République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE
DEPARTEMENT : INFORMATIQUE

MEMOIRE DE MASTER

OPTION :

Modélisation informatique des connaissances et du raisonnement (MICR)

Thème

Alignement des modèles de processus métiers

Présenté par :

Mr.Chadli Mohamed Amine

Mr.Cheriti Ihab

Encadré par :

Mr: A.Mostfai

Promotion : juin 2018

Remerciements

Avant tout, Nous remercions الله ALLAH tout puissant de nous avoir donné la force et le courage pour terminer ce travail.

Mes remerciements s'adressent particulièrement au M.mostfai AbdelKader, Pour tous les conseils.

Nous n'oublions pas non plus nos enseignant, qui tout au long du cycle d'études à université Dr. Tahar Moulay SAIDA

Nous adressons une particulièrement affective à nos Amis de l'Université qui rendu agréable nos longue années d'études.

Un grand merci également à tous ceux qui ont contribué, de près ou de loin, à l'aboutissement de ce travail.

IHAB & AMINE

Dédicace

Merci Allah de m'avoir donnée la force et la patience afin d'atteindre mon objectif.

Tant recherché avec bonheur. Je dédie ce mémoire :

A mes parents :

A ma mère, qui a œuvrer pour ma réussite, de par son amour, son soutien, ces sacrifices consentis et ses précieux conseils, pour son assistance et sa présence dans ma vie ; Reçois à travers ce travail aussi Modest soit-il, l'expression de mes sentiments et de mon éternelle gratitude

A mon père, qui trouve ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie, à l'éducation reçue et aux valeurs nobles inculquée ; Je lui dire Merci

A mes sœurs.

A mon petit chère frère Oualid

A toute ma grande famille ....

A mon encadreur Pour tous les conseils.

A mon partenaire Ihab Cheriti qui accompagné durant tout mon cycle universitaire.

A mes amis (Sofiane, Mohieddine, Abdellah) que je remercie pour leurs soutiens pendent toutes ces années.

CHADLI MOHAMED AMINE

Dédicace

Merci Allah de m'avoir donnée la force et la patience afin d'atteindre mon objectif.

Tant recherché avec bonheur. Je dédie ce mémoire :

A mon grand-père Allah yer7mou.

A mes parents :

A ma mère, qui a œuvrer pour ma réussite, de par son amour, son soutien, ces sacrifices consentis et ses précieux conseils, pour son assistance et sa présence dans ma vie ; Reçois à travers ce travail aussi Modest soit-il, l'expression de mes sentiments et de mon éternelle gratitude

A mon père, qui trouve ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie, à l'éducation reçue et aux valeurs nobles inculquée ; Je lui dire Merci

A mon chère frère Sami

A toute ma grande famille ....

A mon chère olive y :D

A mon encadreur Pour tous les conseils.

Et spécialement au **Smart Club**

A mon partenaire Amine Chadli qui accompagné durant tout mon cycle universitaire.

A mes amis et amies que je remercie pour leurs soutiens pendent toutes ces années.

CHERITI IHAB

Table de matières

Chapitre I : Introduction général

I.1 Introduction

Les communautés d'administration de processus métiers et d'informatique ont orientée leurs attentions ces derniers années vers la gestion de processus métiers. Les membres de ces communautés sont caractérisés par leurs différences en termes de leurs contextes éducatifs et par leurs intérêts. Les personnes qui travaillent dans la gestion des projets sont intéressés par l'amélioration de la performance des opérations de leurs entreprises. Augmenter la satisfaction des clients, réduire les couts d'accomplir leurs travails et créer des nouveaux produits et services au moins cout sont des aspects très important dans la gestion des processus métiers. [1]

Les entreprises ont créé la gestion de processus métiers après qu'ils ont remarqué que chaque produit fourni par eux est fait en performant plusieurs activités. Les processus métiers joue un rôle très important dans l'organisations de ces activités et dans l'amélioration de compréhension pour leurs relations mutuelles. [1]

Les technologies de l'information en général et les systèmes d'information en particulier méritent un rôle important dans la gestion des processus métiers, car de plus en plus d'activités réalisées par une entreprise sont soutenues par des systèmes d'information. Les activités de processus métiers peuvent être exécutées manuellement par les employés de l'entreprise ou à l'aide de systèmes d'information. Il existe également des activités de processus métiers qui peuvent être mises en œuvre automatiquement par les systèmes d'information, sans intervention humaine. [1]

Une entreprise peut atteindre ses objectifs commerciaux de manière efficiente et efficace uniquement si les personnes et les autres ressources de l'entreprise, telles que les systèmes d'information, performe bien ensemble. Les processus métiers sont un concept important pour faciliter cette collaboration efficace. [1]

Dans de nombreuses entreprises, il existe un écart entre les aspects commerciaux organisationnels et les technologies de l'information qui est en place. Il est important de réduire cet écart entre l'organisation et la technologie, car dans les marchés dynamiques d'aujourd'hui, les entreprises sont constamment forcées de fournir des produits meilleurs et plus spécifiques à leurs clients. Les produits qui réussissent aujourd'hui pourraient ne pas réussir demain. Si un concurrent fournit un produit moins cher, mieux conçu ou plus facilement utilisable, la part de marché du premier produit diminuera probablement. [1]

Les moyens de communication basés sur Internet diffusent les informations sur les nouveaux produits à la vitesse de l'éclair, de sorte que les cycles de produits traditionnels ne sont pas adaptés pour faire face aux marchés dynamiques d'aujourd'hui. Les capacités à créer un nouveau produit et à le mettre rapidement sur le marché, et à adapter un produit existant à moindre coût sont devenus des avantages compétitifs pour les entreprises performantes. [1]

Tandis qu'au niveau organisationnel, les processus métiers sont essentiels pour comprendre le fonctionnement des entreprises, les processus métiers jouent également un rôle important dans la conception et la réalisation de systèmes d'information flexibles. Ces systèmes d'information constituent la base technique pour la création rapide de nouvelles fonctionnalités permettant de réaliser de nouveaux produits et d'adapter les fonctionnalités existantes pour répondre aux nouvelles exigences du marché [1].

I.2 Problématique

Pour contrôler leurs opérations commerciales, les organisations investissent de plus en plus de temps et d'efforts dans la création de modèles de processus. Dans ces modèles de processus, les organisations capturent les activités essentielles de leurs processus métiers ainsi que les dépendances d'exécution de l'activité. La taille croissante des références de modèles de processus dans l'industrie et le besoin qui en résulte de techniques de traitement automatisées ont conduit au développement de diverses techniques d'analyse de modèles de processus. Un type de telles techniques d'analyse sont des approches d'alignement de modèles de processus, qui visent à soutenir la création d'un alignement entre des modèles de processus, c'est-à-dire, l'identification des correspondances entre leurs activités. [2].

Nous proposons dans ce mémoire une approche basée sur WordNet pour aligner les modèles de processus métiers.

Chapitre II : Background

II.1 Génie logiciel

II.1.1 Histoire du génie logiciel

La première utilisation communément connue du terme génie logiciel était en 1968 comme titre d'une conférence de l'OTAN sur le génie logiciel. Un article de A. J. S. Rayl publié dans un magazine de la NASA en 2008, commémorant le cinquantième anniversaire de la NASA, indique que la scientifique de la NASA, Margaret Hamilton, avait inventé le terme plus tôt (Rayl, 2008). [3]

La nature des logiciels informatiques a considérablement changé au cours des quarante-cinq dernières années, avec des changements accélérés dans les quinze à vingt dernières années. Le génie logiciel a mûri dans la mesure où un corpus commun de connaissances en génie logiciel, connu sous l'acronyme SWEBOK, a été développé. Voir l'article « Software Engineering Body of Knowledge » (SWEBOK, 2013) pour plus de détails. Même avec cette bonne collection de connaissances de la discipline du génie logiciel, les changements rapides et continus dans le domaine font qu'il est essentiel pour les étudiants et les praticiens de comprendre les concepts de base du sujet et de comprendre quand certaines technologies et méthodologies sont appropriées ou ne sont pas. [3]

Nous commençons avec une brève histoire. À la fin des années 1970 et au début des années 1980, les ordinateurs personnels commençaient tout juste à être disponibles à un coût raisonnable. Il y avait beaucoup de magazines informatiques disponibles dans les kiosques à journaux et les librairies ; ces magazines étaient remplis d'articles décrivant comment déterminer le contenu des emplacements de mémoire spécifiques utilisés par les systèmes d'exploitation informatiques. D'autres articles décrivent des algorithmes et leur implémentation dans certains dialectes du langage de programmation BASIC. Les étudiants du secondaire gagnent parfois plus d'argent en programmant des ordinateurs pendant quelques mois que leurs parents ne le font en un an. La couverture du média a suggéré que les possibilités d'un programmeur talentueux et solitaire étaient illimitées. Il semblait probable que l'informatisation de la société et les changements fondamentaux provoqués par cette informatisation étaient motivés par les actions d'un grand nombre de programmeurs indépendants. Cependant, une autre tendance se produisait, largement cachée à la vue du public. Le logiciel prenait de plus en plus de taille et devenait

extrêmement complexe. L'évolution du logiciel de traitement de texte en est une bonne illustration. [3]

À la fin des années 1970, des logiciels tels que Microsoft Word et WordStar fonctionnaient avec succès sur de petits ordinateurs personnels avec une mémoire utilisateur de 64 kilo-octets seulement. Les premières versions de WordStar permettaient à l'utilisateur d'insérer et de supprimer du texte à volonté, de couper et coller des blocs de texte, d'utiliser l'italique et le gras pour définir le texte, modifier la taille des caractères et sélectionner un ensemble limité de polices. Un vérificateur d'orthographe était disponible. Un petit nombre de commandes était autorisé et l'utilisateur devait connaître les options disponibles avec chaque commande. Des listes de commandes étaient disponibles sur des modèles en plastique ou en carton placés sur le clavier pour rappeler aux utilisateurs quelles combinaisons de touches étaient nécessaires pour les différents types d'opérations. Les modèles en carton ou en plastique étaient généralement vendus séparément du logiciel. [3]

Microsoft Word et WordStar ont évolué avec le temps, tout comme la plupart de leurs concurrents, y compris le logiciel de traitement de texte Pages d'Apple. Presque tous les systèmes de traitement de texte modernes incluent toutes les fonctionnalités des traitements de texte originaux. En outre, un logiciel de traitement de texte moderne a généralement les caractéristiques suivantes :

- Il existe une interface utilisateur graphique qui utilise une souris ou un autre périphérique de pointage.
- Il existe un ensemble de formats de fichiers dans lesquels un document peut être ouvert.
- Il existe un ensemble de formats de fichiers dans lesquels un document peut être enregistré.
- Il existe un grand nombre de polices autorisées.
- Le logiciel a la capacité d'insérer, et peut-être de modifier, des tables importées à partir d'une feuille de calcul.
- Il existe des outils pour calculer le nombre de mots et d'autres statistiques sur le document.
- Il existe des fonctionnalités facultatives pour vérifier la grammaire.

