REP UBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

**Département du Génie Électrique et de l'Informatique**

**Mémoire de fin d'étude**

**Présenté pour l'obtention du diplôme**

**de Master en Informatique**

***Option* : Ingénierie des Systèmes et Technologie du Web**

**THEME**

# découverte des communautés basée service

**Proposé et dirigé par :**

*Mr. Boudjebbour Karim*

**Préparé et présenté par :**

*Mlle. Saidani Nacera*

*Mlle. Terras Ikram*

**Année universitaire 2014/2015**

# Dédicaces

*Je dédie Ce travail*

*A Mes très chers parents, ma mère qui a beaucoup sacrifié pour moi et qui n'a jamais cessé de m'encourager, mon père qui a fait beaucoup Pour nous.*

*A mes très chers grands parents*

.

*Et en particulier à mon très Cher petit frère Youcef et mes cherès Sœurs (Ghania et Dalila et Hanane et Nour-el-houda et Abla)*

*Aussi à mes tantes, mes oncles, mes cousines et cousins que j'aime beaucoup,*
*A mon binôme Ikram ainsi que sa famille*
*ET toute ma famille qui m'a soutenue dans les moments les plus difficiles que Dieu les protèges tous.*

*Sans oublier tous mes collègues.*

*Merci*



*Nacera*

# Dédicaces

*Je dédie Ce travail*

*A Mes très chers parents, ma mère qui a beaucoup sacrifié pour moi et qui n'a jamais cessé de m'encourager, mon père qui a fait beaucoup Pour nous.*

*A ma très chère grande mère*

.

*Et en particulier à ma très Chère petite sœur Anfoula et mes chers Frères (Abd elfateh et Abd elghani)*

*à la mémoire de l'inoubliable mon très cher frère « allah yarhamou »*
*Aussi à mes tantes et mes oncles et mes cousines que j'aime beaucoup,*
*A mon binôme Nacera ainsi que sa famille*
*ET toute ma famille qui m'a soutenue dans les moments les plus difficiles que Dieu les protèges tous.*

*Sans oublier tous mes collègues*

*Merci*



*Ikram*

# REMERCIEMENT

MERCI !

*Au terme de ce travail, nous tenons à remercier en premier lieu « Dieu » qui nous a donné la force pour achever ce travail.*

*Nous remercions vivement notre promoteur*
*Mr .Boudjebbour .K*
*Et tous les enseignants de l'université de Médéa*

*Nous remercions du fond du cœur nos chers parents de tout ce qu'ils ont fait pour nous.*

*Nous tenons également à remercier tous ceux qui nous ont aidés à réaliser ce travail de près ou de loin.*

*Nos remerciements s'adressent également à tous les membres de jury, qui font l'honneur de juger notre travail*



*Ikram et Nacera*

## Résumé

La détection des communautés dans les réseaux sociaux occupe une grande importance dans divers domaines. En effet, l'identification de la structure des communautés dans les réseaux complexes aide à extraire des informations utiles à analyser et à comprendre. Ces informations concernent les éléments du réseau social et les interactions entre eux.

Plusieurs méthodes de détection de la communauté ont été proposées pour identifier les structures communautaires efficaces. Cependant, la plupart de ces méthodes ne convenaient pas pour les réseaux complexes car ils ont des limitations importantes.

Dans ce travail, nous proposons une méthode de détection de communautés fondées sur les profils des utilisateurs et plus particulièrement sur leurs interactions avec les services Web. Pour ce faire, nous proposons une nouvelle approche de trois processus basé sur une mesure de similarité pour créer un graphe des utilisateurs et de services web, ainsi qu'une nouvelle méthode de détection de communauté en utilisant le graphe biparti résultant.


**Mots-clés:** réseau social, diffusion de l'information, de détection de communautés, réseau complexe, les services Web, segmentation.


## Abstract

The community detection has a great importance in various fields. Indeed, the identification of the community structure of complex networks helps in extracting useful information to analyze and understand it. This information relates to the elements of the social network and the interactions between them.

Several methods of community detection have been proposed to identify effective community structures. However, most of these methods were not suitable for complex networks as they have significant limitations.

In this paper, we propose a method of detecting communities based on user profiles and more specifically on their interactions with web services. To do this we propose a new approach based on two processes based on a similarity measure to create the graph users/web services and a new community detection method using the resulting bipartite graph.


