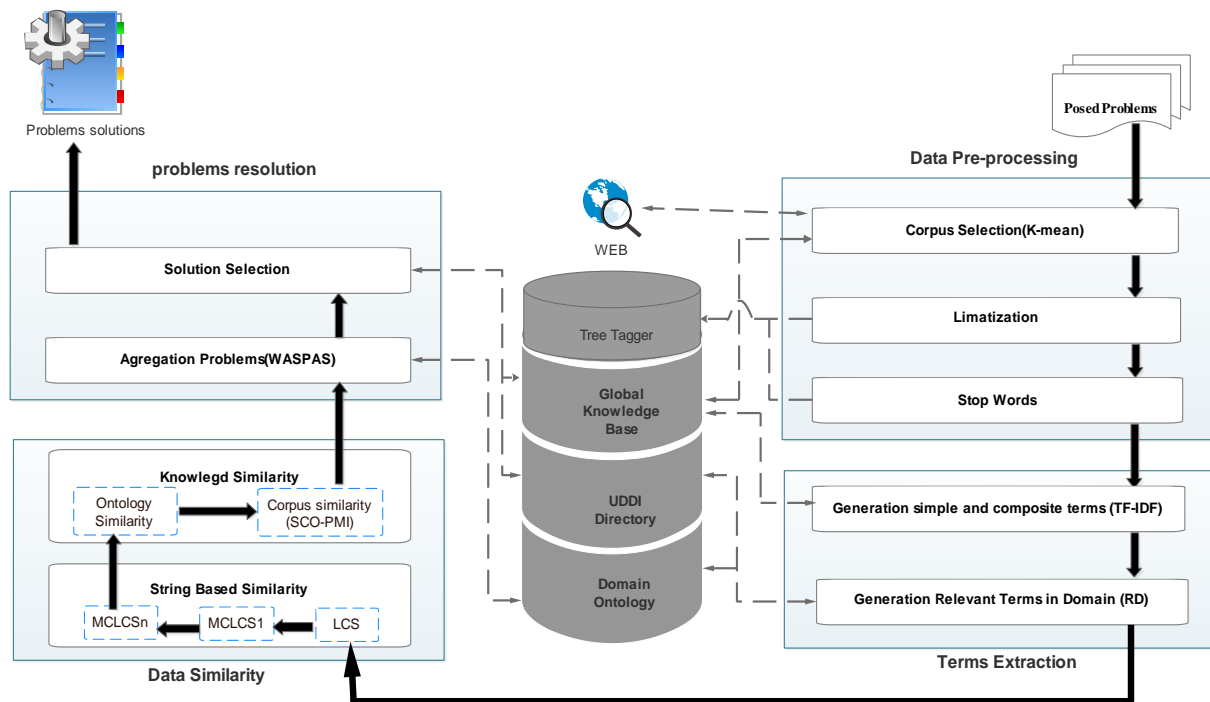


Figure20 : l'architecture MAITD-2 a base de module [s.elandaloussi et al, 2021].

Étape 1: prétraitement des données

Il est d'abord nécessaire de traiter efficacement le problème avant de mettre en œuvre une méthode. Par conséquent, pour chaque problème, nous commençons par sélectionner un corpus dans une base de connaissances globale et / ou le web afin de construire notre ressource externe en appliquant une méthode de regroupement basé sur l'algorithme K-means. Ce corpus est essentiel pour le bon déroulement des étapes 1 et 2 qui seront définies ici. De plus, une phase de lemmatisation est lancée en utilisant l'outil Tree Tagger pour obtenir une forme canonique de mots de la même famille, par exemple, le nom, le pluriel, l'infinitif du verbe...etc. Enfin, nous avons généré une liste de mots vides en éliminant «**les mots vides**», la «**ponctuation**» et les «**retours chariot**».

Etape2 : Extraction des termes

Dans notre travail, la tâche d'extraction automatique des termes clés vise à extraire les mots les plus représentatifs des problèmes saisis. Dans l'étape actuelle, nous évaluons les mots importants qui apparaissent ensemble dans un corpus centré sur l'algorithme **TF-IPF (Term Frequency–Inverse Problems Frequency)** qui sera présenté dans la section 4.2 de ce chapitre. De plus, un algorithme qui calcule la pertinence dans le domaine sera défini par la suite dans la section 4.1. Il est mis en œuvre pour extraire les mots pertinents en interrogeant l'ontologie du domaine et le corpus de données généré auparavant.

Etape 3 : La similarité des données

La mesure de similarité quantifie la relation entre deux objets. En effet, cette étape est divisée en deux phases. Dans notre travail, la première phase consiste à éliminer les erreurs apparues après une mauvaise identification du problème en combinant trois types d'algorithmes, ce qui est bien expliqué dans la section 5.1. La deuxième phase est basée sur deux concepts. Premièrement, une similarité à base d'ontologie afin d'obtenir des relations sémantiques. Deuxièmement, une similarité à base de corpus implémenté sur des informations obtenues à partir de grands volumes de données. L'objectif de cette deuxième phase est de déterminer la similitude entre des mots qui ne cohabitent pas fréquemment par l'implémentation de l'algorithme **SCO-CMI (Second Order Co-occurrence Pointwise Mutual Information)** qui sera défini dans la section 5.2.

Etape4 : La résolution du problème

Pour obtenir un ensemble de solutions pertinentes liées au problème étudié, nous commençons par l'agrégation d'une solution générée en implémentant l'algorithme **WASPAS(Weighted Aggregated Sum Product Assessment)** avant de sélectionner la solution à partir d'un UDDI.

4. Extraction des Termes

Parmi toutes les différentes problématiques traitées dans notre thèse. La présente section, vise à répondre à la question suivante : « comment filtrer et traiter efficacement les problèmes entrants dans le but d'extraire les termes pertinents qui devraient couvrir l'ensemble des problèmes décrits par l'utilisateur ? ». De ce fait, nous avons proposé une méthode hybride pour obtenir des bons résultats qui regroupe deux approches statistiques et une autre basée sur le regroupement. Premièrement, nous combinons une approche statistique avec une autre sémantique. Cela nous a amené à calculer la pertinence des termes dans notre ontologie du domaine ou bien dans le corpus initial qui contient une série de problèmes similaires. Ce corpus a été construit automatiquement en utilisant l'algorithme K-means comme méthode de regroupement (clustering), mais la manière de le faire n'est pas précisée dans notre travail de thèse, dans lequel l'ensemble de données généré est essentiellement axé sur le web et également à partir des interventions archivées. Deuxièmement, la fréquence d'un mot est calculée comme seconde approche statistique utilisant l'algorithme TF-IPF sur un corpus prédéfini afin de générer des termes simples et composés. Les auteurs dans [J.Mothe et F.Ramiandrisoa,2016] ont fait une enquête analysant toutes ces méthodes. Nous dérivons les clés essentielles suivantes :

a. La pertinence dans le domaine

Pour calculer la pertinence des termes dans un domaine, nous avons d'abord collecté des corpus dans plusieurs domaines via différentes sources de données selon les différents problèmes rencontrés autour de l'infrastructure IT (**Information Technology**). Nous avons focalisé notre travail sur la méthode de regroupement K-means. Cependant, les techniques décrites dans cette section combinent entre les approches statiques et sémantiques comme mentionné précédemment. Selon [P.Velardi et al, 2001] un score spécifique, appelé pertinence dans le domaine (DR), a été défini. Plus précisément, étant donné un ensemble de n domaines (D1...

Dn). La pertinence de domaine d'un terme T est calculée pour obtenir des termes pertinents (PT) en utilisant la formule25 suivante :

$$PD(T_i, D_i) = \frac{P(T_i, D_i)}{\sum_{i=1}^n P(T_i, D_i)} \dots \dots \dots (25)$$

d'oùP(T_i,D_i) qui signifié les probabilités conditionnelles du terme T_i sur le domaine D_i et qui est estimée comme :

$$P(T_i, D_i) = \frac{freq(T_i \text{ in } D_i)}{\sum_{i=1}^n freq(T_i, D_i)} \dots \dots \dots (26)$$

b. La fréquence des termes TF-IPF

Le score **TF×IDF** nommé **TF.IPF** (Term Frequency–Inverse Problems Frequency) dans notre enquête est mis en œuvre pour sélectionner le type de terme composé ou bien simple dans le but de comparer la fréquence des mots composés, qui sont incorporés dans un problème particulier avec la fréquence de ce bloc dans un corpus. Ce corpus est représenté par un ensemble de problèmes similaires au problème défini. Ces problèmes similaires sont générés automatiquement via le web ou bien par des interventions passés en utilisant une méthode de regroupement K-mean. Le score TF.IPF pour les termes composés dans les problèmes est donné par la formule27. :

$$TF - IPF(T, T') = \frac{nb(T, T')}{nb(T) * nb(T')} \dots \dots \dots (27)$$

Où

nb(T,T') est le nombre de fois dont T et T' apparaissent consécutivement sur le corpus, **nb(T) et nb(T')** comptent le nombre de fois où T et T'apparaissent sur le même corpus.

5. La mesure de similarité textuelle

La mesure de similarité entre les documents textuels est l'un des enjeux importants dans plusieurs disciplines telles que la classification de texte, l'analyse de données de texte [Berkhin,2006], l'extraction de connaissances à partir de données textuelles (**TextMining**) ou de récupération d'informations [O.Medelyan, I .Witten, 2008], le regroupement de documents [Berkhin,2006]. De plus, trouver des similitudes entre deux textes consiste à comparer les mots de ceux-ci. Les mots peuvent être similaires de deux manières : lexicalement et sémantiquement. D'une part, les mots sont similaires lexicalement s'ils ont une séquence de caractères similaire. En revanche, les mots sont sémantiquement similaires s'ils ont été utilisés dans le même contexte [A. islam, D. Inkpen,2008].