La complexité ajoutée ne vient pas gratuitement, et un ingénieur ne peut pas gérer tout le code source, même s'il a compris tout le code et les fichiers de données nécessaires pour une nouvelle version du logiciel de traitement de texte, il n'y aurait pas assez de temps pour faire les changements nécessaires en temps opportun. Ainsi, la nature

concurrentielle du marché des logiciels de traitement de texte et la complexité des produits eux-mêmes obligent essentiellement à employer des équipes de développement de logiciels. C'est le cas dans le développement de logiciels modernes - il est généralement effectué par des équipes plutôt que par des individus. Les membres de ces équipes sont souvent appelés « ingénieurs logiciels ». Les ingénieurs logiciels peuvent travailler seuls sur des projets particuliers, mais la majorité d'entre eux sont susceptibles de consacrer la majeure partie de leur carrière à des équipes de développement logiciel. Les équipes elles-mêmes changeront au fil du temps en raison de l'achèvement de vieux projets, le début de nouveaux projets, et d'autres changements dans la carrière des individus et les changements technologiques rapides. [3]

Les problèmes liés aux systèmes logiciels de traitement de texte abordés dans cette section sont des problèmes typiques rencontrés par les ingénieurs logiciels. [4]

II.1.2 Définition du génie logicielle

Une définition formelle de génie logicielle pourrait ressembler à quelque chose comme « Une approche analytique organisée pour la conception, le développement, l'utilisation et la maintenance de logiciels ». [3]

Plus intuitivement, génie logicielle est tout ce dont vous avez besoin de faire pour produire un logiciel performant. Il comprend les étapes qui prennent une idée brute, peut-être nébuleuse et la transformer en une application puissante et intuitive qui peut être améliorée pour répondre aux besoins changeants des clients pour les années à venir. [3]

II.1.3 L'importance du génie logiciel

Produire une application logicielle est principalement simple dans son concept : prendre une idée et la transformer en un programme utile. Malheureusement pour des projets de portée réelle, il existe d'innombrables façons qu'un concept simple peut mal tourner. Les programmeurs peuvent ne pas comprendre ce que les utilisateurs veulent ou ont besoin (ce qui peut être deux choses distinctes), ils construisent donc la mauvaise application. Le programme peut être tellement plein de bugs qu'il est frustrant à utiliser, impossible à corriger et ne peut pas être amélioré au fil du temps. Le programme pourrait être complètement efficace, mais si déroutant que vous avez besoin d'un doctorat en résolution de puzzle pour l'utiliser. [3]

Le génie logiciel comprend des techniques permettant d'éviter les nombreuses erreurs qui, autrement, risquent d'entraîner l'échec de votre projet. Il s'assure que

l'application finale est efficace, utilisable et maintenable. Il vous aide à respecter les échéances et à produire un projet fini à temps et dans les limites du budget. Peut-être le plus important, Génie logicielle vous donne la possibilité de faire des changements pour répondre à des demandes inattendues sans oblitérer complètement votre calendrier et vos contraintes budgétaires. En bref, Génie logicielle vous permet de contrôler ce qui pourrait ressembler à un tourbillon aléatoire du chaos. [3]

II.2 Processus Métiers

II.2.1 Introduction

Un processus est une série de tâches complétées pour atteindre un objectif. Un processus d'entreprise est donc un processus axé sur la réalisation d'un objectif pour une entreprise. Tout d'un processus simple pour faire un sandwich, la construction d'une navette spatiale utilise un ou plusieurs processus métiers. Les processus sont quelque chose que les entreprises passent tous les jours afin d'accomplir leur mission. Le meilleur de leurs processus, le plus efficace de l'entreprise. Certaines entreprises voient leurs processus comme une stratégie pour obtenir un avantage concurrentiel. Un processus qui atteint son objectif d'une manière unique peut distinguer une entreprise. Un processus qui élimine les coûts peut permettre à une entreprise de baisser ses prix (ou de conserver plus de profit). [5]

II.2.2 Définitions

Définition 1.1 : Un processus métier consiste en un ensemble d'activités exécutées en coordination dans un environnement organisationnel et technique. Ces activités réalisent conjointement un objectif d'affaires. Chaque processus métier est mis en œuvre par une seule organisation, mais il peut interagir avec les processus métiers exécutés par d'autres organisations. Après un premier examen des processus métiers, de leurs constituants et de leurs interactions, le point de vue est élargi. La gestion des processus métiers couvre non seulement la représentation des processus métiers, mais également des activités supplémentaires. [1]

Définition 1.2 La gestion des processus métiers comprend des concepts, des méthodes et des techniques pour prendre en charge la conception, l'administration, la configuration, la mise en œuvre et l'analyse des processus métiers [1]

Définition 1.3 Un système de gestion des processus métiers est un système logiciel générique piloté par des représentations de processus explicites pour coordonner la mise en œuvre des processus métiers. [1]

II.2.3 BPMN (BUSINESS PROCESS MODELING NOTATION)

En 2000, un consortium d'entreprises impliquées dans le développement du commerce électronique, s'est donné pour objectif de définir un langage de description des processus métiers, qui puisse en traduire la complexité tout en restant accessible. Cela a donné lieu à un formalisme orienté activité, BPMN, en partie inspiré d'UML, et qui en 2005, a été adopté par l'OMG comme UML l'avait été quelques années auparavant. [6]

BPMN est une notation, c'est-à-dire un ensemble de symboles permettant de représenter des processus métiers sous forme graphique. Par rapport aux langages antérieurs, on peut relever que le diagramme d'activités d'UML a été une source d'inspiration, mais BPMN a eu un apport majeur dans la représentation des différents échanges entre processus. Il a, en effet, été conçu pour pouvoir modéliser des processus privés (internes à une entreprise) comme des processus publics (qui impliquent deux ou plusieurs organisations). [6]

II.2.3 Les composants essentiaux pour BPMN

Un processus métier doit d'abord être clairement encadré, de façon à le positionner dans une vision métier globale au sein du SI :

- L'évènement déclencheur (ex : le client commande),
- Le (ou les résultats) attendus (ex : livraison et facturation terminées),
- Les objectifs poursuivis (ex : la réduction des délais de livraison).

Ensuite, le nommage. Le processus est nommé avec un verbe ou une locution verbale : par exemple, "Traiter un sinistre", "Instruire un dossier de prêt". On évite à l'inverse les termes flous comme "Gestion des dossiers", ou les termes relevant plus de fonction comme "Facturation". [7]

Les principaux éléments constitutifs du processus sont les suivants (FigureII.2-1):

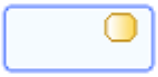
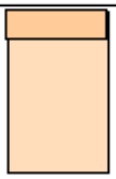
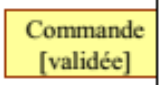
Terme	Activité UML	BPMN	Définition
Activité/ Processus			Représente un processus, et contient les éléments du processus (actions, partitions …).
Action/ Tâche			Unité d'exécution ou tâche prise en charge par une partition.
Partition/ Lane			Représente l'entité en charge de la réalisation des actions. Il peut s'agir d'acteur, de structure d'entreprise ou d'organisation.
Object node /Data Object			Représente les informations échangées entre les actions. Il est possible d'indiquer l'état de l'objet entre crochets.
Transition			Matérialise le passage d'une action à l'autre.
Décision			Permet de définir un branchement conditionnel.
Début de processus			Définit le démarrage du processus.
Fin de processus			Arrêt du processus.
Fin de branche			Termine une branche du processus sans arrêter le processus global, dont certaines branches peuvent continuer.

Figure II.2-1: les éléments de bpmn en comparaison avec uml

II.2.4 Exemple

Par exemple, le processus métier "Commande produit" (Figure II.2.1) a pour objectif de livrer et facturer au client le produit commandé en respectant les délais. Il faut noter qu'un modèle de processus métier décrit en général le métier, et non le système informatique. Certaines actions décrites sont exécutées manuellement, sans interaction avec un composant ou une application logicielle (par exemple, l'action "Livrer produit" peut être réalisée sans utilisation d'un élément logiciel).

Un processus métier est transverse, il s'appuie en général sur plusieurs structures et applications d'une organisation, voire de plusieurs organisations. (Par exemple, le processus de constitution d'un séjour intègre l'agence de voyage, le tour opérateur et la compagnie aérienne). [7]

En UML, les processus métiers sont représentés à l'aide du diagramme d'activité (Figure II.2.2).

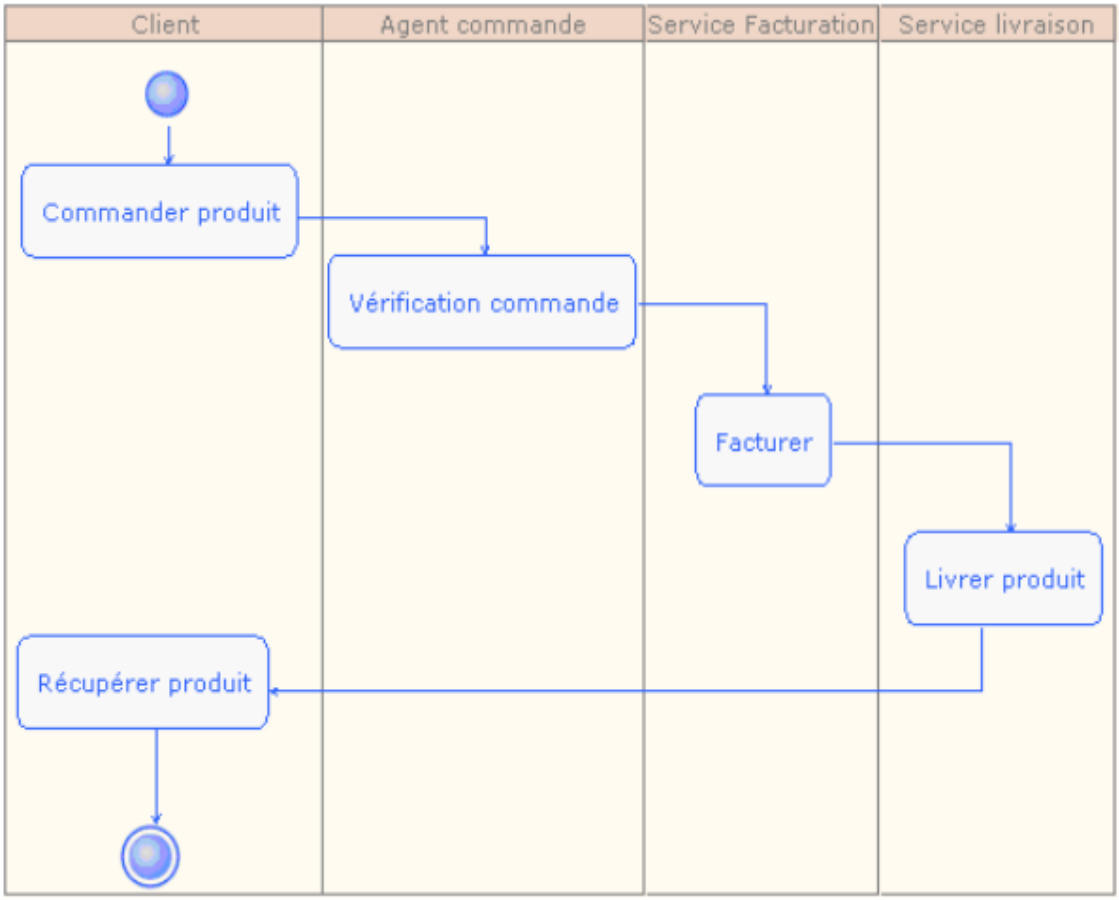


Figure II.2-2 Représentation d'un processus métier

II.2.4.1 Processus métiers et cas d'utilisation

Il existe des relations entre les processus métiers et les cas d'utilisation. Chaque action d'un processus métier non manuelle s'appuie sur des interactions avec un élément du système informatique, qui sont représentées par des cas d'utilisation. Par exemple, l'action "Vérifier commande" est liée au cas d'utilisation "Vérifier commande", qui décrit de manière détaillée comment procède "l'agent commande" pour effectuer cette vérification . [7]

Cas d'utilisation	Processus métier
1 seul acteur bénéficiaire (focalisé sur un acteur)	Collaboration entre plusieurs acteurs.
Unité de temps réduite	Peut durer plusieurs années.
Non interruptible (un flux simple)	Généralement interrompu, notion d'état et reprise sur évènement.
Localisé20	Transverse. Sur plusieurs structures, voire plusieurs entreprises.