**Keywords:** Social network, information diffusion, communities detection, Complex network, web services, clustering

## ملخص

اكتشاف الجماعات تشغل حيز واسع في مختلف المجالات. إن تحديد هيكلة الجماعات في الشبكات المعقدة تساعد في استخراج معلومات ذات جدوى لتحليلها وفهمها. هذه المعلومات تتعلق بعناصر الشبكة الاجتماعية والتفاعل فيما بينها.

هناك مجموعة من الطرق تم اقتراحها لكشف الجماعة وتحديد الهياكل المتصلة الفعالة. لكن أغلبية هذه الطرق لا تتوافق مع الشبكات المعقدة لأنها تعاني من نقائص معتبرة.

في هذا العمل نقترح طريقة لتحديد مجموعة الأشخاص التي لهم نفس الاهتمامات، تعتمد على بيانات المستخدم وبحالة خاصة على تفاعله مع خدمات الواب. من أجل تحقيق هذا العمل، نقوم باقتراح منهجية جديدة من ثلاث عمليات تعتمد على قياس التشابه من أجل استخراج رسم بياني للمستعملين و خدمات الواب ، وبالتالي طريقة جديدة لكشف الجماعة باستعمال الرسم البياني الثنائي الناتج.


**الكلمات المفتاحية:** الشبكة الاجتماعية، نشر المعلومة، اكتشاف الجماعة، شبكة معقدة، خدمات الواب، تقسيم

.

.