5.1 Mesure de similarité lexicale(String bases similarity measure)

Cette méthode ne nécessite pas de comprendre le vocabulaire ou la grammaire de la langue du texte. Une variété d'algorithmes comme a été décrit dans le chapitre3 de cette thèse: **Longest Common Substring (LCS)**, **Damerau-Levenshtein**, **N-gram NGD**, **Cosine Similarity**, **Euclidian Distance**, **Jacquard Similarity**, **PMI-IR**, **SCO-PMI**, **HAL**, **LSA**, **GLSA**, **CL-ESA**

...etc. Afin d'estimer et d'éliminer une mauvaise description du problème, nous avons utilisé comme similarité lexicale une approche basée sur les chaînes en déployant trois versions modifiées de l'algorithme LCS (**Longest Common Subsequence**). En effet, la similarité à base de chaîne entre un problème posé P_i et un problème candidats PC_j est estimée en calculant la matrice de données correspondante $SSMTT'_{ij}$ comme le montre le Tableau7. Cette dernière contient c colonnes correspondant au T_i termes pertinents de P_i et r lignes pour ceux des termes T'_j des différents PC_j . De plus, $ssmtt'_{ij}$ représente les mesures de similarité sémantique correspondantes. Ainsi, la similarité à base de chaîne entre deux termes est calculée avec une fonction qui est définie comme suit (Formule28):

$$SSmtt'_{ij} = \frac{\text{lenght LCS}(T_i, T'_i)^2}{3 \text{lenght}(T_i) * \text{lenght}(T'_i)} + \frac{\text{lenght MCLCS1}(T_i, T'_i)^2}{3 \text{lenght}(T_i) * \text{lenght}(T'_i)} + \frac{\text{lenght MCLCSn}(T_i, T'_i)^2}{3 \text{lenght}(T_i) * \text{lenght}(T'_i)} \dots \quad (28)$$

Où LCS signifie la sous-séquence commune la plus longue et qui n'est pas obligatoirement consécutive et MCLCS1 est la sous-séquence commune la plus longue commençant au premier caractère et qui a un sens consécutif. Exemple : la sous-sequence MCLCS1 entre les deux concepts Process et processeur est « **process** ». Enfin MCLCSn signifiait la plus longue commune sous-séquence commençant à n'importe quel caractère n avec un sens consécutif. Le tableau7 est un exemple de calcul de similarité entre les termes de deux problèmes différents.

Terme T_i \ Terme T'_j	démarrage	machine	interrompu
impossible	0.19	0.22	0.41
accéder	0.18	0.2	0.12
Système exploitation	0.29	0.27	0.43
Machine virtuelle	0.17	0.79	0.31

Tableau7 : Exemple d'une matrice de similarité à base de chaîne entre deux problèmes.

5.2 La mesure de similarité sémantique

Pour assurer la similarité sémantique de notre cas d'étude, et par rapport aux divers algorithmes apparus. On peut définir deux principales approches pour ces algorithmes dans la conception de notre système :

5.2.1 Approche à base de Corpus

Principalement, nous prenons maintenant la similarité sémantique, dans laquelle nous avons calculé les valeurs de similarité **SOC-PMI** (Second Order Co-occurrence Pointwise Mutual Information) [M. Aminul, D. Inkpen, 2006] en utilisant notre corpus comme source de fréquences et de contextes. Notre algorithme est utilisé pour construire la matrice de similarité sémantique $n * m$ entre les termes T'_1 et T'_i qui correspondent respectivement à PP_i

(i ème Problème Posé) et PC_j (j ème Problème Candidat) respectivement, à savoir $MSMT_{ii}$ comme le montre le Tableau 8 suivant :

Terme i \ Terme j	démarrage	machine	interrompu
impossible	0.72	0.42	0.57
accéder	0.64	0.41	0.23
Système exploitation	0.63	0.74	0.7
Machine virtuelle	0.81	0.91	0.69

Tableau 8 : Exemple d'une matrice de similarité sémantique à base de Corpus.

5.2.2 Approche à base de connaissance

Les approches de similarité des mots basées sur la connaissance sont concentrées sur un réseau sémantique de mots, certainement, la structure de la base de données est une taxonomie dans laquelle chaque nœud est un concept tel que WordNet [C. Fellbaum, 1998]. Comme c'est défini auparavant de nombreux algorithmes sont développés en fonction de ce contexte tel que : RES, LIN, JCN, LCH, PATH, WUP.

5.2.3 Approche Hybride

Inversement, les méthodes hybrides utilisent à la fois des mesures basées sur le corpus et des mesures basées sur les connaissances pour déterminer la similarité du texte. Plusieurs recherches sont réalisées dans ce sens comme les travaux décrits dans [R. Mihalcea et al, 2006], [L.Y. Mclean et al, 2006]. De plus, deux approches sémantiques à base d'un corpus et la seconde à base de connaissances à l'aide de l'algorithme SOC-PMI sont utilisées pour calculer la similarité entre deux mots qui ne coexistent pas fréquemment car ils coexistent avec les mêmes mots voisins, également en intégrant notre ontologie de domaine comme base de données sémantique. En effet, la fusion entre ontologie et corpus est l'un des aspects les plus importants de notre approche, de tel sorte l'ontologie nous a permis de calculer le poids de chaque concept et le corpus est très important dans le calcul de la similarité sémantique.

6. Agrégation des solutions

La fonction d'agrégation produit un calcul spécifique sur un ensemble de valeurs et renvoie une valeur unique. En effet, au moment de construire l'ensemble des matrices $MSMT_{ii}$ correspondant aux différents problèmes candidats face à un i ème problème posé, nous regroupons toutes les matrices $MSMT_{ii}$ dans une même matrice $MSMT_{Pij}$ (semantic Similarity Matrix Term to Problem) pour obtenir notre table de performance avec n colonnes (termes pertinents de P_i) et r lignes (problèmes similaires à P_j). Comme le montre l'algorithme suivant :

L'algorithme de similarité termes problèmes

```

Input:  $T_i, T'_i, MSMTT_{ii}^*$ 
Output:  $MSMTP_{ij}$  "Matrixe desimilarité semantiquetermes  $T_i$  et problem  $P_j$ "
  For  $i \leftarrow 1$  to  $r$  do "  $c$  est le numéro de colonne de  $MSMTT_{ii}$  "
    For  $j \leftarrow 1$  to  $k$  do "  $r$  c'est le nombre de ligne de  $MSMTT_{ii}$  "
       $msmtp_{ij} \leftarrow \max(msmtt_{ij})$ 
    End
  End
End
  
```

Afin de synthétiser la performance sémantique en un score global comme $\text{sim}(P_i, P_j)$ dans le but d'extraire la plupart des problèmes similaires. Plusieurs fonctions d'agrégation sont apparues, MOORA, COPRAS, TOPSIS, WASPAS, EDAS...., telles qu'elles sont définies dans le chapitre précédent. Dans l'implémentation de notre thèse, nous utilisons la méthode WASPAS (Weighted Aggregated Sum Product Assessment), l'une des méthodes de prise de décision multicritères les plus récemment développées, conçue pour résoudre des problèmes quantitatifs. De plus, elle est simple à mettre en œuvre et elle donne en général des résultats satisfaisants. Au final, le problème est considéré comme mal exprimé ou non rencontré, si cette fonction donne une faible valeur d'agrégation. A ce niveau la les agents Réunion interviennent donc pour préparer les sessions de collaboration. Cependant, cette méthode combine la somme pondérée et le produit comme mentionné dans la formule 29.

$$A_j = 0.5 \sum_{i=1}^c (msmtp_{ij}) * w_i + 0.5 \prod_{i=1}^c (msmtp_{ij})^{w_i} \dots \dots \dots (29)$$

Où

w_i correspond au poids du critère i et A_j comme une agrégation d'alternatives j (problèmes candidats).

7. Conclusion

Dans ce chapitre, nous avons présenté notre approche du MAITD. L'objectif de cette proposition est de pouvoir développer une plateforme générique de diagnostic collaborative basée sur une architecture orientée services par l'intégration d'un ensemble d'agents. La notion d'agent a rendu la réalisation de notre système moins complexe en divisant les tâches entre ces unités autonomes. Chaque agent réalise un ensemble de tâches en se basant sur des techniques et des algorithmes différents qui sont considérés comme des services. Ces services sont stockés dans un UDDI pour faciliter la tâche de maintenance et de mise à jour.

Un des défis de notre approche est le diagnostic non supervisé de la requête des participants même s'ils ont des niveaux de compétences différents. Contrairement à d'autres contributions qui se focalisent sur la stratégie de recherche de la meilleure solution sans prendre en considération la compétence du participant comme facteur critique dans la satisfaction de résultat. Cette approche traite l'information entrée différemment par l'intégration des techniques d'extractions et de similarité récente et adaptable à notre contexte d'étude dans le but de générer des recommandations de diagnostic adaptés. De plus, l'agrégation de ces solutions a été prise en compte dans notre approche avant la sélection du service préconisé comme solution pertinente et qui a été jugé comme meilleure alternative par notre système.

Pour motiver nos travaux de recherche d'un point de vue pratique, le chapitre suivant décrit l'implémentation de notre approche.