Tableau II.2-1 comparaison entre cas d'utilisation et PM

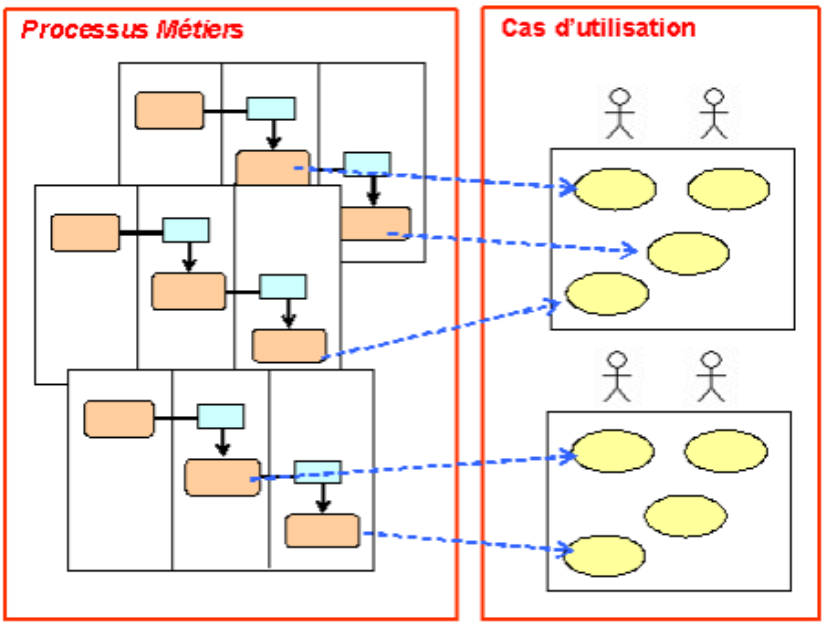


Figure II.2-3: Relation processus métier - cas d'utilisation

Par contre, les actions strictement manuelles ne sont pas liées aux cas d'utilisation, car elles ne nécessitent aucune interaction avec le système. [7]

II.2.4.2 Caractéristiques d'un processus métier :

Un processus métier as un ensemble des caractéristiques suivant :

- Durée (moyenne) : un jour, plusieurs années.
- Fréquence d'exécution : entre une exécution par an et plusieurs exécutions par jour
- Nombre d'utilisateurs : (par type d'utilisateur).
- Resource utilisé (applications, référentiel …).

Ces caractéristiques pourront être utilisées pour déterminer des priorités. au niveau d'une grand organisation ,on peut pas de détaillé tous les processus , pour cela en donne des priorité aux processus les plus critiques pour cette organisation [7]

II.2.5 Conclusion

Les processus métiers aujourd'hui sont trouvé par tout à cause de leur importance el leur rôle, ils sont surtout utilisés dans les activités orientées métiers (expression des besoins, spécification ou analyse suivant la terminologie employée). Les modèles de processus métiers constituent également une partie importante des activités transverses de l'entreprise (urbanisation, cartographie, BPM et SOA) [7].

II.3 Mesures de similarité

II.3.1 Définition de Mesures de similarité

Les mesures de similarité sont fréquemment utilisées dans l'analyse de similarité de texte et dans le Clustering. Toute mesure de similarité ou de distance mesure habituellement le degré de proximité entre deux entités, qui peut être n'importe quel format de texte, comme des documents, des phrases ou même des termes. Ces mesures de similarité peuvent être utile pour identifier des entités similaires et distinguer des entités clairement différentes les unes des autres. Les mesures de similarité sont très efficaces, et parfois choisir la bonne mesure peut faire beaucoup de différence dans la performance de votre système d'analyse final. [9]

II.3.2 Catégories de Mesures de similarité

Les concepts de similarité et de distance sont cruciaux dans de nombreuses applications scientifiques. Les mesures de similarité et de distance sont principalement nécessaires pour calculer les similarités / distances entre différents objets, une exigence essentielle dans presque toutes les applications de reconnaissance de modèle incluant la classification, la sélection de caractéristiques, la régression et la recherche. Il existe un grand nombre de mesures de similarité dans la littérature ; ainsi, choisir une mesure de similarité la plus appropriée pour une tâche particulière est une question cruciale. Le succès ou l'échec de toute technique de reconnaissance de modèle dépend en grande partie du choix de la mesure de similarité. Les mesures de similarité varient en fonction des types de données utilisés.

Les mesures de similarité peuvent être largement divisées en plusieurs catégories :

- Mesures de similarité pour les variables binaires.
- Mesures de similarité pour les variables catégoriques.
- Mesures de similarité pour les variables ordinales.
- Mesures de similarité pour les variables quantitatives.
- Dissimilarité entre deux groupes de variables. [10]

II.3.2.1 Similarité / Dissimilarité pour les variables binaires

Les variables binaires ne peuvent prendre que les valeurs 0 ou 1 / oui ou non / vrai ou faux / positif ou négatif, etc. La similarité de dissimilarité (distance) de deux objets représentés par des variables binaires peut être mesurée en termes de nombre d'occurrence (fréquence) de positif et négatif dans chaque objet. [11]

L'utilisation la plus courante de la dissimilarité binaire (distance) est :

- **La distance de Simple matching :**

  Simple matching coefficient et simple matching distance Sont utile lorsque les valeurs positives et négatives sont toutes égales (symétrie). Par exemple (mâle et femelle) a un attribut de symétrie parce que le nombre d'hommes et de femmes donne une information égale.

- **La distance de Jaccard's :**

  Jaccard's coefficient (mesure la similarité) et la distance de Jaccard (mesure la dissimilarité) sont des mesures d'informations asymétriques sur des variables binaires (et non binaires).

- **La distance de Hamming :** distance de Hamming pour les variables binaires

La séquence binaire 0 et 1 finie est parfois appelée un mot dans la théorie du codage. Si deux mots ont la même longueur, nous comptons le nombre de chiffres dans les positions où ils ont des chiffres différents. La longueur des différents chiffres est appelée distance de Hamming. [11]

II.3.2.2 Distance pour la variable nominale / catégorique

Les variables nominales / catégorielles sont celles qui ne peuvent être mesurées de manière quantitative ; ils représentent plutôt certaines catégories de données. Dans le cas de variables nominales ou catégoriques, les nombres ne sont utilisés que pour représenter différentes catégories ; Par exemple, le genre est une variable nominale dont la valeur est 1 = masculin et 2 = féminin. La caractéristique principale de la variable nominale ou catégorique est que les catégories doivent être étiquetées "de manière cohérente". L'ordre des étiquettes n'est pas important, mais la cohérence est très importante. À titre d'exemple, l'étiquetage selon le genre peut être changé en 10 = féminin, 15 = masculin, sans affecter la logique de représentation. [10]

Cependant, cet étiquetage devrait être utilisé de manière cohérente tout en utilisant certaines variables catégoriques. On peut générer son propre étiquetage tant que la consistance est préservée. Pour calculer la distance entre deux objets ayant des caractéristiques catégorielles, il faut d'abord compter le nombre de catégories possibles pour chaque caractéristique. Dans le cas de deux catégories, des mesures de distance pour des variables binaires telles que simple matching, la distance de Jaccard ou de Hamming peuvent être utilisées. Si le nombre de catégories est supérieur à deux, la transformation de ces catégories en un ensemble de variables binaires est nécessaire. Il existe deux méthodes pour transformer une variable catégorielle (avec un nombre de catégories supérieur à 2) en variables binaires:

1. Chaque catégorie est représentée par une variable binaire.

2. Chaque catégorie est représentée par plusieurs variables binaires.

II.3.2.3 Distance pour les variables ordinales

Les variables ordinales fournissent généralement un classement d'un ensemble de données donné en termes de degrés. Ces variables sont principalement utilisées pour indiquer l'ordre / classement relatif de certains points de données. Ainsi, la différence quantitative entre ces variables n'est pas importante, mais leur ordre est important ; Par exemple, considérez les « notes de notation » qui sont utilisées pour fournir des notes aux étudiants. AA serait classé plus haut que AB, et BB est plus élevé que CC. Ici encore, le classement est important, pas la distance précise entre AA et AB [10]. Parfois, les nombres peuvent également être utilisés pour représenter des variables ordinales. Voici quelques exemples d'échelle ordinale :

• Classement relatif : AA = excellent, AB = bon, BB = moyen, DD = médiocre.

• Rang de priorité : 1 = meilleur, une valeur plus élevée indique une priorité faible.

• Évaluation de la satisfaction : 1 = très insatisfait, 100 = très satisfait.

Afin de calculer les distances / dissimilarités entre les variables ordinales, les méthodes suivantes sont principalement utilisées : transformation de rang normalisée, distance de Spearman, distance de footrule, distance de Kendall, distance de Cayley, distance de Hamming, distance de Chebyshev / Maximum et distance de Minkowski. [10]

II.3.2.4 Distance pour les variables quantitatives

Les variables quantitatives peuvent être mesurées sur une échelle numérique. Ainsi, ils sont différents des variables catégoriques, qui représentent principalement certaines catégories, ainsi que des variables ordinales, qui représentent l'ordre des variables. Ces variables quantitatives sont mesurées en nombre. Ainsi, toutes les opérations arithmétiques peuvent être appliquées à des variables quantitatives. Des exemples de variables quantitatives sont la taille, le poids, l'âge, la somme d'argent, le salaire minimum, le salaire, la température, la superficie, etc. [10]

Pour calculer les distances / dissimilarités entre les variables quantitatives, on utilise les méthodes suivantes : Euclidean Distance, Minkowski Distance of Order λ, City Block Distance, Chebyshev Distance, Canberra Distance, Bray-Curtis Distance, Angular Separation, Correlation, CoefficientMahalanobis Distance. [10]

II.3.3 La distance de Levenshtein

La distance de Levenshtein (LD) [12] est une mesure de similarité entre deux chaînes(String), S & T ; La distance est le nombre de suppressions, d'insertions ou de substitutions nécessaires pour transformer s en t.

Quelques cas d'utilisation de la mesure de similarité Levenshtein :

- Vérification orthographique
- Reconnaissance de la parole
- Analyse de l'ADN
- Détection de plagiat

II.3.3.1 Les étapes de l'algorithme

Etapes	Description
1	Définir n comme étant la longueur de s. Définissez m comme étant la longueur de t. Si n = 0, retournez m et quittez. Si m = 0, renvoyez n et quittez. Construire une matrice contenant 0..m lignes et 0..n colonnes.
2	Initialisez la première ligne à 0.n. Initialisez la première colonne à 0..m.
3	Examinez chaque caractère de s (i de 1 à n).
4	Examinez chaque caractère de t (j de 1 à m).
5	Si s [i] est égal à t [j], le coût est 0. Si s [i] n'est pas égal à t [j], le coût est 1.
6	Définir la cellule d [i, j] de la matrice égale au minimum de: a. La cellule immédiatement supérieure plus 1: d [i-1, j] + 1. b. La cellule immédiatement à gauche plus 1: d [i, j-1] + 1. c. La cellule diagonalement au-dessus et à gauche plus le coût: d [i-1, j-1] + coût.
7	Après que les étapes d'itération (3, 4, 5, 6) sont terminées, la distance est trouvée dans la cellule d [n, m].

Table II.3-1 les étapes d'algorithme [13]

II.3.3.2 Exemple

On applique l'algorithme sur deux chaine s1 = 'Fast' et S2= 'Fats' après les itérations, le tableau suivant (Table II.3.2-) présent le résultat de LD (s1, s2) dans la cellule en gras.

		F	A	S	T
	0	1	2	3	4
F	1	0	1	2	3
A	2	1	0	1	2
T	3	2	1	1	1
S	4	3	2	1	**2**

Table II.3-2 résultat obtenue après l'application d' algorithme

1. Si s est "FAST" et T est "FAST", alors LD (S, T) = 0, car il n'y pas aucune suppressions, insertion ou substitution
2. Si s est "FATS" et t est "FAST", alors LD (S, T) = 2, car 2 substitution (change "T" en "S" & "S" en "T") pour transformer S en T

LD (FAST, FATS) = 2(résultat cadrée dans (tableau)), par ce qu'en as besoin 2 substitution pour transforme Fats to Fast.

II.3.4 Precision et Recall

Dans la reconnaissance de formes et la recherche d'information, la précision (aussi appelée valeur prédictive positive) est la fraction des instances récupérées qui sont pertinentes, tandis que le recall (aussi appelé sensibilité) est la fraction des instances pertinentes récupérées.