## Table des matières

Résumé ..... i

Table des matières ..... iii

Liste des figures ..... vii

Liste des acronymes ..... viii

Introduction générale ..... 1

**Chapitre I: Les services web**

I.1 Introduction ..... 3

I.2 L'Architecture orientée services (AOS) ..... 3

I.2.1 Les composants de base d'une AOS ..... 3

I.2.1.1 Le fournisseur de service ..... 3

I.2.1.2 L'annuaire de publication ..... 3

I.2.1.3 Le consommateur de service ..... 4

I.3 Service web ..... 4

I.3.1Définition ..... 4

I.3.2 Langage de description XML ..... 5

I.3.3 Protocoles de bases de services web ..... 5

I.3.3.1 WSDL ..... 5

I.3.3.2 SOAP ..... 6

I.3.3.3 UDDI ..... 7

I.3.4 Fonctionnement de services web ..... 8

I.4 Conclusion ..... 10

**Chapitre II : Les réseaux sociaux**

II.1 Introduction ..... 11

II.2 Définition d'un réseau social ..... 11

II.2.1 Les Réseaux sociaux sur Internet ..... 12

II.2.2 Avantages et inconvénients des réseaux sociaux ..... 12

II.3 Les aspects de réseaux sociaux ..... 13

II.4 Analyse des réseaux sociaux ..... 14

II.4.1 Calculs en analyse des réseaux sociaux : ..... 15

II.5 Définition de la notion de communauté ..... 21

II.6 Propriétés topologiques des communautés ..... 22

II.7 Algorithmes de détection de communautés ..... 25

II.7.1 Approches hiérarchiques ..... 26

II.7.2 Algorithmes basés sur les marches aléatoires ..... 27

II.7.3 Algorithmes utilisant les propriétés spectrales des réseaux ..... 29

II.8 Conclusion ..... 30

**Chapitre III : Conception**

III.1 Introduction ..... 31

III.2 L'idée proposée: ..... 31

III.2.1 Processus de découverte des services web ..... 33

III.2.1.1 Profils utilisateur et service ..... 33

III.2.2 Processus d'accès aux services web ..... 39

III.2.3 Processus de découverte des communautés ..... 40

III.2.3.2 Notre approche de détection des communautés basé service ..... 40

III.3 Diagramme de classe de l'application ..... 42

III.4 Algorithmes proposés ..... 43

III.4.1 Algorithme de calcule de similarité ..... 43

III.4.1.1 Principe de fonctionnement ..... 43

III.4.1.2 Présentation de l'algorithme ..... 44

III.4.2 Algorithme d'accès au service ..... 45

III.4.2.1 Principe de fonctionnement ..... 45

III.4.2.2 Présentation de l'algorithme ..... 45

III.4.3 Algorithme de détection de communautés ..... 46

III.4.3.1 Principe de fonctionnement ..... 46

III.3.2 Présentation de l'algorithme ..... 47

III.5 Conclusion ..... 48

**Chapitre IV : Implémentation et mise en œuvre**

IV.1 Introduction ..... 49

IV.2 Les outils et les langages utilisés en développement ..... 49

IV.2.1 visuel studio.Net ..... 49

IV.2.2 le langage C# ..... 50

IV.2.3 SQL Server 2008 ..... 50

IV.2.4 L'éditeur de graphes « pajek » ..... 50

IV.3 Présentation de notre application ..... 51

IV.4 Expérimentation ..... 53

IV.5 Conclusion ..... 57

Conclusion générale et perspectives ..... 58

Bibliographies ..... 59

## Liste des figures

Figure I.1:Entête du protocole SOAP ..... 7

Figure I.2:Le contenu de l'annuaire UDDI ..... 8

Figure I.3:Principe général du fonctionnement des Services Web ..... 9

Figure II.1:Exemple d'un réseau social centré sur un acteur ..... 18

Figure II.2 : Exemple de couplage bibliographique entre i et j ..... 20

Figure II.3:Réseaux artificiels générés avec le modèle de Lancichinetti et al. [16] ..... 22

Figure III.1 : Architecture de notre approche ..... 32

Figure III.2:Représentation graphique du contexte ..... 35

Figure III.3 : Diagramme de classe de l'application ..... 42

Figure IV.1: L'interface principale de l'environnement de développement ..... 49

Figure IV.2: Architecture de fonctionnement SQL Server 2008 ..... 50

Figure IV.3:L'interface principale du pajek ..... 51

Figure IV.4: L'interface de l'application ..... 52

Figure IV.5: La structure génrale du fichier généré ..... 53

Figure IV.6: Graphe de similarité (services/utilisateurs) ..... 54

Figure IV.7: Graphe d'accès au service web ..... 54

Figure IV.8 : Découverte des communautés relatives à un service ..... 55

Figure IV.9 : Découverte des communautés relatives à deux services ..... 55

Figure IV.10: Le nuage de communautés avec 200 utilisateurs et 15 service……..56

Figure IV.11: Graphe de communautés finale…….58

## Liste des acronymes

**AOS :** Architecture Orientée Service.

**XML:** eXtensible Markup Language.

**WSDL:** Web Services Description Language.

**SOAP:** Service Oriented Access Protocol.

**UDDI:** Universal Description Discovery and Integration.

**W3C:** World Wide Web Consortium.

**IDE:** Integrated Development Environement.

**AJAX:** Asynchronous Java Script And XML.

**SGML:** Standard Generalized Markup Language.

**SQL:** Structured Query Langague.

# Introduction générale

La détection de communautés occupe une grande importance dans divers domaines. En effet, l'identification de la structure communautaire des réseaux complexe contribue dans l'extraction des informations utiles permettant de les analyser et de les comprendre.

Les méthodes de détection de communautés dans les réseaux sociaux ont fait l'objet de nombreux travaux. La plupart d'entre eux modélisent les réseaux sociaux par graphes, ou les sommets représentent les individus, et les arêtes, les relations entre eux. Ces méthodes consistent à détecter la présence de groupes de sommets plus fortement connectés entre eux qu'avec les autres sommets du graphe. Ainsi, elles se basent sur les éléments du réseau et les interactions entre eux, se qui soulève des problèmes de composition et de compréhension des structures de communautés.

En premier lieu, vu la diversité des interactions entre les utilisateurs, les communautés détectées peuvent graviter autour d'une infinité de critères : elles peuvent représenter des groupes d'individus entretenant des relations sociales, partagent des centres d'intérêts communs, des activités communes etc. Le problème réside dans le fait que nous ne connaissons pas l'objet de l'interaction à l'origine de la construction des communautés.

D'autre part, ces méthodes de détection positionnent l'utilisateur comme l'élément central des interactions, ce qui les rend inappropriées aux réseaux complexe. En effet, nous assistons ces dernières années à l'émergence des réseaux sociaux en ligne tels que Facebook, Twiter etc. Ces réseaux comptent à ce jour des millions d'utilisateurs et amassent de très nombreuses données les concernant. Dès lors, se pose pour ces méthodes de détection, le problème de l'exploitation des données et de l'exploitation des utilisateurs.