Chapitre5

Expérimentation & Validation

Plan du chapitre5

1. Introduction
2. La description d'un problème
3. Prétraitement de donnée
4. La pertinence dans le domaine
5. La Composition des termes
6. Mesure de similarité lexical
7. Mesure de similaritésémantique
 - 7.1 Approche a base de Corpus
 - 7.2 Construction de notre Ontologie du Domaine
 - 7.3 Enrichissement de notre ontology
 - 7.4 Détermination du poids de conceptdans une ontologie
8. Agrégation des solutions
9. Preparation d'une session de collaboration
10. Supervision de diagnostic du problème
11. Validation et evaluation des solutions
 - 11.1 La validation F-measure
 - 11.2 La méthodologie de validation
12. Conclusion

1. Introduction

Dans le précédent chapitre, nous avons proposé une architecture multi couche à base d'agents appliquée dans un cadre collaboratif semi supervisé, avec l'appui d'une approche multicritères centrée sur la fouille de texte et la décomposition des phrases.

Ce chapitre présente quelques interfaces du MAITD définies auparavant. Il s'agit d'un cadre expérimental où notre approche a été testée et évaluée. En effet, les fonctions et le déploiement multi-agents sont respectivement implémentés en utilisant l'environnement Java Net Beans et la plate-forme Jade pour la création et l'interaction des agents. De plus, le serveur WAMP pour le déploiement de bases de données, Tree Tagger comme étiqueteur de lemmatisation et les outils Protégé sont dédiés au développement d'ontologies de domaine. Dans notre approche, le corpus est créé à partir des problèmes réels de bases de données d'infrastructure informatique générés par des experts spécialisés.

2. La description d'un problème

La description d'un problème est la première étape de notre prototype conçu. Autrement dit, elle joue le rôle de déclencheur du déroulement de notre système. En effet, Après d'être connecté à sa session privée, l'utilisateur identifie son problème et le domaine correspondant comme la vue donnée dans la Figure22.

Figure21 : Identification du Problème.

3. Le prétraitement de données (Couche1)

L'interface donnée à la figure 22 est le résultat de l'étape 1 qui fait partie de la première couche (couche de présentation) exécutée par l'agent participant. Par exemple, pour un problème posé comme suit : «pas de démarrage de la machine virtuelle en mode Shell », le système génère 9 problèmes candidats numérotés dans la figure 22 de 1 à 9 en appliquant l'algorithme de regroupement K-means. De plus, nous présentons la liste initiale des termes pertinents après les tâches de lemmatisation et d'arrêt des mots. A ce stade de déroulement, notre système sélectionne et affiche les termes pertinents pour chaque problème candidat en utilisant les algorithmes TF-IPF et Pertinence dans le domaine (PD) présentés précédemment dans le chapitre4 (Voir Figure23 et Figure24). Par exemple, pour un problème candidat « I can't start my VMware virtual machine » qui est probablement similaire au problème initialement posé a comme termes pertinents : "Start, vmware, virtual, machine".

Num.	Problems	Pertinents Terms
1	I can't start my VMware virtual machine	start, vmware, virtual, machine,
2	Vmkernel service stop and no response from the virtual machine	vmkernel, service, stop, response, virtual, machine,
3	Vcenter vmware does not communicate with Esxi	vcenter, vmware, communicate, esxi,
4	I can't migrate my VMware virtual machine from one Esxi to another, problem with vmkernel	migrate, vmware, virtual, machine, esxi, problem, vmkernel,
5	No pool resources are assigned to this virtual machine	pool, resources, assigned, virtual, machine,
6	No increase of the memory capacity of the my virtual machine	increase, memory, capacity, virtual, machine,
7	No memory space available on esxi	memory, space, available, esxi,
8	Problem with vcenter and virtual machine	problem, vcenter, virtual, machine,
9	No start of virtual machine on schell mode	start, virtual, machine, schell, mode,

Figure22 : Prétraitement de donnée.

4. La pertinence dans le domaine

En pratique, selon les derniers travaux [P. Berkhin et al, 2006], le seuil de pertinence entre les termes dans un domaine a été estimé à $PDT(T_i, D_i) = 0,35$ pour être jugé « satisfaisant ». De sorte que T_i : représente le terme i et D_i : représente le domaine i .

La Figure23 donne un aperçu des différentes mesures générées par notre système autour d'un problème défini « I can't start my vmware virtual machine ».Cependant, le corpus généré par la méthode K-means contient 15 problèmes différents comme problèmes candidats provenant de 4 sources de données différentes (2 sites WEB, base des cas du système, UDDI). Le volume de mots inspiré de ce corpus correspond à 94 mots.

Designation	Volume words	Nb Problems	Nb Data Source
Virtualization	94	15	4

Terms T_i	PD (T_i, D_i)	Decision
Can't	0.29	No
Start	0.46	Yes
Vmware	0.82	Yes
Virtual	0.91	Yes
Machine	0.76	Yes

Figure23 : Pertinence dans le domaine.

Comme le montre la figure 23, l'ensemble de termes {Start,VMware,Virtuel, Machine} sont pertinents dans le domaine virtualisation. De plus, ils ont tous une valeur supérieure au seuil $\alpha=0,35$.

5. La Composition des termes

Cette étape est nécessaire pour différencier les mots simples des mots composés. Ainsi, si les termes T et T' apparaissent consécutivement ensemble très souvent nous allons considérer que T et T' construisent le même mot nommé $T T'$. Le seuil minimal de satisfaction est estimé à 0,7. Cette valeur est déduite après la simulation de 50 requêtes utilisant 12 Corpus différents. La Figure24 présente les représentations vectorielles obtenues après exécution de notre méthode TF.IPF pour détecter les mots simples et composés d'un problème posé ainsi que des informations statistiques sur notre corpus.

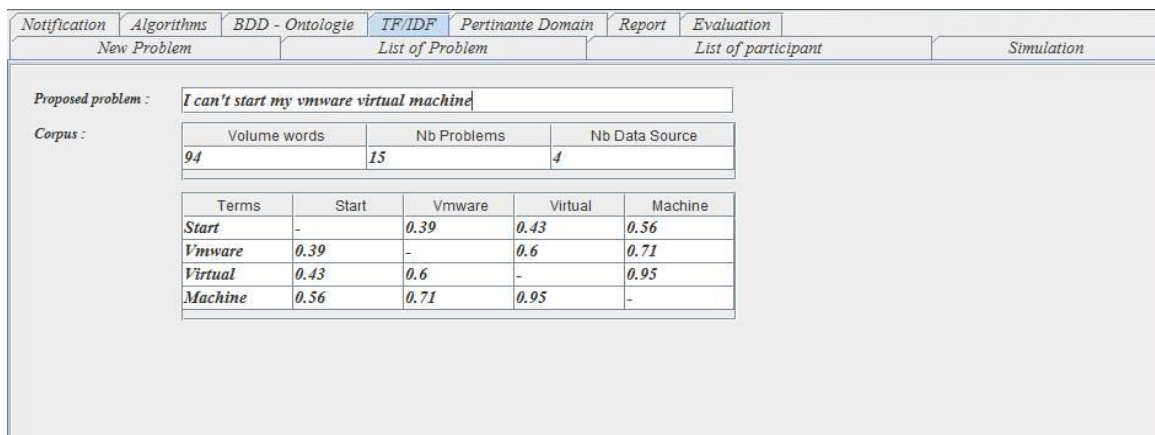


Figure24 : La Fréquence des Termes TF-IPF.

Après l'exécution de l'algorithme TF-IPF, l'ensemble des termes composites apparus dans le problème actuel posé est {Machine virtuelle} car la valeur TP-IPF entre les deux mots "Virtuel" et "Machine" a une valeur supérieure au seuil cité précédemment. De plus, l'ensemble de données de termes simples est composé de : {start, VMware}.

6. La mesure de similarité lexicale(String bases similarity measure)

Comme le montre la Figure 25, notre matrice $MSSTT''i'$ (Matrice de similarité statique entre termes T_i du problème posé et T_j du problème candidat similaire contient 4 colonnes {start, vmware, virtual, machine} et 4lignes : {Vmkernel,service,stop,response}. Par conséquent, les valeurs de similarité entre les termes de la matrice sont visualisées comme suit :

$msstt'\{vmkernel,start\}=0,0082,$
 $msstt'\{vmkernel,vmware\}=0,165,$
 $mstt'\{service,machine\}=0,0875.$

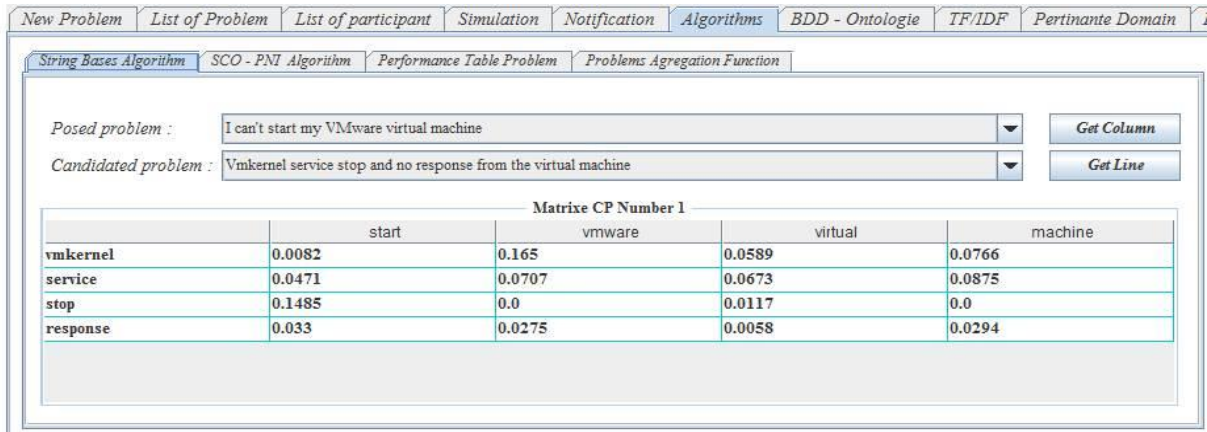


Figure25 : Mesure de similarité lexicale.