Par exemple, s'il y avait 50 documents pertinents dans un corpus où 20 des 50 documents étaient pertinents pour un utilisateur, et qu'un système de recherche d'information (IR) renvoyait 20 documents, où 6 des documents étaient pertinents, le recall serait 6 / 50 = 0,12, et la précision serait de 6/20 = 0,3.

precision = vrai positif / (vrai positif + faux positif)

recall = vrai positif / (vrai positif + faux négatif)

II.3.5 La F Mesure

La F mesure (score F1 ou score F) est une mesure de l'exactitude d'un test et elle est définie comme la moyenne harmonique pondérée de la précision et du recall du test. F Mesure = 2 * ((Precision * Recall) / (Precision + Recall)).

II.4 Le traitement automatique de la langue (TAL)

II.4.1 Définition du TAL

Le traitement du langage naturel est défini comme un domaine spécialisé de l'informatique et de l'intelligence artificielle avec des racines en linguistique computationnelle. Il est principalement concerné par la conception et la construction d'applications et de systèmes qui permettent l'interaction entre les machines et les langages naturels évolués pour une utilisation par les humains. Cela rend également le traitement du langage naturel lié au domaine de l'interaction homme-machine (IHM). Les techniques de traitement du langage naturel permettent aux ordinateurs de traiter et de

comprendre le langage humain naturel et de l'utiliser davantage pour fournir des résultats utiles. [9]

II.4.2 Les applications du TAL

Les principales applications de TAL :

- Traduction automatique
- Systèmes de reconnaissance vocale
- Systèmes de questions réponses
- Résumé de texte
- Catégorisation du texte

II.4.2.1 Traduction automatique

La traduction automatique est peut-être l'une des applications les plus convoitées et les plus recherchées pour TAL. Elle est définie comme la technique qui aide à fournir une traduction syntaxique, grammaticale et sémantiquement correcte entre deux paires de langues. C'était peut-être le premier grand domaine de recherche et développement en TAL. À un niveau simple, la traduction automatique est la traduction du langage naturel effectuée par une machine. Par défaut, les blocs de construction de base pour le processus de traduction automatique impliquent une simple substitution de mots d'une langue à une autre, mais dans ce cas, nous ignorons des choses comme la grammaire et la cohérence de la structure phrastique. Ainsi, des techniques plus sophistiquées ont évolué sur une période de temps, y compris la combinaison de grandes ressources de corpus de texte avec des techniques statistiques et linguistiques. [9]

II.4.2.2 Systèmes de reconnaissance vocale

C'est peut-être l'application la plus difficile pour la PNL. Le test de Turing est peut-être le test le plus difficile de l'intelligence dans les systèmes d'intelligence artificielle. Ce test est défini comme un test d'intelligence pour un ordinateur. Une question est posée à un ordinateur et à un humain, et le test est passé s'il est impossible de dire laquelle des réponses données a été donnée par l'humain. Au fil du temps, beaucoup de progrès ont été réalisés dans ce domaine en utilisant des techniques comme la synthèse de la parole, l'analyse, l'analyse syntaxique et le raisonnement contextuel. Mais une limitation principale reste pour les systèmes de reconnaissance vocale : ils sont très spécifiques au domaine et ne fonctionneront pas si l'utilisateur s'éloigne même un peu des entrées scriptes attendues nécessaires au système. Les systèmes de reconnaissance

vocale se trouvent maintenant dans de nombreux endroits, des ordinateurs de bureau aux téléphones mobiles en passant par les systèmes d'assistance virtuels. [9]

II.4.2.3 Systèmes de questions réponses

Les systèmes de questions réponses (SQR) reposent sur le principe de la réponse aux questions, basé sur l'utilisation des techniques de TAL et de recherche d'information (RI). SQR est principalement concerné par la construction de systèmes robustes et scalable qui fournissent des réponses aux questions posées par les utilisateurs sous forme de langage naturel. Imaginez être dans un pays étranger, posant une question à votre assistant personnalisé dans votre téléphone en langage naturel pur, et obtenir une réponse similaire de celui-ci. C'est l'état idéal vers lequel travaillent les chercheurs et les technologues. Un certain succès dans ce domaine a été atteint avec des assistants personnalisés comme Siri et Cortana, mais leur portée est encore limitée car ils ne comprennent qu'un sous-ensemble de clauses et de phrases clés dans tout le langage naturel humain. [9]

II.4.2.4 Résumé de texte

Le but principal de le résumer de texte est de prendre un corpus de documents texte - qui pourrait être une collection de textes, de paragraphes ou de phrases - et de réduire le contenu de façon appropriée pour créer un résumé qui conserve les points clés de la collection. Le résumer peut-être effectuée en regardant les différents documents et en essayant de trouver les mots-clés et les phrases qui ont une importance importante dans l'ensemble de la collection. Deux principaux types de techniques de résumer de texte comprennent le résumer par extraction et le résumer par abstraction. Avec l'arrivée d'énormes quantités de texte et de données non structurées, le besoin de résumer le texte pour obtenir rapidement des informations précieuses est très demandé. [9]

II.4.2.5 Catégorisation du texte

Le but principal de la catégorisation de texte est d'identifier à quelle catégorie ou classe un document spécifique doit être placé en fonction du contenu du document. C'est l'une des applications les plus populaires de TAL et d'apprentissage numérique, car avec les bonnes données, il est extrêmement simple de comprendre les principes qui sous-tendent ses composants internes et de mettre en place un système de catégorisation de texte fonctionnel. Des techniques d'apprentissage automatique supervisées et non supervisées peuvent être utilisées pour résoudre ce problème, et parfois une combinaison

des deux est utilisée. Cela a permis de développer de nombreuses applications efficaces et pratiques, notamment des filtres anti-spam et la catégorisation des articles d'actualité. [9]

II.4.3 Les étapes d'analyses dans le TAL

Traditionnellement, le travail dans le traitement du langage naturel a tendance à considérer le processus d'analyse du langage comme étant décomposable en plusieurs étapes, reflétant les distinctions linguistiques théoriques établies entre SYNTAX, SEMANTICS et PRAGMATICS. La vue simple est que les phrases d'un texte sont d'abord analysées en fonction de leur syntaxe ; ceci fournit un ordre et une structure qui se prêtent mieux à une analyse en termes de sémantique ou de signification littérale; et ceci est suivi d'une étape d'analyse pragmatique où la signification de l'énoncé ou du texte dans le contexte est déterminée. Cette dernière étape est souvent considérée comme étant préoccupante avec DISCOURSE, alors que les deux précédentes concernent généralement les questions de forme [14]

Les étapes sont les suivants :

- Prétraitement de texte
- Analyse lexicale
- Analyse syntaxique
- Analyse sémantique
- Analyse pragmatique

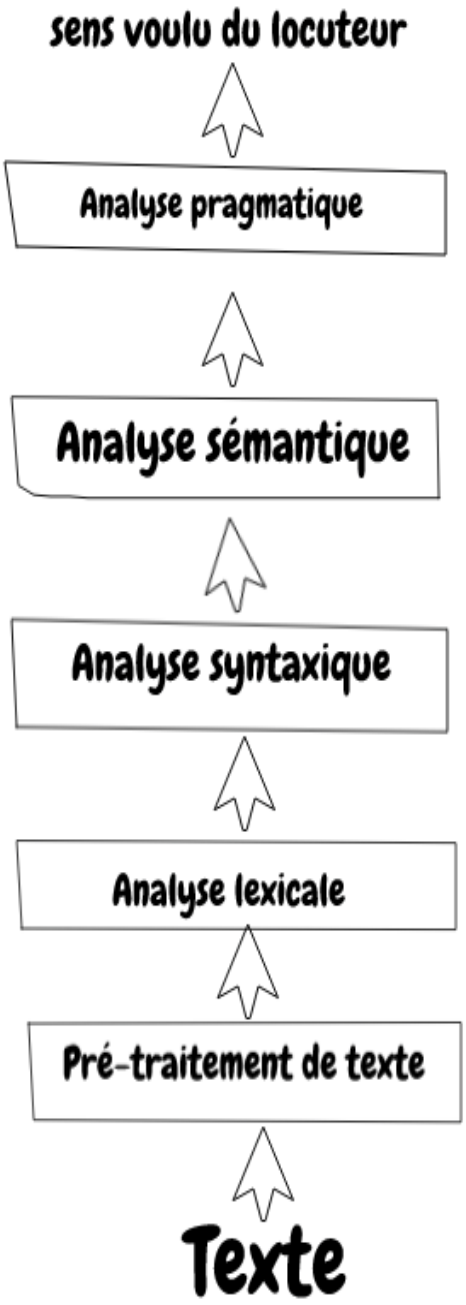


Figure II.4-1:Les étapes d'analyse dans le TAL

II.4.3.1 Prétraitement de texte

Tous les algorithmes d'apprentissage automatique (ML), qu'il s'agisse de techniques supervisées ou non supervisées, fonctionnent généralement avec des entités en entrée de nature numérique. Bien qu'il s'agisse d'un sujet distinct sous Génie des fonctionnalités, que nous explorerons en détail, pour y parvenir, vous devez nettoyer, normaliser et prétraiter les données textuelles initiales. Habituellement, les corpus de texte et autres données textuelles dans leur format brut natif ne sont pas bien formatés et normalisés, et bien sûr, nous devrions nous attendre à cela - après tout, les données de texte sont très déstructurées ! Le traitement de texte, ou pour être plus précis, le prétraitement, implique l'utilisation de diverses techniques pour convertir le texte brut en séquences bien définies de composants linguistiques qui ont une structure et une notation standard. [9]

Souvent, des métadonnées supplémentaires sont également présentes sous la forme d'annotations pour donner plus de sens aux composants du texte, comme les balises. La liste suivante nous donne une idée de certaines des techniques de prétraitement de texte les plus populaires utilisées dans TAL : [9]

- Tokinisation
- Tagging
- Chunking
- Stemming
- Lemmatisation

II.4.3.2 Analyse lexicale

L'étape précédente a abordé le problème de décomposer un texte en mots et phrases qui seront soumis à un traitement ultérieur. Les mots, bien sûr, ne sont pas atomiques et sont eux-mêmes ouverts à une analyse plus poussée. Ici nous entrons dans les domaines de la morphologie computationnelle, l'objet du chapitre d'Andrew Hippisley. En séparant les mots, nous pouvons découvrir des informations qui seront utiles aux étapes ultérieures du traitement. La combinatoire signifie également que la décomposition des mots dans leurs parties, et le maintien des règles de formation des combinaisons, est beaucoup plus efficace en termes d'espace de stockage que si nous classions simplement chaque mot comme un élément atomique dans un énorme inventaire. Et, revenant une fois de plus à notre préoccupation avec la manipulation de vrais textes, il y aura toujours des mots manquants dans un tel inventaire ; Le traitement morphologique peut permettre de traiter de tels mots non reconnus. Hippisley fournit une revue détaillée et détaillée des techniques pouvant être utilisées pour effectuer un traitement morphologique, en s'appuyant sur des exemples tirés de langues autres que l'anglais

pour démontrer le besoin de méthodes de traitement sophistiquées ; en cours de route, il donne un aperçu des aspects théoriques pertinents de la phonologie et de la morphologie. [14]

II.4.3.3 Analyse syntaxique

Un présupposé dans la plupart des travaux sur le traitement du langage naturel est que l'unité fondamentale de l'analyse du sens est la phrase : une phrase exprime une proposition, une idée ou une pensée et dit quelque chose sur un monde réel ou imaginaire. Extraire le sens d'une phrase est donc une question clé. Les phrases ne sont cependant pas de simples séquences linéaires de mots, et il est donc largement reconnu que pour mener à bien cette tâche, il faut une analyse de chaque phrase, qui détermine sa structure d'une manière ou d'une autre. Dans les approches TAL basées sur la linguistique générative, ceci implique généralement la détermination de la structure syntaxique ou grammaticale de chaque phrase. Dans leur chapitre, Ljunglöf et Wirén présentent une gamme de techniques pouvant être utilisées à cette fin. Ce domaine est probablement le mieux établi dans le domaine du TAL, permettant aux auteurs de fournir un inventaire des concepts de base dans l'analyse, suivi d'un catalogue détaillé des techniques d'analyse qui ont été explorées dans la littérature. [14]