Dans le cadre de ce travail nous allons nous intéresser au problème de la détection de communautés. Pour pallier aux manques décelés dans les méthodes actuelles, nous proposons une nouvelle approche de détection de communautés. Contrairement aux méthodes existantes, elle ne se base pas sur les interactions entre les utilisateurs, mais sur l'interaction des utilisateurs avec les services similaires à leurs besoins et leurs centres d'intérêts. Notre méthode utilisera une mesure de similarité qui se base sur les interactions des utilisateurs à travers les services web. On parlera ainsi de découverte des communautés basée service. Chaque communauté détectée sera rela-

tive à un service ou à un groupe des services web donné et sera formée des utilisateurs utilisant ce (ou ces) service (s).

Notre choix s'est porté sur les services web comme éléments moteurs des découverte et non pas les utilisateurs pour les raisons suivants : d'une parte de regrouper les utilisateurs utilisant les mêmes services web peut s'avérer extrêmement utile dans divers domaines comme la publicité ciblé. D'autre part contrairement à l'exploitation des utilisateurs qui est fastidieuse dans les réseaux complexes, celle des services web est simplifiée par l'existence d'un annuaire standard pour le référencement de l'ensemble des services web.

Notre mémoire est structuré comme suit : le premier chapitre définit les services web et décrit les technologies clés sur lesquels ils reposent. Le deuxième chapitre est consacré aux réseaux sociaux. Il décrit les concepts les plus importants de leur analyse, le troisième chapitre présente la méthode de détection de communautés que nous avons proposé et le dernier chapitre présente l'implémentation de notre méthode de détection de communautés basée service par des outils adéquats.

## I.1 Introduction

Dans ce chapitre nous allons introduire une des concepts théoriques concernant notre projet, nous allons commencer par définir brièvement l'architecture orientée service. Nous présentant plus en détail les services web en fournissant les protocoles de bases le fonctionnement de services web.

## I.2 L'Architecture orientée services (AOS)

L'AOS est une approche architecturale permettant la création des systèmes basés sur une collection de services développés dans différents langages de programmation, hébergés sur différentes plates-formes avec divers modèles de sécurité et processus métier [1].

Chaque service représente une unité autonome de traitement et de gestion de données, communiquant avec son environnement à l'aide de messages. Les échanges de messages sont organisés sous forme de contrats d'échange. L'idée maîtresse de l'architecture orientée service est que tout élément du système d'information doit devenir un service identifiable, documenté, fiable, indépendant des autres services, accessible, et réalisant un ensemble de tâches parfaitement définies

### I.2.1 Les composants de base d'une AOS

L'AOS s'articule autour de trois concepts fondamentaux à savoir le fournisseur de services, l'annuaire de publication et le consommateur de services [1].

#### I.2.1.1 Le fournisseur de service

Désigne une personne ou une organisation qui publie le service sur un serveur et génère sa description contenant les informations nécessaires afin de l'utiliser, ainsi que les opérations disponibles et leur mode d'invocation.

#### I.2.1.2 L'annuaire de publication

Est un répertoire de description de services. Il représente aussi un médiateur entre les consommateurs et les fournisseurs de services. En fait, l'annuaire automatise les communications

entre ces deux derniers en fournissant aux consommateurs les informations techniques et sémantiques sur le fonctionnement du service.

#### I.2.1.3 Le consommateur de service

Il représente une application ou un serveur qui lance une requête à l'annuaire pour récupérer la description d'un service qui répond à ses besoins.

Le consommateur peut alors utiliser cette description pour établir une connexion avec le fournisseur et invoque les opérations du service choisi.

L'AOS encapsule plusieurs avantages bénéfiques pour le domaine de la technologie d'information et de communication. Elle offre la simplicité à travers les concepts de décomposition, de découplage et de réutilisation. En plus, elle participe à la réduction du coût de développement des grands projets et rend le développement plus efficace.

## I.3 Service web

### I.3.1Définition

Suite à l'étude des différentes définitions sur la composition de services web, nous retenons la définition de W3C (World Wide Web Consortium) qui nous paraît la plus consistante. En effet, selon W3C, « *A web service is a software application or component that can accessed over the internet using a platform/language-neutral data interchange format to invoke the service and supply the response, using a rigorously defined message exchange pattern, and producing a result that is sufficiently well defined to be processed by a software application.* »

Un service web est une technologie modulaire permettant à des applications de dialoguer à distance via internet avec d'autres applications ou des systèmes hétérogènes dans des environnements distribuent. Ainsi, il s'appuie sur un ensemble de protocoles internet comme HTTP an de communiquer à travers l'échange de messages. En outre, les services web se basent sur trois standards fondamentaux tels que WSDL (Web Service Description Languag), SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration).