7. La Mesure de similarité sémantique

7.1 Approche à base de Corpus

Principalement, nous prenons maintenant la similarité sémantique, dans laquelle nous avons calculé les valeurs de similarité utilisant l’algorithme SOC-PMI (Second Order Co-occurrence Pointwise Mutual Information) défini dans [M. Aminul, D. Inkpen, 2006], en considérant notre corpus comme source de fréquences et de contextes. Notre algorithme est utilisé pour construire notre MSMIT_{ii} la matrice de similarité sémantique n*m entre les termes T_i et T_j qui correspondent respectivement au problème posé PP_i et problème candidat PC_j comme le montre la Figure 26.

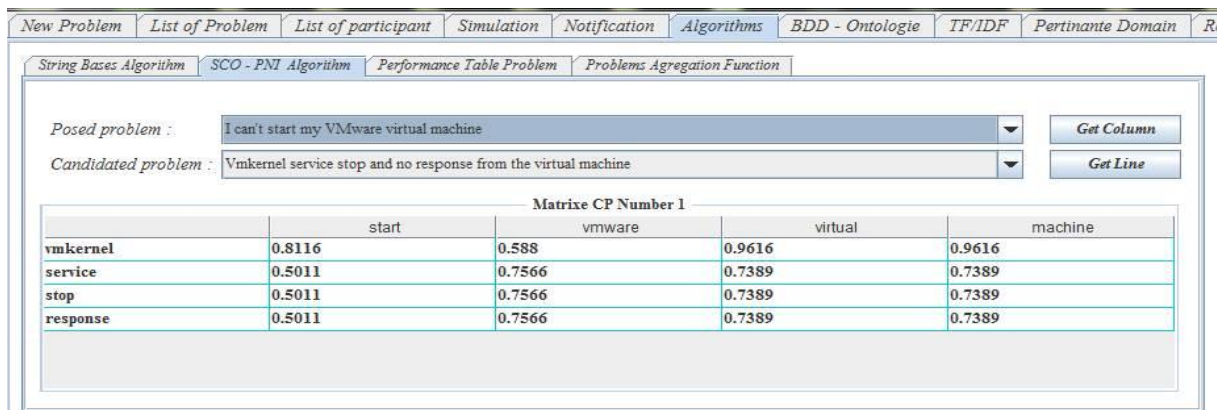


Figure26 : Mesure de similarité sémantique Termes a Termes.

Dans cette figure, les termes "vmkernel" et "virtual" sont sémantiquement similaires car ils coexistent souvent avec des mots en commun sur le même corpus.

7.2 Construction de notre Ontologie du Domaine :

La connaissance associée à notre domaine d'étude (Infrastructure IT) peut être représentée de façon plus formelle à travers une ontologie. En effet, pour une requête d'un utilisateur et en accédant à la connaissance de l'ontologie à partir de laquelle l'information d'un corpus a été indexée ce qui lui permet de spécifier ses similarités sémantiques et les lacunes de sa première description par rapport à l'information idéale qui est disponible. D'autre part, la représentation de l'information à partir d'une ontologie peut définir un vocabulaire contrôlé (termes et concepts) à partir duquel l'utilisateur spécifiera son besoin.

La construction d'une ontologie nécessite l'utilisation d'un processus permettant de passer des données brutes à l'ontologie. Dans notre contexte les données brutes sont présentées dans un tableau (Voit Tableau9) récapitulatif contenant l'ensemble de donnée semble importante à connaître dans une infrastructure IT. Ses connaissances ont été validées par des experts du domaine et peuvent être transformées selon le format décrit dans la figure

Domaine	Produits	Propriétés
Serveurs	Châssis(UCS)	Power/Temp/FAN
	Blade(UCS)	HDD(Statut RAID/capacity)
	F.interconnect(UCS)	Networking (trafic flow Uplink + SAN Link/ Power /FAN
	F.Extender	Networking (Link status entre F.ext et F.interconnect
Réseaux	ASR(Cisco)	Cisco IOS : Update, bug fixes, error log
	ASA(Firewall)	securityattacks
	AC	securityattacks, Networking
	Catalyst	security (access);
	QOS	
	Access Points	AP Stauts (up/down), # of connected users, uplink, trafic flow
Voip IP	UC 560	phone registration, PSTN lines ;
Virtualisation		
	vCenter	CPU/RAM/HDD Networking Service (DRS, HA, vMotion,...), ESX :

	VM	Status, resources, attached VMs Statut des ressources assignées (CPU / RAM (load), Network ports (link up/down), HDD,
Stockage	EMC	Networking HDD/Statut RAID
Applications	OS (Windows/Linux)	Windows service, LINUX ?????? Number of request per second, database I/O, log
	SQL	files, number of waiting times, Latency times, transact SQL, resolve problems
	AD	Performance, Security, Latency
	DNS	number of overall queries and responses processed by a DNS server
	ERP AAA VPN Syslog Server	Ressources, Database Statut du service ; # Sessions, Statut du service, # Session, Logs rotation (based on size and date),
Acces	Modem	Statut (UP/Down), updates, config. (cf ACS)
	AP Mesh	Status (UP/Down) / Trafics, Mesh interconnection
Liens	Bandwidth	Protocols, Bandwidth
Autres	Printer	Networking, Level Toner
	UPS	Interface, Batteries, Power,

Tableau 9 : Récapitulatif des concepts de diagnostic IT [s. elandaloussi, 2020].

La transformation de ces connaissances en classe et sous classe nous a permis d'établir les concepts affichés dans la Figure 27. Comme le montre cette dernière, notre ontologie est divisée en deux grands axes (HARDWARE et SOFTWARE). La classe HARDWARE est composée des sous-classes suivantes : Réseaux, Sécurité, Stockage, etc. La classe SOFTWARE regroupe tout ce qui est applicatif ;

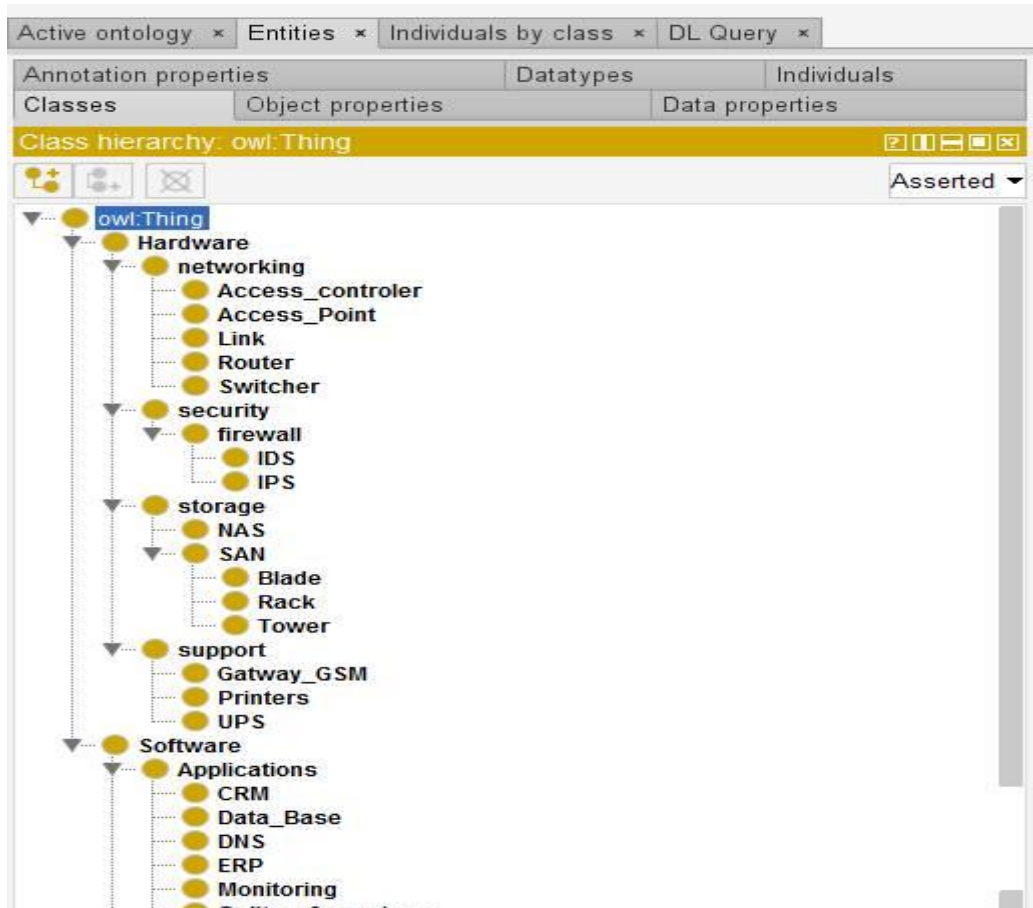


Figure27 : Définition des classes de notre ontologie initiale.

Après la définition de l'ensemble des classes et des sous classes. Une étape de déclaration de tous les individus qui font partie des domaines étudiés est nécessaire comme le montre la Figure 28. Par exemple l'individu Aradial appartient à la classe SOFTWARE et qui fait partie du domaine réseaux. De plus EMC est un équipement qui appartient la classe HARDWARE du domaine stockage.....etc.