II.4.3.4 Analyse sémantique

Identifier la structure syntaxique d'une séquence de mots n'est qu'une étape dans la détermination de la signification d'une phrase ; il fournit un objet structuré plus facilement manipulable et interprété ultérieurement. Ce sont ces étapes suivantes qui dérivent le sens de la phrase en question. Le chapitre de Goddard et Schalley se tourne vers ces problèmes plus profonds. C'est ici que nous commençons à atteindre les limites de ce qui a été jusqu'ici étendu du travail théorique à l'application pratique. Comme indiqué plus haut dans cette introduction, la sémantique du langage naturel a été moins étudiée que les problèmes syntaxiques, et donc les techniques décrites ici ne sont pas encore développées dans la mesure où elles peuvent facilement être appliquées dans une large couverture. [14]

Après avoir défini la scène en examinant une gamme d'approches existantes de l'interprétation sémantique, Goddard et Schalley fournissent une exposition détaillée du Métanguisage sémantique naturel, une approche de la sémantique qui est susceptible d'être nouvelle pour beaucoup travaillant dans le traitement du langage naturel. Ils finissent par cataloguer certains des défis à relever si nous voulons développer des analyses sémantiques de couverture vraiment larges. [14]

II.4.3.5 Analyse pragmatique

Après l'analyse sémantique, la prochaine étape du traitement porte sur la pragmatique. Malheureusement, il n'y a pas de distinction universellement acceptée entre la sémantique et la pragmatique. Nous, en commun avec plusieurs autres auteurs [Russel & Norvig (c)] fait la distinction comme suit :

L'analyse sémantique associe le sens à des énoncés / phrases isolés; l'analyse pragmatique interprète les résultats de l'analyse sémantique du point de vue d'un contexte spécifique (le contexte du dialogue ou de l'état du monde, etc.). Cela signifie qu'avec une phrase comme « Le grand chat a chassé le rat », l'analyse sémantique peut produire une expression qui signifie le grand chat, mais ne peut pas effectuer l'étape supplémentaire d'inférence nécessaire pour identifier le grand chat comme Félix. Cela serait laissé à l'analyse pragmatique. Dans certains cas, comme dans l'exemple qui vient d'être décrit, l'analyse pragmatique correspond simplement à des objets / événements réels qui existent dans un contexte donné avec des références d'objets obtenues lors de l'analyse sémantique. Dans d'autres cas, l'analyse pragmatique peut désambiguïser des phrases qui ne peuvent être complètement désambiguïsées durant les phases de syntaxe et d'analyse sémantique.

II.5 Ontologie

II.5.1 Introduction

Historiquement, l'ontologie est une discipline de la philosophie qui a pour objet l'étude systématique de la nature et de l'organisation de l'être. Apparu dans son acception informationnelle il y a une dizaine d'années, dans le domaine de l'ingénierie des connaissances et de l'intelligence artificielle, ce terme désigne les « artefacts » élaborés dans le cadre d'une modélisation conceptuelle apte à jouer un rôle de référentiel conceptuel. Les travaux sur les ontologies se sont plus particulièrement développés dans un contexte informatique et ont pris leur essor avec le web sémantique.

Une ontologie fournit le vocabulaire spécifique à un domaine de la connaissance et, selon un degré de formalisation variable, fixe le sens des concepts et des relations qui les unissent. L'article publié en 1996 par M. Uschold et M. Gruninger [15] reste à notre sens le texte fondateur sur les ontologies. On y trouve cette définition : « Il s'agit du terme utilisé se référant à la compréhension partagée (à shared understanding) d'un domaine d'intérêt qui peut être utilisé comme cadre unificateur pour résoudre les problèmes de communication entre les gens et d'interopérabilité entre les systèmes. » Les composantes d'une ontologie sont les suivantes : une ou plusieurs taxinomies ordonnées en classes et sous-classes composées d'instances représentant les individus ou objets; les types d'attributs ou propriétés qui peuvent être attachés à ces objets; les types de relations entre les concepts d'une taxinomie; des axiomes ou des règles d'inférence permettant de définir les propriétés de ces relations.

Le développement des ontologies s'est fait parallèlement à celui de la notion de métadonnée. Pour être susceptibles d'être exploitées automatiquement, les métadonnées doivent être entièrement explicites et exprimées dans un vocabulaire clairement et formellement défini. Les ontologies sont le réceptacle de ces définitions.

On y représente les « valeurs » que l'on peut donner aux métadonnées et l'interprétation que les systèmes peuvent en faire, c'est à dire les concepts d'un domaine, les relations qu'ils entretiennent, la sémantique de ces relations et les règles de raisonnement qui leur sont applicables. [16]

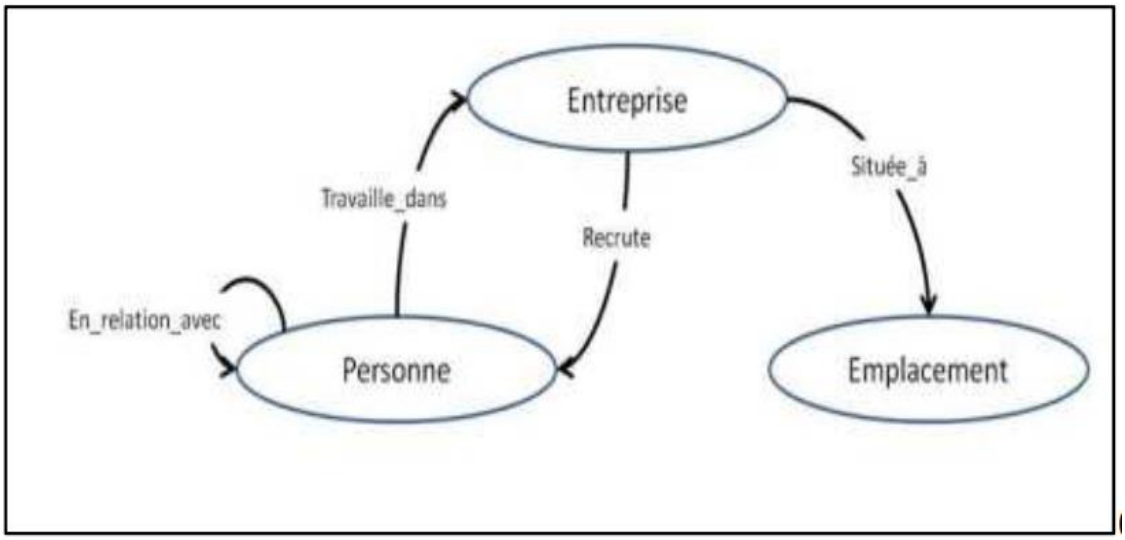


Figure II.5-1 exemple

II.5.2 Rôles des ontologies

II.5.2.1 Modularité et réutilisation des connaissances

Gruber [17]insistait sur le rôle que pouvaient tenir les ontologies pour favoriser la modularité et la réutilisation dans les systèmes informatiques. En effet, ces ontologies permettent l'étude de conceptualisations, indépendamment du formalisme choisi pour les représenter et doivent être définies indépendamment du langage utilisé pour la programmation des applications, de la plate-forme utilisée et des protocoles de communication (protocoles réseaux).

II.5.2.2 Communication

Philippe Martin [18] propose d'aider les spécialistes de l'ingénierie de la connaissance, en utilisant la terminologie définie dans Word Net comme base de la communication, car c'est un standard.

Dans le domaine pédagogique c'est la communication entre auteurs et informaticiens qui est parfois difficile, d'où l'intérêt d'utiliser des ontologies dans les environnements auteur pour la définition d'un vocabulaire convivial et précis dans la définition des tâches pédagogiques [19]. L'ontologie joue alors le rôle d'un méta-modèle.

Les ontologies peuvent également être utilisées pour harmoniser la communication entre différentes applications où entre différents agents [20].Cette idée, également sous-jacente dans les publications de Gruber [21], repose souvent sur une

ontologie du domaine. Pourtant Mizoguchi [22]veut aller plus loin en dotant les agents d'une connaissance sur une ontologie de tâche indépendante du domaine.

II.5.3 Wordnet

Def 1.1 : WordNet est une base de données lexicale pour l'anglais organisée conformément aux théories psycholinguistiques actuelles. Les concepts lexicaux sont organisés par des relations sémantiques (synonymie, antonyme, hyponymie,..etc.) pour les noms, les verbes et les adjectifs. [23]

Def2.2 : WordNet est un grand réseau sémantique reliant des mots et des groupes de mots au moyen de relations lexicales et conceptuelles représentées par des arcs étiquetés. Les blocs de construction de WordNet sont des ensembles de synonymes (synsets), des ensembles non ordonnés de mots et de phrases cognitivement synonymes (Cruse, 1986). Chaque membre d'un synset donné exprime le même concept, bien que tous les membres de synset ne soient pas interchangeables dans tous les contextes. Les exemples sont {car, automobile}, {hit, strike} et {big, large}. Tous les synsets contiennent en outre une brève définition, et la plupart incluent une ou plusieurs phrases illustrant l'utilisation des synonymes. Un label de domaine (sport, médecine, biologie) marque de nombreux synsets. L'appartenance commune des mots dans un synset donné illustre le phénomène de la synonymie. L'appartenance d'un mot à plusieurs synonymes reflète la polysémie ou la multiplicité des significations de ce mot. Ainsi, tronc apparaît dans WordNet dans plusieurs synsets différents, y compris {trunk, tree trunk}, {trunk, torso}, et{trunk, proboscis} [10]

II.5.3.1 Le contenu de wordnet

WordNet se compose de quatre composants distincts, chacun contenant des synsets avec des mots des catégories majeures, openclass, syntaxiques: noms, verbes, adjectifs et adverbes. WordNet 2.1 contient près de 118 000 synsets, comprenant plus de 81 000 synsets nominaux, 13 600 synsets verbaux, 19 000 synsets adjectifs et 3 600 synsets adverbes. [24]

Chapitre III : L'état de l'art

III.1 Introduction

De nombreuses approches ont été proposées pour identifier les correspondances entre les activités des modèles de processus métiers. Nous présentons un bref aperçu des outils qui ont participé à la compétition de processus métiers de l'année 2015. [2]

Dans ce chapitre, nous donnons un aperçu des approches d'alignement de processus métiers participants. Au total, 12 techniques d'alignement ont participé à la compétition d'alignement de processus métiers. Le tableau suivant donne un aperçu des approches participantes et des auteurs respectifs. Dans les sous-sections suivantes, nous fournissons un bref aperçu technique de chaque approche d'alignement. [2]

NO.	Approche	Auteurs
1	AML-PM	Marzieh Bakhshandeh, Joao Cardoso, Goncalo Antunes, Catia Pesquita, Jose Borbinha
2	BPLangMatch	Eitam Sheetrit, Matthias Weidlich, Avigdor Gal
3	KnoMa-Proc	Mauro Dragoni, Chiara Di Francescomarino, Chiara Ghidini
4	Know-Match-SSS (KMSSS)	Abderrahmane Khiat
5	Match-SSS (MSSS)	Abderrahmane Khiat
6	RefMod-Mine/VM 2 (RMM/VM2)	Sharam Dadashnia, Tim Niesen, Philip Hake, Andreas Son -ntag, Tom Thaler, Peter Fettke, Peter Loos
7	RefMod-Mine/NHCM (RMM/NHCM)	Tom Thaler, Philip Hake, Sharam Dadashnia, Tim Niesen, Andreas Sonntag, Peter Fettke, Peter Loos
8	RefMod-Mine/NLM (RMM/NLM)	Philip Hake, Tom Thaler, Sharam Dadashnia, TimNiesen, Andreas Sonntag, Peter Fettke, Peter Loos
9	RefMod-Mine/SMSL (RMM/SMSL)	Andreas Sonntag, Philip Hake, Sharam Dadashnia, TimNiesen, Tom Thaler, Peter Fettke, Peter Loos
10	OPBOT	Christopher Klinkmüller, Ingo Weber
11	pPalm-DS	Timo Péus
12	TripleS	Andreas Schoknecht