**Les principales caractéristiques d'un service web sont :**

- L'interopérabilité grâce à son indépendance par rapport aux systèmes d'exploitation et des langages de programmations
- Publiable et accessible en utilisant le langage XML, les protocoles ouverts et les standards du web.
- Vise à exposer une ou plusieurs fonctionnalités.
- Ne pose pas de problèmes lors de son fonctionnement avec les pare-feux.

### I.3.2 Langage de description XML

Est un format de texte simple, très flexible tiré du SGML « Standard Markup Language » (l'ISO 8879). À l'origine, conçu pour la publication électronique à grande échelle, XML. Joue aussi un rôle de plus en plus important dans l'échange d'une large variété de données sur le web et ailleurs.

W3C recommande depuis 1998 XML en tant que standard de description de données.XML est un méta langage permettant d'identifier la structure d'un document.

Un document XML est composé d'une structure et d'un contenu. La structure d'un document XML est souvent représentée graphiquement comme un arbre. La racine du document constitue le sujet du document, et les feuilles sont les éléments de ce sujet. De ce fait, XML est alors flexible et extensible, et est devenu rapidement le standard d'échange de données sur le web [3] ».

### I.3.3 Protocoles de bases de services web

La technologie de services web se base sur un ensemble de protocoles standards pour l'échange de données entre les applications :

#### I.3.3.1 WSDL

WSDL (Web Services Description Language) [4] est un langage à base de XML qui permet de décrire et de publier les formats et les protocoles d'un service web d'une manière standard. En effet, ce langage désigne une description de type « boite noire »en termes de format de messages requis pour communiquer avec ce service, les protocoles de communication et les

opérations exposées par un service web. Deux types de description sont possibles avec WSDL tels que **:**

- **La description abstraite :** elle définit d'une façon abstraite l'interface du service en terme de :

**-<types>** : qui permettent de définir les types de données utilisées dans les messages échangés.

**-<message>** : qui définit les types des données échangées.

**-<opérations>** : qui définissent d'une manière abstraite l'ensemble des actions supportées par le service.

**-PortType>** : ou Interface qui permet la définition abstraite d'un ensemble d'opérations.

- **La description concrète :** elle désigne la définition de l'implémentation d'un service donné en terme de :

**-<binding>** : qui permet la description des protocoles (SOAP, HTTP,..) utilisés pour l'échange de données et de formats de messages pour chaque <porttype>.

**-<Port> :** Définition d'un point d'entrée au travers duquel il est possible d'accéder à un ensemble d'opérations.

**-<service>** : permet la définition d'un service web comme un ensemble de <port>.

### I.3.3.2 SOAP

SOAP (Service Oriented Access Protocol) est un protocole de transmission de messages entre les services distants dans un environnement distribué. SOAP a été initialement défini par une initiative conjointe entre Microsoft et IBM en fusionnant XML et HTTP (HyperText Transfer Protocol) dans un protocole commun pour transporter les messages d'une façon structurée et assurer plus d'interopérabilité. [5]

**Structure d'un message soap**

Le message SOAP est un document XML présenté sous forme normalisée. Il est composé de deux parties :

- Une enveloppe qui contient les informations sur le message et le processus d'acheminement et de traitement.
- Un modèle de données qui désigne le for mat de message à échanger.