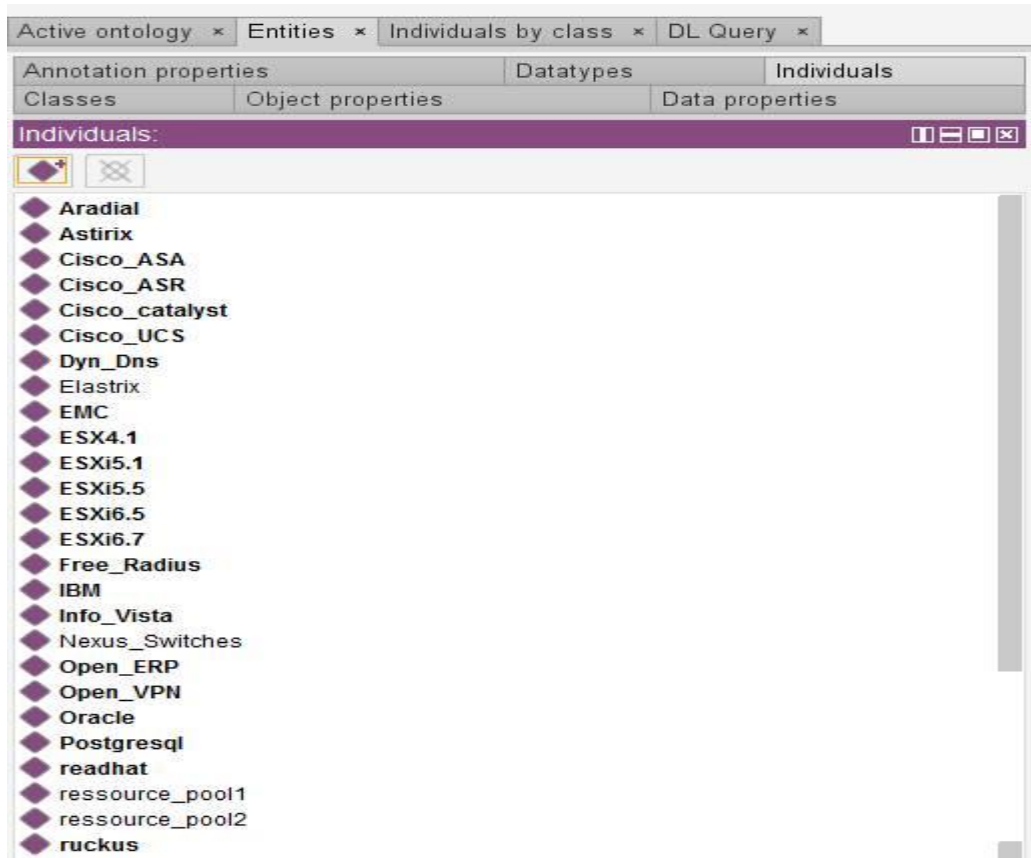


Figure28 : Les individus de notre ontologie initiale.

La déclaration de ces individus oblige d'avance la définition des propriétés pour chaque objet et chaque donnée comme le montrent les deux Figures29 et Figure30 respectivement.

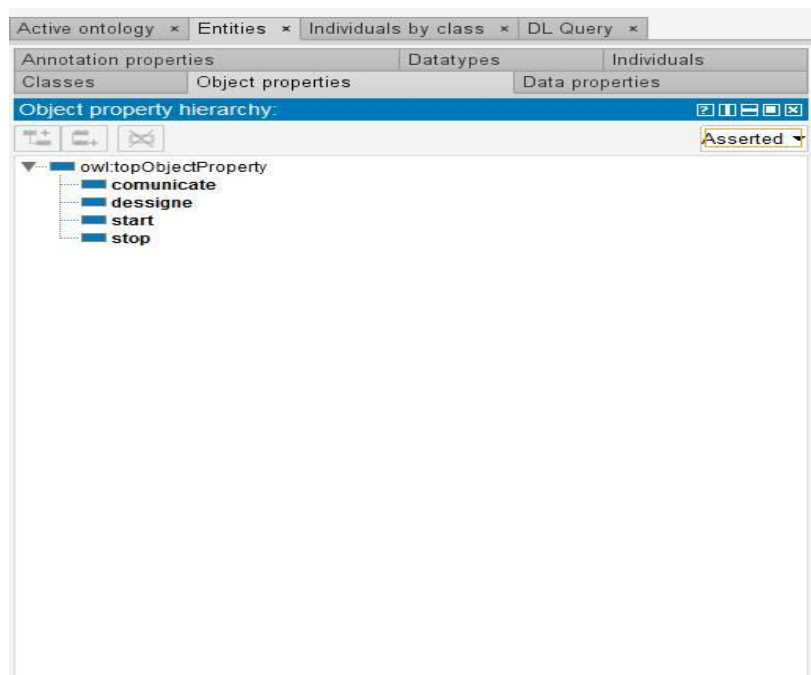


Figure29 : Propriété d'un objet de notre ontologie.

Les propriétés présentés dans cette vue (figure29) concernent l'objet "**machine**", par exemple : "**la machine communique avec le serveur**", "**la machine ne peut pas démarrer en mode Shell**"...etc.

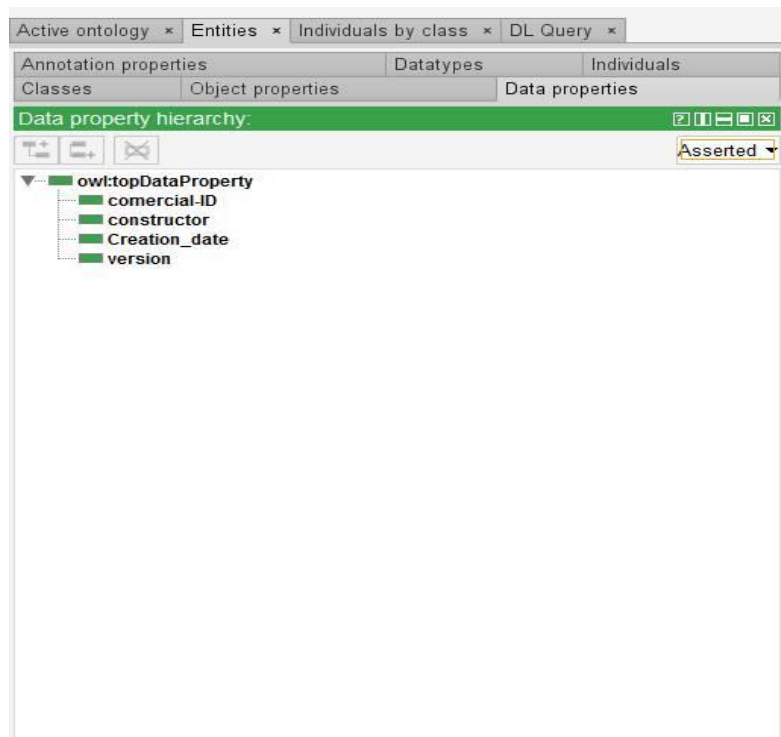


Figure30 : Propriété de donnée de notre ontologie.

Chaque individu (donnée) possède ses propres propriétés. Par exemple, pour un point d'accès "Ruckus" les informations suivantes sont générées :

Comercial-ID = **Ruckus**, Constructor= **Cisco**, Creation-date=**2011**, Version= **v3.11.2**

7.3 Enrichissement de notre ontologie

Après la phase de création de notre ontologie initiale. Les étapes les plus importantes pour l'enrichissement de notre ontologie de domaine sont :

- 1- Pour chaque terme T_i du corpus : extraction des concepts C_i relatifs à T_i par des calculs de similarité correspondante.
- 2- Comparaison des concepts extraits avec ceux de l'ontologie initiale (pour le premier passage et l'ontologie V_i pour le i ème passage).
- 3- Détermination des relations sémantiques.
- 4- Alimentation de l'ontologie avec les concepts C_i relatifs à T_i respectant les liens sémantiques établis.
- 5- Alimentation de l'ontologie avec les liens conceptuels correspondants.

7.4 Détermination du poids de concept dans une ontologie

Le calcul du poids de chaque concept ou sous concept dans notre système est généré automatiquement en utilisant l'ontologie de domaine déjà construite. De telle sorte que, les concepts qui se situent dans le même niveau hiérarchique dans l'ontologie ont le même poids (alignement horizontale). De plus, le poids s'incrémente par 1 à chaque avancement en profondeur pour les concepts. Cette technique nous a permis d'obtenir un poids pour chaque concept, sous concept et même pour les individus. La figure31 donne une vue globale de notre ontologie de domaine construit.