Table III.1-1 Les approches [2]

III.2 Les approches

III.2.1AML-PM

The AgreementMakerLight (AML) est un système d'alignement des ontologies qui a été optimisé pour gérer l'alignement de plus grandes ontologies. Il a été conçu avec la flexibilité et l'extensibilité à l'esprit, et permet ainsi l'inclusion de pratiquement n'importe quel algorithme d'alignement. AML contient plusieurs algorithmes d'alignement basés à la fois sur les propriétés lexicales et structurelles, et prend également en charge l'utilisation de ressources externes et la réparation d'alignement. Ces fonctionnalités ont permis à AML d'atteindre les meilleurs résultats sur plusieurs pistes OAEI 2013 et 2014. Cependant, AML fonctionne sur les ontologies OWL, il était donc nécessaire de prétraiter les données d'entrée et de les traduire en OWL. Ensuite, un pipeline d'alignement a été appliqué qui comprenait plusieurs alignements lexicaux et une étape d'optimisation de similarité globale pour arriver à un alignement final. [2] [25]

III.2.2BPLangMatch

Cette technique d'alignement est adaptée aux modèles de processus qui comportent des descriptions textuelles des activités. Introduit en détail dans [26] En utilisant des idées issues de la modélisation du langage dans la recherche d'information, l'approche utilise ces descriptions pour identifier les correspondances entre les activités. Plus précisément, ils combinent deux flux de travail différents sur la modélisation du langage probabiliste. Tout d'abord, ils adoptent la modélisation basée sur le passage de telle sorte que les activités sont des passages d'un document représentant un modèle de processus. Deuxièmement, ils considèrent les caractéristiques structurelles des modèles de processus par la modélisation du langage positionnel. En combinant ces aspects, ils dépendent sur un nouveau modèle de langage basé sur le passage positionnel pour créer une matrice de similarité. Les scores de similarité sont ensuite adaptés en fonction des informations sémantiques dérivées par l'étiquetage de Part-Of-Speech. [2]

III.2.3KnoMa-Proc

Le système KnoMa-Proc proposé traite le problème d'alignement de modèle de processus d'une manière originale. Il met en œuvre une approche basée sur l'utilisation de techniques de recherche d'information (RI) pour découvrir les correspondances entre les entités du modèle de processus. L'utilisation de solutions basées sur la recherche d'information pour des entités basées sur l'alignement basées sur la connaissance est une

tendance récente qui a déjà montré des résultats prometteurs dans le domaine de l'alignement d'ontologies et dans le processus correspondant.

L'idée du travail est basée sur la construction et l'exploitation d'une représentation structurée de l'entité à mapper et de son « contexte », à partir des informations textuelles associées. Dans le cas des ontologies, la notion de « contexte » fait référence à l'ensemble des concepts directement liés (via une propriété «is-a») au concept à mapper, ou qui en sont distants (en termes de « is-a » "relations à traverser") inférieur à un certain degré. Lors de l'examen des processus, la sémantique du « contexte » doit être révisée. [2]

III.2.4Match-SSS and Know-Match-SSS

Le système Match-SSS (MSSS) utilise des techniques NLP pour normaliser les descriptions d'activité des deux modèles à faire correspondre. Il utilise d'abord des algorithmes basés sur des String et WordNet. Enfin, l'approche sélectionne les similarités calculées par ces deux aligneurs en fonction d'une stratégie maximale avec un seuil permettant d'identifier des activités équivalentes. Le système Know-Match-SSS (KMSSS) est similaire, mais utilise une autre technique basée sur la catégorie de mots. [2]

III.2.5RefMod-Mine/VM 2

L'approche RefMod-Mine / VM 2 d'alignement de processus métiers présentée ci-dessous est un affinement de leur concept décrit dans [27]. Il se concentre sur les étiquettes d'un modèle de processus pour déterminer les correspondances entre les activités en fonction de leur similarité textuelle. Par conséquent, les techniques établies dans le domaine de la recherche d'information sont combinées avec le traitement du langage naturel (TAL) pour d'informations provenant des statistiques textuelles. En guise d'étape préparatoire, chaque modèle à comparer est importé et transformé en un format de modèle générique, où les importateurs de BPMN, EPC et Petri-Nets sont fournis. Comme la notion de matches distincts 1 : 1 - i. e. une étiquette de nœud d'un modèle A ne peut pas être mappée à plus d'une étiquette de nœud à partir d'un modèle B - est sous-jacente, plusieurs correspondances possibles sont supprimées du mappage final en tant que dernière étape. [2]

III.2.6RefMod-Mine/NHCM

Cet outil améliore l'approche RefMod-Mine / NSCM présentée au PMC 2013 et se compose de 3 phases générales. Dans la phase de prétraitement les modèles d'entrée

sont transformés en un format générique, ce qui permet une application de l'approche de correspondance aux modèles de différents langages de modélisation. Dans la phase de traitement, tous les modèles disponibles d'un ensemble de données sont utilisés comme entrée pour l'aligneur de cluster n-aire, qui utilise une mesure de similarité basée sur le TAL pour une comparaison de nœuds par paire. En conséquence, plusieurs ensembles de clusters contenant des nœuds de tous les modèles considérés sont produits, qui sont ensuite extraits vers des mappages complexes binaires entre deux modèles. Enfin, les mappages complexes binaires sont en cours de post-traitement afin d'éliminer les alignements non correspondants issues des clusters.

La technique a été implémentée sous la forme d'un outil de ligne de commande PHP et peut être consultée publiquement sur https://github.com/tomson2001/refmodmine. Il est également disponible en tant qu'outil en ligne dans le contexte du RefMod-Miner as a Service à l'adresse http://rmm.dfki.de. [2]

III.2.7RefMod-Mine/NLM

The Natural Langage Matcher (NLM) identifie les étiquettes de modèle de processus correspondantes et, par conséquent, les nœuds correspondants. Il est principalement basé sur des techniques de traitement du langage naturel utilisant un concept de sac de mots. Contrairement à l'approche actuelle de sac de mots [28].le NLM utilise la classification des mots. L'aligneur est capable d'identifier des correspondances simples ainsi que des correspondances complexes entre deux modèles de processus. Puisque l'approche repose principalement sur les étiquettes utilisées dans les modèles de processus, elle peut être appliquée à tout type de langage de modélisation de processus. L'aligneur est implémenté en Java 1.8 et intégré dans le jeu d'outils RefMod-Mine 7. [2]

III.2.8RefMod-Mine/SMSL

RefMod-Mine / SMSL est un algorithme de correspondance sémantique basé sur une approche d'apprentissage numérique supervisée. L'approche se compose de deux étapes :

D'abord l'algorithme est donné un référentiel de modèles de processus et de son gold standard. L'algorithme identifie les marques des étiquettes de processus et détermine leur catégorie grammaticale (verbe, nom, ...). Ensuite, il effectue une recherche de mots liés sémantiquement dans le Wordnet [23].Comme mesure de la quantification de la relation sémantique de deux mots, on utilise une fonction composée qui dépend de la

distance sémantique entre les deux mots et les mots intermédiaires dans Wordnet. Toutes les étiquettes et la distance sémantique sont pondérées. Lorsque l'algorithme a calculé toutes les relations sémantiques comme d'alignements, il stocke tous les poids de la fonction et la précision atteinte, le recall et la F-mesure. Ces poids sont ensuite optimisés par la F-mesure résultante dans une recherche locale.

Lorsque les poids ont été stockés / appris, l'algorithme applique les poids les mieux trouvés sur de nouveaux alignements donnés. [2] [29]

III.2.9OPBOT

La technique de préservation de l'ordre de sac à mots (OPBOT) repose sur deux piliers : l'amélioration de l'alignement des étiquettes et la préservation des ordres. Pour améliorer l'efficacité de l'alignement des étiquettes, ils identifient d'abord les activités labellisées en égalité, puis ils réduisent le niveau de détail des étiquettes restantes. Le premier est motivé par l'observation que les activités également étiquetées constituent le plus souvent des correspondances 1 : 1. Ce dernier s'appuie sur leurs travail précédent [28] où l'élagage d'étiquettes a été utilisé pour augmenter le rappel. Ici, ils emploient leur similarité de sac-de-mots d'élagage maximum (MPB) qui a bien fonctionné dans l'itération précédente du compétition d'alignement [30]. La préservation d'ordres repose sur l'idée que les correspondances multiples entre deux modèles se produisent dans le même ordre dans les deux modèles. Pour ce faire, ils utilisent the relative start node distance (RSD) [31]. OPBOT traite toutes les paires de modèles en une seule fois. Il est basé sur le workflow de correspondance général pour la correspondance de schéma. [32] [2].

III.2.10 pPalm-DS

Avec cette approche, ils fournissent une base rudimentaire pour l'alignement des processus, en se concentrant sur la recherche d'un alignement entre les activités (nœuds) avec des étiquettes sémantiques similaires. Ils ne considèrent pas les informations telles que la structure, le comportement ou les différents types de nœuds dans les modèles de processus. En un mot, à partir de chaque processus p, ils récupèrent l'ensemble des nœuds pertinents, appelés ci-après activités (types de nœuds utilisés pour l'alignement dans le Gold Standard correspondant). A partir de l'ensemble des activités, ils obtiennent l'ensemble d'étiquettes correspondant I ∈ L p. Pour calculer l'alignement d'une paire de processus (p 1, p 2), ils comparent chaque étiquette l ∈ L p 1 à chaque étiquette l 0 ∈ L p

2 par une fonction de similarité. Si la similarité des deux labels est égale ou supérieure à un certain seuil, (sim (l, l 0) ≥ seuil), ils incluent la paire d'activités correspondante dans l'ensemble des correspondances. [2]

III.2.11 TripleS

L'approche d'alignement utilisée dans la deuxième compétition d'alignement de modèles de processus en 2015 est essentiellement la même que celle utilisée en 2013. L'approche d'alignement Triple-S [33]adhère toujours au principe KISS en évitant les techniques d'alignement complexes et en les gardant simples et stupides. Leur version du 2015 a été étendue pour correspondre non seulement aux transitions dans Petri-Nets mais aussi aux fonctions des modèles EPC et aux tâches des modèles en notation BPMN, c'est-à-dire que les composants "actifs" des modèles de processus sont appariés. [2] [33]

Chapitre IV : L’approche

IV.1 L'approche proposée

Dans ce chapitre on va expliquer l'approche proposée. Cette approche est illustrée dans la figure (Fig IV.1-1) ci-dessous. L'approche proposée est composée de 4 étapes : extractions des activités, prétraitement, construction de la matrice de similarité et filtrage de résultats.

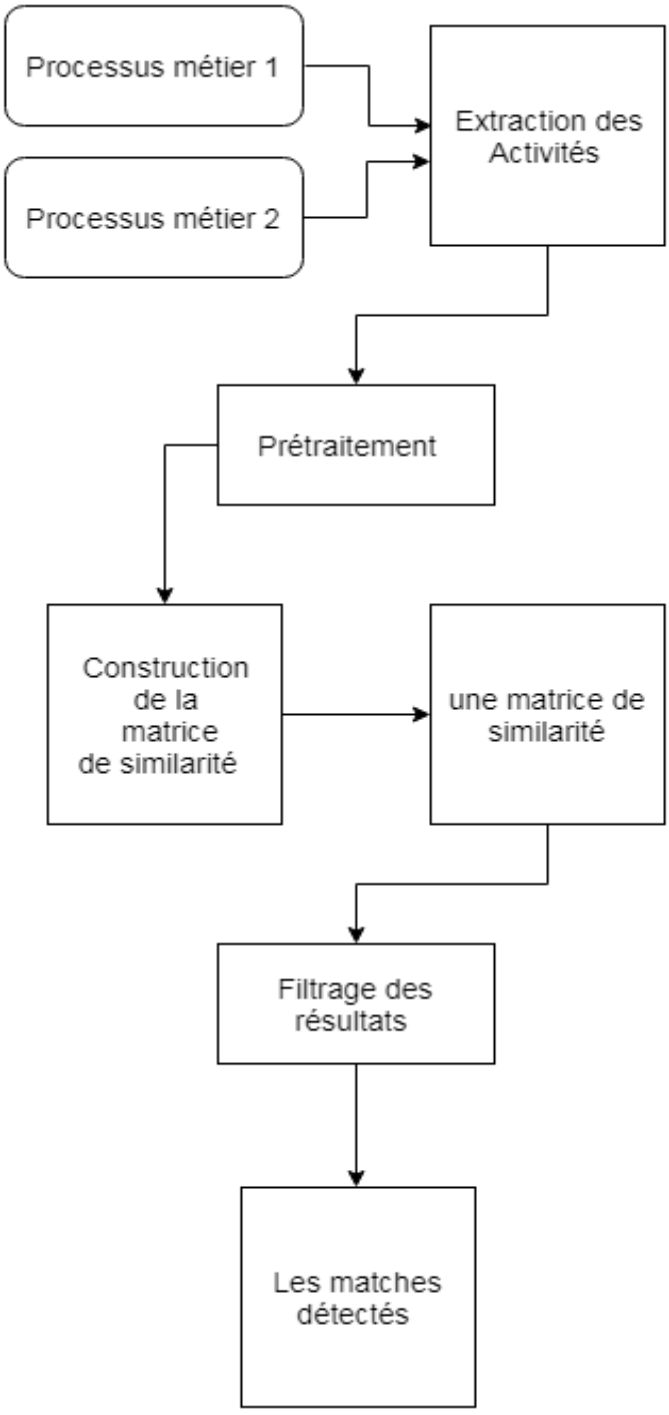


Figure IV.1-1 L'approche proposée

IV.1.1 Extraction des Activités

Premièrement, on va faire l'extraction de toutes les informations des activités qui sont disponibles dans les benchmarks sous la forme textuelle (BPMN, EPML et EPC). Exemple :

Voici un exemple d'un fichier retenu du premier benchmark (University Admission processes) sous la forme BPMN illustré dans la figure ci-dessous.

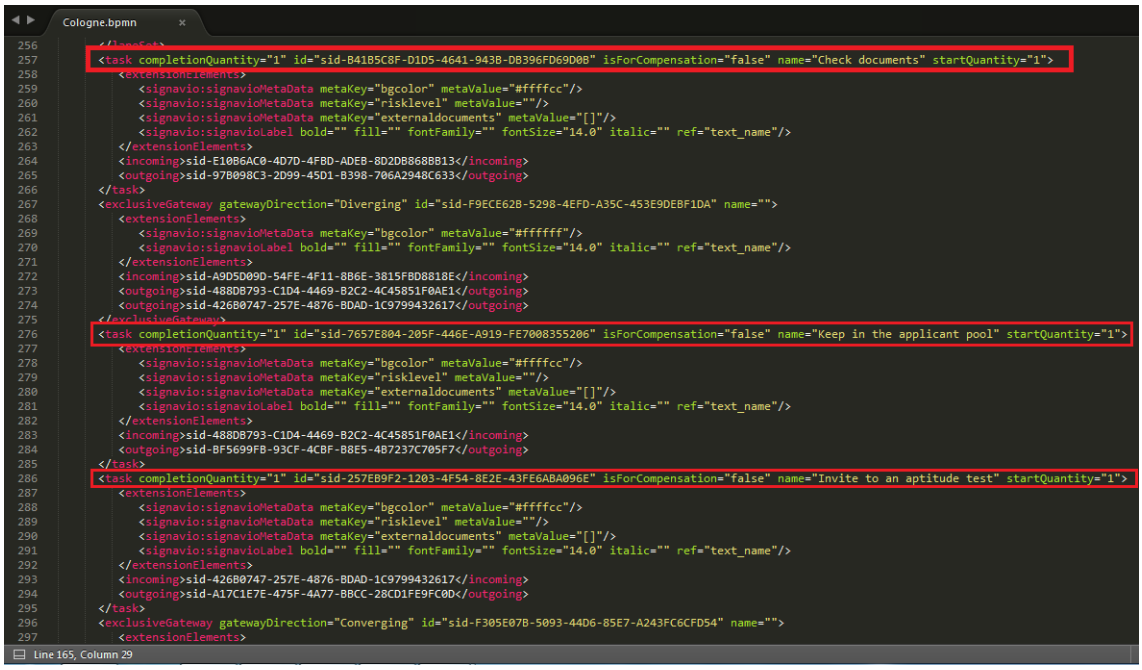


Figure IV.1-2 UA exemple

Les données extraites depuis ces fichiers sont les identifions des activités et la description textuelle de ces activités. Exemple :

Les données d'activité 01 :

L'identifiant = {sid-B41B5C8F-D1D5-4641-943B-DB396FD69D0B}

La description textuelle = {Check, documents}

Les données d'activité 02 :

L'identifiant = {sid-7657E804-205F-446E-A919-FE7008355206}

La description textuelle = { Keep, in, the, applicant, pool }

IV.1.2 Prétraitement

On applique un prétraitement basé sur :

- Tokenisation des phrases à des mots.
- Suppression des mots vide (stop liste).
- Racinisation des mots (Stemming); pour ce dernier on a utilisé un algorithme fameux; PorterStemmer (Snowball) qui fait la racinisation.

Exemple (activité extraite à partir du 1er benchmark) :
{Rank students according to GPA and the test results}

- Tokenization :

[Rank, Students, according, to,GPA ,and , the , test , results]

- Suppression des mots vide :

[Rank, Students, according ,GPA,test,results]

- Racinisation (Porterstemer) :

[Rank, student, accord,GPA , test , result]

IV.1.3 Construction de la matrice de similarité

La matrice de similarité stocke les scores de similarité entre tous les pairs d'activités des deux modèles de processus métiers. Le calcul du score de similarité se fait en utilisant l'algorithme suivant :

L'algorithme prend en entrée deux descriptions textuelles de deux activités, une appartient au premier modèle de PM et l'autre au deuxième et retourne une valeur qui représente le degré de similarité entre ces deux activités. Cette valeur est calculée selon la démarche suivante :

Premièrement, on commence par la génération de toutes les descriptions équivalentes (combinaisons) de la première activité en utilisant les synonymes (Synsets) des mots qui composent cette description. Les synonymes (Synsets) de chaque mot sont tirés à partir de Wordnet . Généralement, si une description contient n mots et chaque mot

à m synsets en moyenne, alors le nombre total des descriptions équivalentes est de l'ordre de n puissance m.

Par exemple étant donnée les synsets suivantes des mots composant la description normalisée [Rank, student, accord, GPA, test, result]:

- Rank : rate, rank, range, order, grade, place
- Student : scholar, scholarly person, bookman, student
- Accord : agreement, accord, conformity, accordance

Les descriptions équivalentes produites sont:

1. [Rate, student, accord, GPA, test, result]
2. [Range , student, accord, GPA, test, result]
3. [Rank , student, accord, GPA, test, result]
4. [order , student, accord, GPA, test, result]
5. [palce , student, accord, GPA, test, result]
6. [Rank , scola r, accord, GPA, test, result]
7. [Rank , scholary, accord, GPA, test, result]
8. ...etc

Deuxièmement, chaque description équivalente est comparée à la deuxième description appartenant au deuxième process model en utilisant l'algorithme Levenshtein qui calcul le nombre de mots à insérer, ou à modifier ou à supprimer pour transformer la première description en deuxième.

Pour obtenir un score de similarité entre la première et la deuxième description. La plus petite valeur parmi toutes les valeurs obtenues en comparant toutes les descriptions équivalentes au deuxième activité (i.e., distance minimale) sera retenu, comme score de similarité entre la première activité et la deuxième, après normalisation selon la formule suivante:

Min = (1- (distance minimal/(nombre du terme de deux phrases))

La fonction de normalisation Min produit des valeurs entre [0...1], quand Min(activity1, activity2) = 1 alors activity1 est totalement équivalent à activity2, quand Min(activity1, activity2) = 0 alors activity1 est totalement diffèrent de activity2, sinon ils sont similaires à un certain degré, Dans la dernière étape, il suffit de fixer un seuil pour décider si deux activités sont similaires.

IV.1.4 Filtrage des résultats et la détection des correspondances

Deux activités A1 et A2 ou a1 appartient au premier pm et A2 appartient au deuxième pm sont considérées comme équivalentes si leurs scores de similarité sont supérieurs à certain seuil S. ce seuil est fourni par l'utilisateur.

Chapitre V : L'expérimentation

V.1 L'expérimentation

L'étude est réalisée par un outil écrit en langage python, qui va automatiser toutes les étapes de notre approche proposée, il contient des programmes qui extraite les informations des fichiers XML de processus métiers (i.e., epml, BPMN and EPC), et construire la matrice de similarité qui contient les similarités entre les activités de modèles de processus métiers, cette procédure trouve des bons matchs entre les paires de processus métiers dans chaque Benchmark.

V.1.1 Les Benchmarks

L'étude a utilisée trois Benchmarks de problème d'alignement de modèles de processus :

1. **University Admission Processes (UA) :** Ce Benchmark se compose de 36 paires de modèles qui ont été dérivés de 9 modèles représentant la procédure d'inscription pour les étudiants de Master de neuf universités allemandes.
2. **Birth Registration Processes (BR) :** Ce Benchmark comprend 36 paires de modèles dérivées de 9 modèles représentant les processus d'enregistrement des naissances en Allemagne, en Russie, en Afrique du Sud et aux Pays-Bas.
3. **Asset Management (AM) :** ce Benchmark comprend 36 paires de modèles dérivées de 72 modèles de la collection de modèles de référence SAP.

Ces Benchmarks sont accompagnés d'un GoldStandard qui indique pour chaque paire de BPS dans un benchmark l'alignement correct. Cet GoldStandard est utilisé pour calculer la précision, le rappel et la f-mesure.

V.1.2 Le calcule de F mesure, Precision et Recall

Dans notre cas les vrais positifs sont les matchs détectés en tant que vrai et qui sont réellement vrai, et les faux positifs sont les matchs détectés en tant que vrai et qui sont réellement faux, et les faux négatifs sont les matchs détectés en tant que faux et qui sont réellement vrai.

- **Precision** = vrai positif / (vrai positif + faux positif)
- **Recall** = vrai positif / (vrai positif + faux négatif)
- **F Mesure** = 2 * ((Precision * Recall) / (Precision + Recall)).

V.1.3 Résultats

Dans cette section on vous présente les résultats d'alignement pour les trois Benchmarks, les tables 1, 2 et 3 présente les moyennes valeurs de Precision, Recall et F-measure pour notre approche pour les Benchmarks UA, BR et AM.