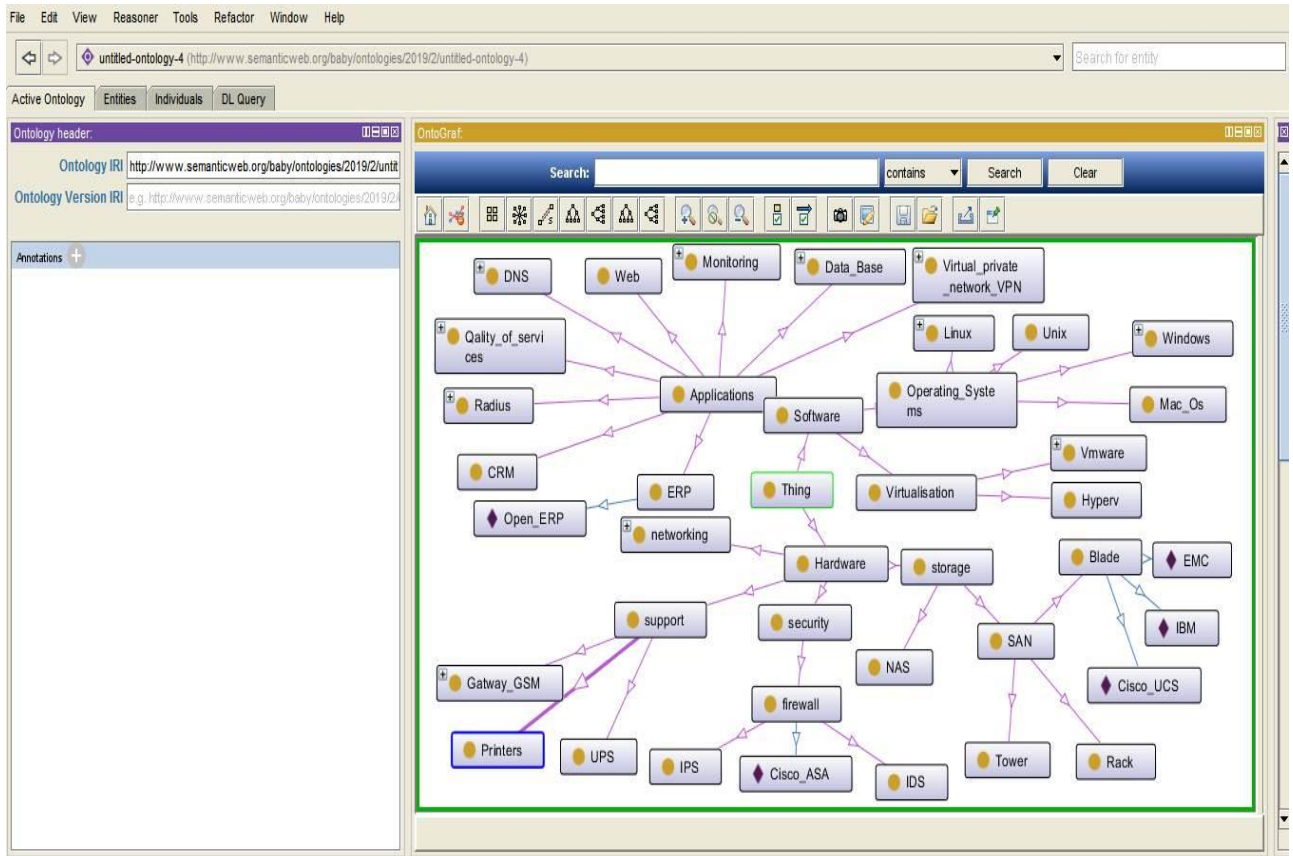


Figure31 : Exemple d'ontologie de domaine pour une infrastructure informatique.

Après la création de notre ontologie, nous l'avons défini comme un service Web dans notre directory UDDI.

8. Agrégation des solutions

Dans cette section, nous allons procéder à l'agrégation de toutes les performances pour avoir une seule valeur par alternative qui nous permettra de classer les alternatives. Cependant, une bonne agrégation est fondée sur un calcul de poids adéquat et efficace. Dans notre travail, le calcul de poids est généré à partir de notre ontologie de domaine présenté dans la sous-section 7 de ce chapitre. Sur la base du classement final qui est généré par le système, nous pouvons cibler la solution à tester. Cette dernière sera localisé différemment : dans la base des cas archivée ou bien dans le registre UDDI ou bien via le WEB.

Problems Agregation						
	start	vmware	virtual	machine	WAS-PAS	Rank
Problem1	0.4099	0.4136	0.5102	0.5191	0.55547	5
Problem2	0.5003	0.4903	0.4467	0.4644	0.57151	2
Problem3	0.4356	0.4556	0.5174	0.5262	0.5784400000000001	1
Problem4	0.267	0.4308	0.3825	0.4283	0.43620000000000003	7
Problem5	0.5013	0.4756	0.4373	0.453	0.5627300000000001	3
Problem6	0.5013	0.4722	0.4373	0.453	0.56205	4
Problem7	0.4194	0.4416	0.394	0.3907	0.49149000000000004	6
Problem8	0.3391	0.4334	0.3335	0.365	0.43187000000000003	8

Figure32 : Agrégation des solutions avec l'algorithme WAS-PAS.

Comme le montre la Figure 32, notre algorithme WAS-PAS utilisé par le système donne un rang important. Par exemple, le problème 3 est plus similaire au problème traité. De plus, toutes les agrégations A_j qui sont supérieures au seuil $\Omega \approx 0,55$ génèrent un résultat satisfaisant et acceptable.

9. Préparation d'une session de collaboration (Couche2)

La session de collaboration a été préparée par les agents de réunion. En effet, cette session sera programmée dans le cas où aucune solution n'est trouvée. Cet agent de réunion collabore avec tous les agents participants et les experts pour élaborer et contrôler cette session.

Reference	Description	Stat	Domain
59	impossible to create raid 5 in mt Blade UCS with raid software manager	0	Storage
60	F interconnect with low flow trafic	0	Storage
61	ASR(Cisco), Cisco IOS : Update	0	networking
62	informations missing from vmotion and vstorage into my vcenter	0	Virtualization
63	OS(Windows/Linux), Windows service, LINUX	0	Applications
64	no access with ma active directory account name to my private session	0	Applications
65	all catalyst power link are down	0	Networking
66	no access to my local area network with my catalyst switch	0	Networking

Meething Id	Forum Id	participant Id	Prb Id	Start Date	Start Time	End Date	End Time
4	1	5	67	2018-04-20	23:29:24	2018-04-20	23:29:24

Figure33: Préparation de la Session de Collaboration.

Comme le montre la Figure33, l'agent de réunion donne un aperçu global des différents problèmes avec leur réunion d'information sur l'évolution comme la date/l'heure début et fin, et des informations sur les participants invités et le numéro du forum auquel ils t été invités.

10. Supervision de diagnostic du problème (Multi Couches)

La tâche de surveillance est établie par l'agent participant. Celui-ci supervise dans la même interface l'évolution des problèmes comme le montre la Figure34. Surtout, il reprend l'enchaînement des tâches globales.

Num	Processing	Agent type	Time
1	Generation corpus Complete	Participant Agent	00:00:06
2	Extraction term Complete	Participant Agent	00:00:11
3	Calcul similarity Complete	Analyser Agent	00:00:14
4	Agregate Problems Complete	Analyser Agent	00:00:13
5	Selection Solution Complete	Research Agent	00:00:10
6	Preparing Meeting Complete	Expert Agent	00:00:24

Figure34 : Supervision de diagnostic du problème.

Comme le montre la Figure34 l'agent participant visualise les tâches accomplies et les tâches en cours de traitement pour chaque agent ainsi que leur durée. De plus, il a la possibilité de visualiser la progression globale dans le processus de recherche de la solution au problème posé.

11. Validation et évaluation des solutions

L'évaluation des solutions suit plusieurs méthodes qui sont principalement utilisées dans diverses applications. Dans notre travail, nous mettons en évidence un type de validation important, il s'agit de l'algorithme F-mesure. Celui-ci est considéré comme un moyen harmonique de rappel et la précision avec le maximum possible est de 1 (comme meilleur facteur). Il consiste à faire une comparaison entre le résultat généré par le système et celui espéré par les experts.

11.1 La Validation F-measure

La mesure F ou bien le score F est une mesure de la précision d'un test. Il est calculé à partir de la précision et du rappel du test, où la précision est le nombre de vrais résultats positifs divisé par le nombre de tous les résultats positifs, y compris ceux qui ne sont pas identifiés correctement, et le rappel est le nombre de vrais résultats positifs divisé par le nombre de tous les échantillons qui auraient dû être identifiés comme positifs. La formule 30 de F-measure est donnée comme :

$$F - measure = \frac{2 \textit{precision} * \textit{recall}}{\textit{precision} + \textit{recall}} \dots \dots \dots (30)$$

Pour plus de précision, nous sommes généralement en mesure de définir avec précision le rapport des solutions correctes divisé par le nombre de résultats renvoyés comme ci-dessous (Formule31) :

$$\textit{precision} = \frac{SPPN}{SPPN + SNPPN} \dots \dots \dots (31)$$

SPPN : représente le nombre de problème pertinent sélectionné.

SNPPN : représente le nombre de problème non pertinent sélectionné.

Cependant, pour le rappel, nous pouvons calculer le pourcentage de solutions pertinentes renvoyées par le système comme suit (Formule32):

$$\textit{recall} = \frac{SPPN}{SPPN+NSPPN} \dots \dots \dots (32)$$

NSPPN : Aucun numéro de problème pertinent sélectionné

11.2 La méthodologie de validation

Afin de mesurer la pertinence du système et assurer la qualité des solutions choisies, nous avons conçu un ensemble d'expériences en utilisant la méthodologie suivante :

- ✓ La description de 50 problèmes pannes différents dans plusieurs domaines.
- ✓ Le jeu de données utilisé dans les expérimentations est un ensemble de 50 problèmes posés et 180 problèmes candidats correspondants pour construire plusieurs corpus. Ces corpus ont été construits à l'aide de WWW et de la base de données globale.
- ✓ Évaluation des résultats générés par nos experts dans le domaine du diagnostic informatique afin de classer les solutions proposées de manière pertinente et non

pertinente avant de calculer la précision, le rappel et la F-mesure. Le résultat de cette étape est présenté dans la Figure35.

- ✓ L'ensemble des experts appartient à la société ISSAL FLEX CLOUD (Provider Cloud&Stockage). Cette dernière est une société multinationale qui possède un centre de donnée moderne avec des équipements et des applications de dernière génération.

Dans ce qui suit, nous présentons un échantillon des questions posées à l'ensemble des Ingénieurs de la société ISSAL FLEX CLOUD dans le domaine de la virtualisation. Les experts doivent sélectionner le taux de leur satisfaction en pourcentage par rapport à la solution choisies par notre système pour des différents problèmes (**Voir Tableau 10**).

Problèmes	0% - 25%	25% - 50%	50% -75%	75% -100%
I can't start my VMware virtual machine			✗	
Vmware service stop and no response from the virtual machine				✗
Vcentervmware does not communicate with Esxi			✗	
Ii can't migrate my VMware virtual machine from one Esxi to another, problem with vmkernel			✗	
No pool resources are assigned to this virtual machine			✗	
No memory apace available on esxi				✗
Probleme with vcenter and virtual machine			✗	
No start of virtual machine on scell mode				✗
No increase of the memory capacity of my virtual machine		✗		

Tableau10: Pourcentage de satisfaction des experts.