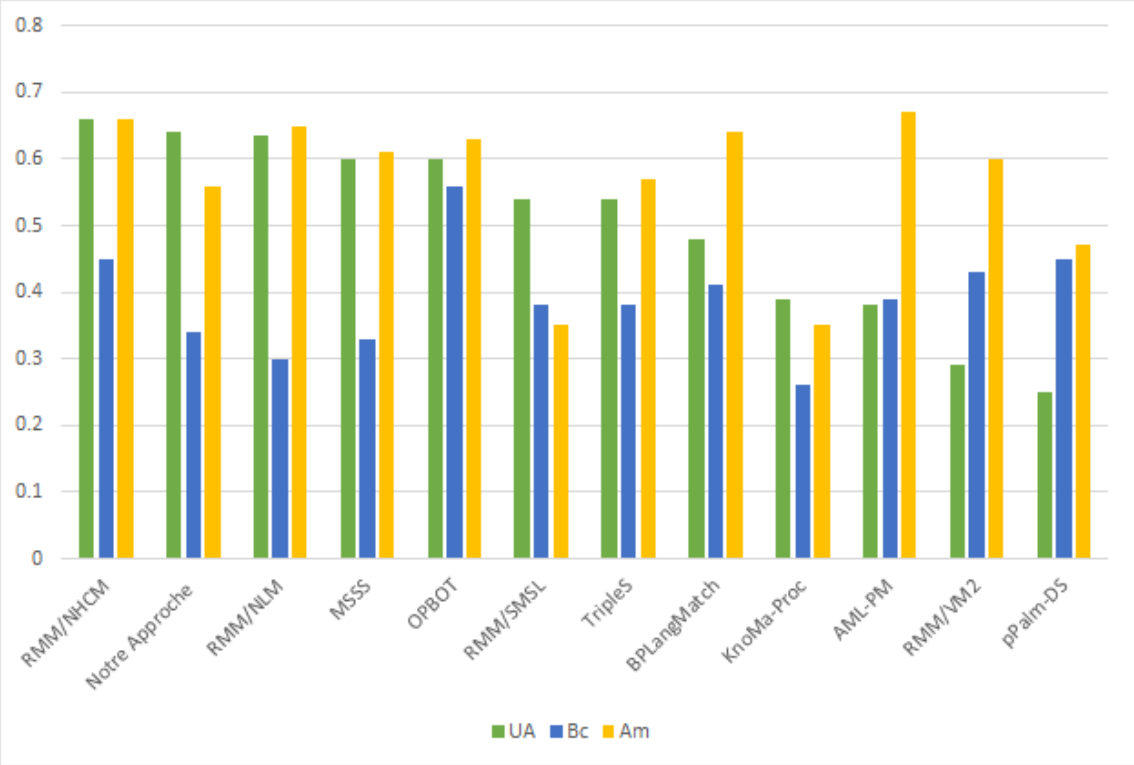


Figure V.1-1Comparaison avec les autres algorithmes de contests 2015

Les résultats de l'expérience indiquent que notre approche a atteint une F-measure moyenne de 0.64 (Tableau V.1-1) pour le Benchmark UA à un seuil de 0.90 et c'est une valeur très élever, en effet on a obtenu le deuxième meilleur résultat par rapport aux matchers qui ont participé au Process Model Matching Contest 2015.

Threshold	**Precision**	**Recall**	**F-Measure**
0.60	0.0792	**0.8556**	0.1449
0.66	0.1466	0.8350	0.2494
0.70	0.1931	0.6391	0.2965
0.75	0.2040	0.6288	0.3080
0.80	0.4849	0.5824	0.5291
0.85	0.7906	0.5257	0.6314
0.90	**0.8429**	0.5257	**0.6475**
0.95	**0.8429**	0.5257	**0.6475**
1.00	**0.8429**	0.5257	**0.6475**

Table V.1-1 Results of University Admission Matching

L'approche a atteint une F-measure de 0.34 (Table V.1-2) pour le Benchmark BR à un seuil de 0.80 et c'est aussi un résultat élever.

Threshold	Precision	Recall	F-Measure
0.60	0.2298	**0.4662**	0.3079
0.66	0.3030	0.3485	0.3242
0.70	0.3375	0.2897	0.3118
0.75	0.3386	0.2766	0.3045
0.80	0.6842	0.2265	**0.3404**
0.85	**0.9230**	0.1568	0.2681
0.90	0.9210	0.1525	0.2616
0.95	0.9210	0.1525	0.2616
1.00	0.9210	0.1525	0.2616

Table V.1-2 Results of Birth Certificate Matching

L'approche a atteint une F-measure de 0.56 (Tableau V.1-3) pour le Benchmark AM à un seuil de 0.85 et ce résultat est très élever. Par conséquent on peut conclure que notre approche a eu une moyenne F-measure de 0.51.

Threshold	Precision	Recall	F-Measure
0،6	**0،53**	0،21	0،30
0،66	0،50	0،32	0،39
0،7	0،47	0،45	0،46
0،75	0،47	0،51	0،49
0،8	0،45	0،60	0،51
0،85	0،51	0،85	**0،56**
0،90	0،41	0،86	**0،56**
0،95	0،39	**0،96**	0،55
1	0،39	**0،96**	0،55

Table V.1-3 Results of Asset Management Matching

On a remarqué aussi que notre approche a eu une Précision égale à 0.84 dans le cas du Benchmark UA et le meilleur Recall (0.96) dans le cas du Benchmark AM.

V.2 Conclusion

Ce travail a présenté une approche heuristique au but de résoudre le problème d'alignement de modèles de processus métiers. Notre approche a utilisé des techniques de TAL et d'ontologies (Wordnet) pour trouver des bons matchs entre les activités. Les résultats de notre travail qui ont été obtenus après l'application de l'approche sur trois benchmarks connus dans le domaine ont montré que notre approche a un grand potentielle d'effectuer des bons match pour les BPM. Dans les futures recherches on va concentrer sur des techniques avancer de TAL et d'ontologies pour améliorer notre approche.

Bibliographie

[1] M. Weske, Businesse process Mangement, Springer, 2007.

[2] Jens Kolb et al, «The Process Model Matching Contest 2015,» *(Eds.): Enterprise Modelling and Information Systems Architectures,*, pp. 127-155, bonn 2015.

[3] R. Stephens, Beginning Software Engineering, 2 mars 2015.

[4] R. J. Leach, Introduction to software engineering.

[5] P. David T. Bourgeois, Information Systems for Business and Beyond, Syalor foundations , 2014.

[6] E. C. Softeam, Le Guide Pratique des Processus Métiers, Softeam.

[7] M. B.-F. ,. Y. G. Chantal Morley, Processus metiers et Systeme dInformation, paris: Dunod , 2011.

[8] D. Sarker, Text Analytics with Python, india: apress, 2016.

[9] S. S. S. Bandyopadhyay, Unsupervised Classification, springer, 2013.

[10] teknomo, «distance for binary variables,» 2015. [En ligne]. Available: http://people.revoledu.com/kardi/tutorial/Similarity/BinaryVariables.html.

[11] GRAHAM CORMODE AT&T Labs–Research and S. MUTHUKRISHNAN Rutgers University, «The String Edit Distance Matching Problem with Moves».

[12] http://www.merriampark.com/mgresume.htm, «Levenshtein Distance, in Three Flavors,» [En ligne]. Available: https://people.cs.pitt.edu. [Accès le 02 06 2018].

[13] N. indurkhya et F. j.DAmerau, HANDBOOK OF NATURAL LANGUAGE PROCESSING, crc press, 2010.

[14] M. G. Mike Uschold, «Ontologies: Principles, Methods and Applications,» *Knowledge Engineering Review*, vol. 11, n° %12, pp. 93-136, 1996.

[15] A.D.B.S, «Langages documentaires et outils linguistiques,» *Documentaliste-Sciences de l'Information*, vol. Vol. 44, p. 120, 2007.

[16] T. R. GRUBER, «A Translation Approach to Portable Ontology Specifications,» *Knowledge Acquisition*, vol. 5, 1995.

[17] P. MARTIN, «the WordNet Concept Catalog and a Relation Hierarchy,» chez *international Workshop on Peirce Université* , Californie, Santa Cruz, USA., 18 aout 1995.

[18] Y. H. J. L. W. C. J. B. K. S. A. R. M. M. IKEDA, «An ontology more than a shared vocabulary,» chez *Ninth International Conference on Artificial Intelligence in Education, AI-ED'99*, Le Mans, France, 19-23 Juillet 1999.

[19] W. C. A. R. MIZOGUCHI, « Communication Content Ontology For Learner Model Agent in multi-Agent Architecture. Workshop on Ontologies for Intelligent Educational Systems,» chez *Ninth International Conference on Artificial Intelligence in Education,AI-ED'99*, Le Mans, France, 19-23 juilet 1999.

[20] T. R. GRUBER, «A Translation Approach to Portable Ontology Specifications,» *Knowledge Acquisition*, vol. 5, 1995.

[21] M. R, «A Step towards Ontological Engineering,» chez *the 12th National Conference on AI of JSAI*, japan, Juin 1998.

[22] P. I. George A. Miller, «WORDNET: A LEXICAL DATABASE FOR ENGLISH,» Princeton University , Princeton, New Jersey 08544 .

[23] F. C, « WordNet(s),» *Encyclopedia of Language & Linguistics Second Edition,* vol. 13, pp. 665-670, 2006.

[24] Daniel Faria, Catia Pesquita, Emanuel Santos, Isabel F. Cruz, Francisco M. Couto, «AgreementMakerLight Results for OAEI 2013,» n° %1Department of Computer ScienceUniversity of Illinois at Chicago, pp. 101-108, 2013.

[25] Weidlich, Matthias; Sheetrit, Eitam; Branco, Moises; Gal, Avigdor:, «Matching Business Process Models Using Positional Language Models,» 32nd International Conference on Conceptual Modeling, ER 2013, Hong Kong, 2013.

[26] Niesen, T.; Houy, C, «Zur Nutzung von Techniken der Natürlichen Sprachverarbeitung für die Bestimmung von Prozessmodellähnlichkeiten,» *springer,* p. 1829–1843, 2015.

[27] Klinkmüller, Christopher ; Weber, Ingo ; Mendling, Jan; Leopold, Henrik; Ludwig, André, «Increasing Recall of Process Model Matching by Improved Activity Label Matching,» *Lecture Notes in Computer Science,* vol. 8094, p. 211–218, Springer Berlin Heidelberg, 2013.

[28] Miller, George, «WordNet: A Lexical Database for English.,» *Communications of the ACM,* p. 38(11):39–41, 1995.

[29] Cayoglu, Ugur; Dijkman, Remco; Dumas, Marlon; Fettke, Peter; Garcıa-Banuelos, Luciano; Hake, Philip; Klinkmüller, Christopher; Leopold, Henrik; Ludwig, André; Loos, Peter et al.:, «The process model matching contest 2013,» 4th International Workshop on Process Model Collections: Management and Reuse (PMC-MR'13), 2013.

[30] Klinkmüller, Christopher; Leopold, Henrik; Weber, Ingo; Mendling, Jan; Ludwig, André, «Listen to Me: Improving Process Model Matching through User Feedback.,» Business Process Management, 2014.

[31] Rahm, Erhard, «Towards Large-Scale Schema and Ontology Matching,» *Schema Matching and Mapping;springer,* pp. 3-27, 2011.

[32] Cayoglu, Ugur; Oberweis, Andreas; Schoknecht, Andreas; Ullrich, Meike, «Triple-S: A Matching Approach for Petri Nets on Syntactic, Semantic and Structural level.,» Technical report,, 2013.

[33] Jens Kolb & al, «The Process Model Matching Contest,» 2015.

[34] Mostefai Abdelkader & Ignacio Garcia Rodriguez, «Process Model Matching using Heuristic Search».

List des figures

Liste des tableaux

Résumé

Notre technique traite le problème d'alignement de processus métiers ; c'est-à-dire la détection de correspondance entre les activités, cette dernière est une tâche très importante dans la gestion des modèles de processus métiers, telles que le fusionnement, regroupement ou l'interrogation des modèles de processus. L'approche traite le problème comme étant un problème de mesure de similarité textuelle entre les activités de processus métiers. La mesure de similarité proposée est une mesure sémantique basée sur Wordnet. L'expérimentation faite sur un benchmark composé de processus métiers de trois domaines montre que notre approche a le potentielle d'aligner effectivement les processus métiers (PM).

تقنيتنا تعالج مشكلة توافق العمليات التجارية؛ أي، الكشف عن التطابقات بين الأنشطة، يعد هذا الأخير مهما جدًا في إدارة نماذج عمليات الأعمال، مثل الدمج أو التجميع أو الاستعلام عن نماذج العمليات. تعامل هذه المقاربة المشكلة كمشكلة لقياس التشابه النصي بين أنشطة العمليات التجارية. مقياس التشابه المقترح هو تدبير دلالي يستند على Wordnet. توضح التجربة التي قمنا بها على مجموعة من البيانات تتكون من العمليات التجارية في ثلاثة مجالات أن مقاربتنا لديها القدرة على الموافقة بين العمليات التجارية بشكل فعال

Our technique addresses the problem of business process alignment; that is, the detection of matches between activities, the latter is a very important task in managing business process models, such as merging, grouping, or querying process models. The approach treats the problem as a problem of measuring textual similarity between business process activities. The proposed similarity measure is a semantic measure based on Wordnet. Experimentation on a benchmark made up of business processes in three domains shows that our approach has the potential to effectively align business processes (PM).

.