La représentation des valeurs de rappel, précision de de la F-mesure pour l'ensemble des 50 problèmes de la validation est donnée dans la Figure 34. Cette dernière affiche ces valeurs sous forme d'un histogramme et sous forme des courbes pour l'ensemble des problèmes posés dans notre système.

Dans la Figure 36, nous observons que la moyenne de valeur qui est mesurée pour le jeu de données étudié dans notre système est estimée à 0,693 | 0,685 | 0,657 concernant la précision, le rappel et la f-mesure respectivement. De plus, l'expérimentation a montré que la valeur minimale acceptable pour chaque fonction est équivalent à 0,61. En conséquence, ces valeurs semblent très encourageantes et montre la persistance de notre système. Ainsi, nous pouvons dire que

l'approche proposée donne des bons résultats. De plus, les graphiques représentés dans la Figure 35 montrent le résultat de la mesure F, du rappel et de la précision afin d'évaluer l'approche proposée.

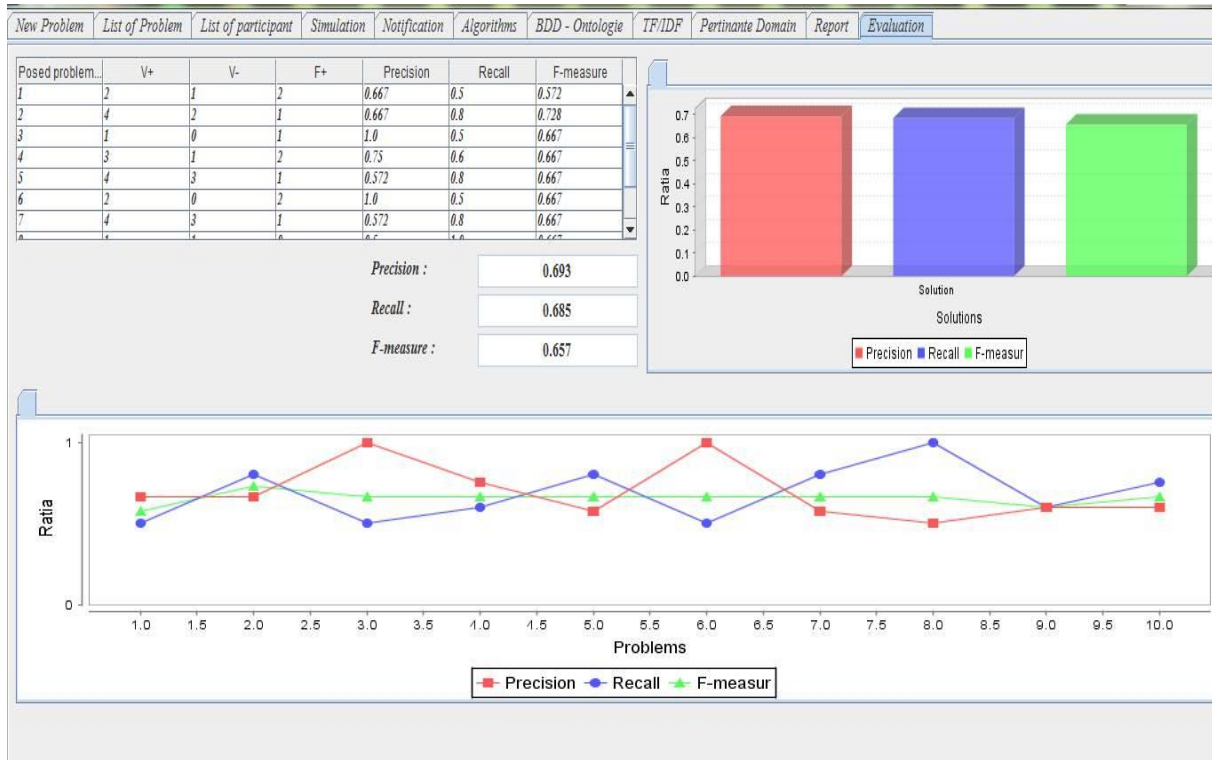


Figure35 : Validation du MAITD.

12. Conclusion

Dans ce chapitre nous avons présenté un exemple de diagnostic dans un domaine précis et choisi aléatoirement selon notre contexte d'étude. Des écrans d'illustration de l'exécution ou le déroulement des différentes phases sont donnés. Nous avons commencé le déroulement par la description du problème jusqu'à le choix et le test de la solution finale.

Pour la validation de notre système proposé nous avons effectué une série de tests après la préparation de notre corpus de données. Ce dernier est très important dans notre approche dans le but d'obtenir des résultats acceptables. Il nous a permis de sélectionner une série de solutions candidates dans l'étape de calcul de la similarité ce qui nous a motivé de les stocker comme un service pour les prochaines interventions similaire. Finalement, Nous pouvons dire que nous avons obtenu des résultats satisfaisants.

Conclusion Générale

Dans notre travail de recherche, une architecture multi couche orienté services pour le diagnostic informatique nommée MAITD est proposée afin d'aider les techniciens de l'entreprise à trouver une solution (s) en peu de temps face à un problème de prise de décision. En effet, nous avons développé une méthode basée sur un clustering non supervisé dans un cadre de prise de décision multicritère (MCDM). Le but ultime de ce travail est de dispatcher les tâches de la maintenance opérationnelle corrective sur un ensemble d'agents, en tenant compte de la complexité et de la représentation dynamique des problèmes. Premièrement, la méthode de clustering utilisée (K-means) regroupe tous les problèmes candidats dans les mêmes clusters similaires au problème initial posé afin de construire le fameux corpus. En effet, pour faire face à la quantité énorme des données disponible dans le Web. Notre corpus est construit en interrogeons le Web comme source principale plus d'autres sources de donnée tel que les problèmes archivés, UDDI...etc.

Secondairement, le système extrait automatiquement et d'une manière non supervisée les termes clés du problème posé d'où l'expert de domaine les jugent pertinents afin d'avoir une couverture de l'ensemble du problème traité. En effet, plusieurs méthodes d'extraction sont utilisé comme la méthode TF-IPF et la méthode PD (Pertinence dans le domaine).

De plus, nous avons montré comment les méthodes de similarité et d'agrégation des solutions sélectionnées sont implémentées dans notre travail avant de sélectionner une solution spécifiée.

Pour cette raison, nous utilisons des systèmes multi-agents pour rechercher la meilleure solution parmi un ensemble de propositions. Ces solutions sont considérées comme des services web stockés dans un annuaire UDDI et qui sont de plusieurs formes : plug-ins, script de mise à jour, ontologies, procédures de résolutions....etc.

Ainsi, dans ce travail nous proposons deux architectures. La première multi couche à base d'agents afin de bénéficier de la fiabilité et l'autonomie de ces agents d'une part et d'avoir une indépendance entre les composants de chaque couche d'autre part. La deuxième architecture est à base de module pour montrer les différentes méthodes et techniques implémentées dans chaque étape de notre approche.

L'approche actuelle fournit les services de surveillance et de diagnostic correctif, qui sont capables de réagir de manière indépendante au problème informatique demandé. En outre, elle pourrait être adaptée à d'autres domaines en introduisant des ontologies correspondantes.

L'étape de validation de notre travail nous a permis de déduire que notre système donne des résultats satisfaisants à un taux qui dépasse les 60%. Dans cette étape de validation, plusieurs experts de différentes entreprises et dans divers spécialités ont fait leur intervention sur un ensemble de problèmes réel souvent rencontrés dans les infrastructures IT. Les résultats démontrent l'efficacité de l'approche proposée par rapport à l'état de l'art.

Perspectives

Parmi les perspectives relatives a nos travaux de recherche, nous prévoyons de :

- ✓ Compléter l'approche proposée pour réaliser une maintenance hybride qui regroupe les deux méthodes de maintenance préventive et corrective.
- ✓ Montrer comment le poids des critères est calculé en détail à partir de notre ontologie de domaine déjà développée avant de faire l'agrégation des solutions comme dernière étape dans notre approche.
- ✓ Montrer les différents mécanismes mis en œuvre pour intégrer une solution issue dans un UDDI. Cette notion d'UDDI n'a pas été bien détaillée dans notre travail mais nos futurs travaux visent à fournir un mécanisme de publication des services qui comprend la fonctionnalité et les besoins d'entrée/sortie.
- ✓ Etendre cette approche de diagnostic sur d'autres domaines tel que les usines industrielles, Trafic de transports.....etc.

Référence Bibliographique

R.Yende. SUPPORT DE COURS DE MAINTENANCE INFORMATIQUE. Licence.Congo-Kinshasa.cel-01966296, 2018

V.JaramilloJimenez, N.Bouhmala, A.Haugen Gausdal: Developing a predictive maintenance model for vessel machinery, Journal of ocean engineering and science 2020 358-386, may 2020.

P.Velardi, M.Missikoff, R.Basili : Identification of relevant terms to support the construction of Domain Ontologies ,2001.

A.NAFI , C.WEREY : Aide a la décision multicritère : introduction aux méthodes d'analyses multicritere de type électre ,Support du cours ENGEES , 2010.

B.Amokrane : Structuration des processus d'aide à la décision par analyse bipolaire, Toulouse 3, 176 p, 20 avril 2017.

M.Abdelhadi : Élaboration d'un modèle de négociation par les web services dans un système d'aide multicritère a la décision de groupe, these université ahmed benbella oran1, mars 2019

F.Soumana : Un système de recommandation pour le choix de l'opérateur d'agregation ,these université de toulouse1 capitole , juillet 2018.

M.C Bekhatri : Les méthodes multi criteres pour analyser les aotitudes des terres agricoles cas du blé tendre en Languedoc-Rousillon analysé avec la méthode AHP , Université de montpellier, Septembre 2015.

R. YAGER :On ordered weighted averaging aggregation operators in multicriteriadecisionmaking,IEEE Trans. Systems, Man &Cybern, vol. 18, pp. 183-190, 1988.

M. SUGENO :Theory of fuzzy integrals and its applications. Thèse doctorale, TokyoInstitute of technology, 1974.

S. elandaloussi, P.Zarate, N.Taghezout : A Text Mining Approach agent based DSS for IT Infrastructure Maintenance, ISSN: 1941-6296 EISSN: 1941-630X DOI: 10.4018, IJDSST, 2021

S. elandaloussi, P.Zarate, N.Taghezout: A Multi Agent Architecture for IT Infrastructure Diagnostic, 5th International Conference on Decision Support System Technology, ISBN: 978-989-8805-44-7,May 2019

J A. Lossio ,V.Clement,J. Mathieu ,R. Maguelonne : Extraction automatique de termes combinant différentes informations, 21ème Traitement Automatique des Langues Naturelles, Marseille, 2014

N. J. VAN ECK, L. WALTMAN, E. C. NOYONS , R. K. BUTER : Automatic term identification for bibliometric mapping. *Scientometrics*, 82(3), 581–596,2010.

X. WAN, J.XIAO :Collabrank : Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, 2008

Z. LIU, X. CHEN, Y. ZHENG, M.SUN : Automatic Keyphrase Extraction by Bridging Vocabulary Gap. In Proceedings of the 15th Conference on Computational Natural Language Learning, 2011

LIU, Z., LI, P., ZHENG, Y. et SUN, M : Clustering to Find Exemplar Terms for Keyphrase Extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1, 2009.

K. JONES : A Statistical Interpretation of Term Specificity and its Application in Retrieval,1979

A.Amine, Z.Elberichi, M.Simonet: Evaluation of textclustering methods using wordnet. *Int. Arab J. Inf. Technol.* 7(4): 349-357,2010

F.Boudin : A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. International Joint Conference on Natural Language Processing (IJCNLP), Nagoya, Japan.pp.834-838, 2013,Oct 2013

K.Sarkar, M.Nasipuri, S.Ghose, :A new approach to keyphrase extraction using neuralnetworks, arXiv preprint arXiv:1004.3274, 2010.

R. GAIZAUSKAS, G. DEMETRIOU, K.HUMPHREYS : Term recognition and classification in biological science journal articles, In Proceeding of the

Computational Terminology for Medical and Biological Applications Workshop of the 2nd International Conference on NLP, p. 37–44, 2000

M. KRAUTHAMMER, G. NENADIC : Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6), 512–526, 2004

Y. MATSUO, M. ISHIZUKA : Keyword Extraction from a Single Document Using Word Co-occurrence Statistical Information, 2004

Z. LIU, P. LI, Y. ZHENG, M. SUN, Clustering to Find Exemplar Terms for Keyphrase Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1*, 2009

Z. KOZAREVA, I. MATVEEVA, G. MELLI , V. NASTASE : *Proceedings of TextGraphs-8 GraphBased Methods for Natural Language Processing*. Seattle, Washington, USA : Association for Computational Linguistics, 2013.

R. MIHALCEA, P. TARAU : TextRank Bringing Order Into Texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

S. BRIN, L. PAGE : The Anatomy of a Large-Scale hypertextual Web Search Engine. In *Proceedings of the 7th International Conference on World Wide Web*, 1998

T. Hisamitsu, et al : Extracting terms by a combination of term frequency and a measure of term representativeness, *Terminology* 6 (2) :211-232, 2000.

M. Paukkeri, T. Honkela : Likey Unsupervised Language-Independent Keyphrase Extraction". In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010.

M. Paukkeri, I. Nieminen, M. Poll, T. Honkela : A Language-Independent Approach to Keyphrase Extraction and Evaluation", 2008.

Z. Liu, X. Chen, Y. Zheng, M. Sun : Automatic keyphrase extraction by bridging vocabulary gap, *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 135–144, Portland, Oregon, USA, 23–24 June 2011.

Z. Liu, P. Li, Y. Zheng, M. Sun : Clustering to Find Exemplar Terms for Keyphrase Extraction", 2009

O. Ferret : Similarité sémantique et extraction de synonymes à partir de corpus , *TALN 2010*, Montréal, 19–23 juillet 2010

A.Tchechmedjiev : JEP-TALN-RECITAL 2012, volume 3: RECITAL, pages 295–308,Grenoble, 4 au 8 juin 2012.

G.Bisson , S. F.Hussain, Chi-sim : A new similarity measure for the coclusteringtask. In ICMLA, pages 211_217,2008.

M.D. Kent, O. Costello, S. Phelan, K. Petrov : Cost Oriented Maintenance management Systems for Manufacturing Processes, Waterford Institute of Technology, Ireland, 2017.

X.Mehmeti, B. Mehmeti, R. Sejdiu:The Equipment Maintenance Management in Manufacturing Enterprises, PapersOnLine, 51-30 at www.sciencedirect.com ,2018.

O.Medelyan, I .Witten : Domain-Independent Automatic Keyphrase Indexing with Small Training Sets, 2008.

M. Aminul, D. Inkpen :Second Order Co-occurrence PMI for Determining the Semantic Similarity of Words, School of Information Technology and Engineering, LREC, 2006.

C. Fellbaum: WordNet: An electronic lexical database, Cambridge, 1998.

T.Slimani B.Yaghlane, K.Mellouli: A new similarity measure based on edge counting. In Proceedings of world academy of science, engineering and technology, 2006.

R.Rada, E. Bicknell, M.Blettner : Development and application of a metric on semantic nets. IEEE Transaction on Systems, Man, and Cybernetics, ,1989

Web Services structure, URL, <https://www.guru99.com/wsdl-web-services-description-language.html>, mis a jour aout 2021.

J.Mothe, F.Ramiandrisoa : Extraction automatique de termes-clés : Comparaison de méthodes non supervisées,2016

Berkhin :Survey of Clustering Data Mining Techniques", Springer-Verlag, Berlin Heidelberg, 2006

islam, D. Inkpen : Semantic text similarity using corpus based word similarity and string similarity, Transaction on knowledge discovery from data, New York, USA ,2008

M. Aminul, D. Inkpen :Second Order Co-occurrence PMI for Determining the Semantic Similarity of Word", School of Information Technology and Engineering, LREC, 2006.

Fellbaum :WordNet: An electronic lexical databas", Cambridge, 1998

R. Mihalcea, C. Corley, C. Strapparava: Corpus-based and knowledge-based measures of text semantic similarity, the American Association for Artificial Intelligence,Boston, 2006.

L.Y. Mclean, D. Bandar, J. O Shea, K. Crockett : Sentence similarity based on semantic nets and corpus statistics, IEEE Trans Knowl. Data Eng, 2006.

M.D. Kent, O. Costello, S. Phelan, K. Petrov :Cost Oriented Maintenance management Systems for Manufacturing Processes", Waterford Institute of Technology, Ireland, 2017

Nagios: IT management and monitoring product, URL, <https://www.nagios.com/>, 2007.

Manage Engine: IT management and monitoring product, URL. <https://www.manageengine.com/> ,November 2007.

Zenoss: IT Infrastructure management and monitoring tool for hybrid IT environment, URL, <https://www.zenoss.com/> , June 2018.

P. Callewaerta, W.J.C. Verhagena , R. Curran : Integrating maintenance work progress monitoring into aircraft maintenance planning decision suppor", 6th CEAS Air and Space Conference, Bucharest, 2017.

H.J Liao , C.H.R Lin,Y.C Lin, K.Y Tung : Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 , pages 16–24, 2013.

J.Campos, E. Jantunen, O. Prakash :Development of a Maintenance System Based on Web and Mobile Technologie", Journal of International Technology and Information Management, Volume 16, Number 4, 2007

P. Berkhin :Survey of Clustering Data Mining Techniques", Springer-Verlag, Berlin Heidelberg,2006

T. Chung: A Corpus Comparison Approach for Terminology Extraction
Terminology9 (2) : 221-245,2003.

J. A. Hartigan and M. A. Wong. Algorithm AS 136 : A k-means clustering algorithm. AppliedStatistics, 28(1) :100_108, 1979

JHipster Domain Language ,URL : <https://www.jhipster.tech/jdl/intro>

J. Farber : Les systèmes multi agents vers une intelligence collective, InyerEdition, Paris 1995.

H.Bond, L.Gasser : Reading in distributed artificial intelligence, morgan Kuffmann , publishers, inc 1988.

B. Roy : Multicriteria Methodology Goes Decision Aiding. Kluwer Academic Publishers, Berlin, 1996

B.Roy : Méthodologie multicritere d'aide a la décision ,economica ,paris , 1985

T. Kvan : Collaborative Design: What Is It? Automation in Construction, 9, 409-415, 2000.

E.Kerhoas, J.Helias, Médiathèque de Landerneau, ISNI **0000 0003 8562 6653**,2006

A.Scharlig : Décider sur plusieurs criteres , panorama de l'aide a la décision multicritere presses polytechniques Romandes Lausanne 1985

T.W. MALONE. What is coordination theory, in National Science Foundation Coordination Theory Workchop, MIT. 1988

C. Hewitt :The effect of political democracy and social democracy on inequality in industrial societies. American Sociological Review, 42, 450-465. Doi :10.2307/2094750 , (1977)

WSDL, URL : <https://crunchify.com/basic-wsdl-structure-understanding-wsdl-explained/> , mis a jour aout 2021