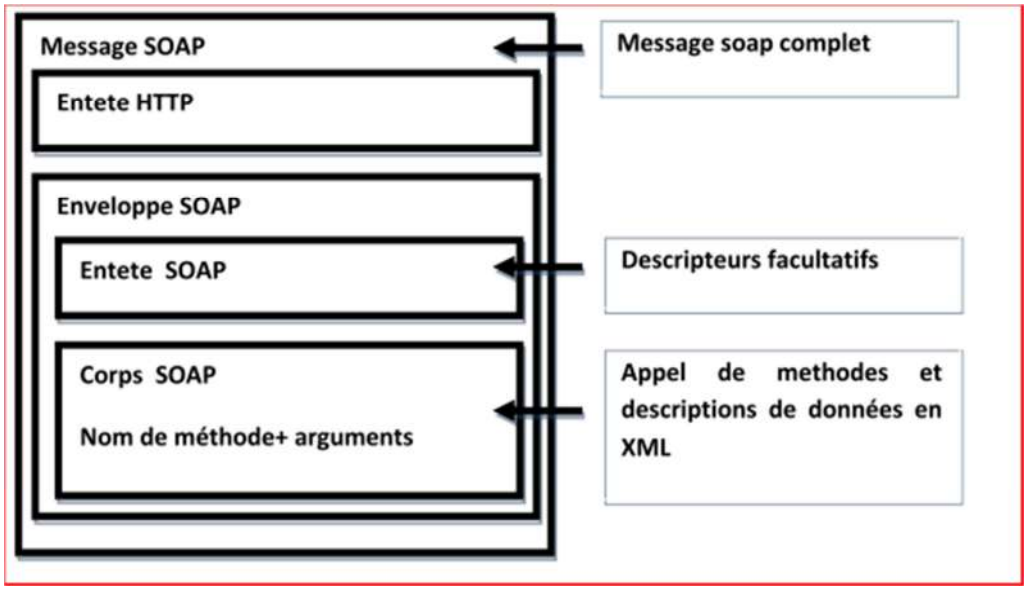



**Figure I.1:Entête du protocole SOAP**

L'entête est facultatif, il peut contenir des informations de sécurité (telles que les signatures électroniques), des informations transactionnelles, des informations de traçabilités…

La partie body est obligatoire elle contient les éléments suivants:

• soit le nom de méthode, avec les données correspondantes, ou un simple document XML, pour le cas d'une requête.

• les valeurs de retour ou un message d'erreur pour le cas d'une réponse.

### I.3.3.3 UDDI

UDDI (Universal Description, Discovery and Integration) [2] est une spécification d'annuaires de services Web : cette norme W3C propose un ensemble de structures à pu-

blier par les fournisseurs de services. Ces structures sont formalisées en XML, elles proposent 03 types d'informations (voir la figure) :

**Les pages blanches** qui décrivent les informations de contacts sur les entreprises

**Les pages jaunes** qui décrivent des informations de classification de services

**Les pages verte** qui donnent des informations techniques (telles que l'accès) des services

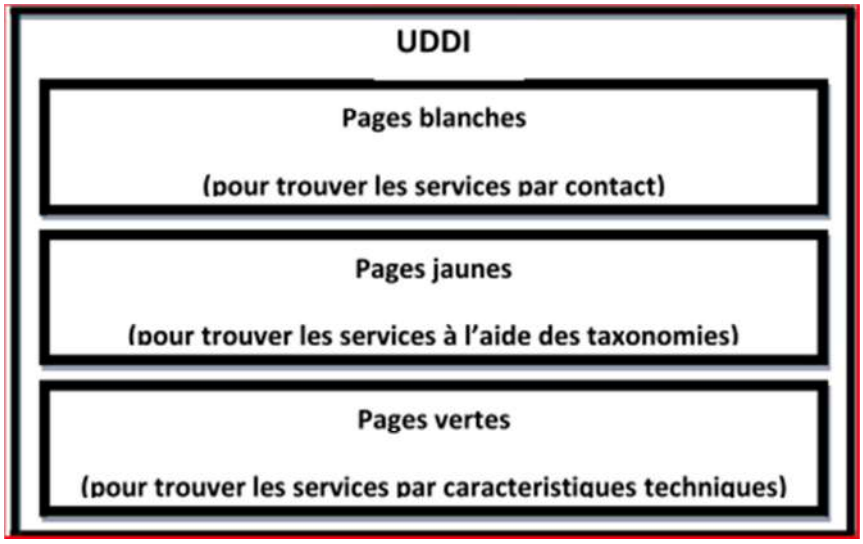



**Figure I.2:Le contenu de l'annuaire UDDI**

UDDI se comporte aussi comme un service web qui dispose d'un ensemble d'opérations comme la recherche, l'ajout et la suppression de services. En outre, il permet aux développeurs de découvrir les détails techniques d'un service web ainsi que les informations métiers à travers le fichier WSDL.

### I.3.4 Fonctionnement de services web

Le scénario classique du processus d'invocation d'un service web est décrit comme suit [6]

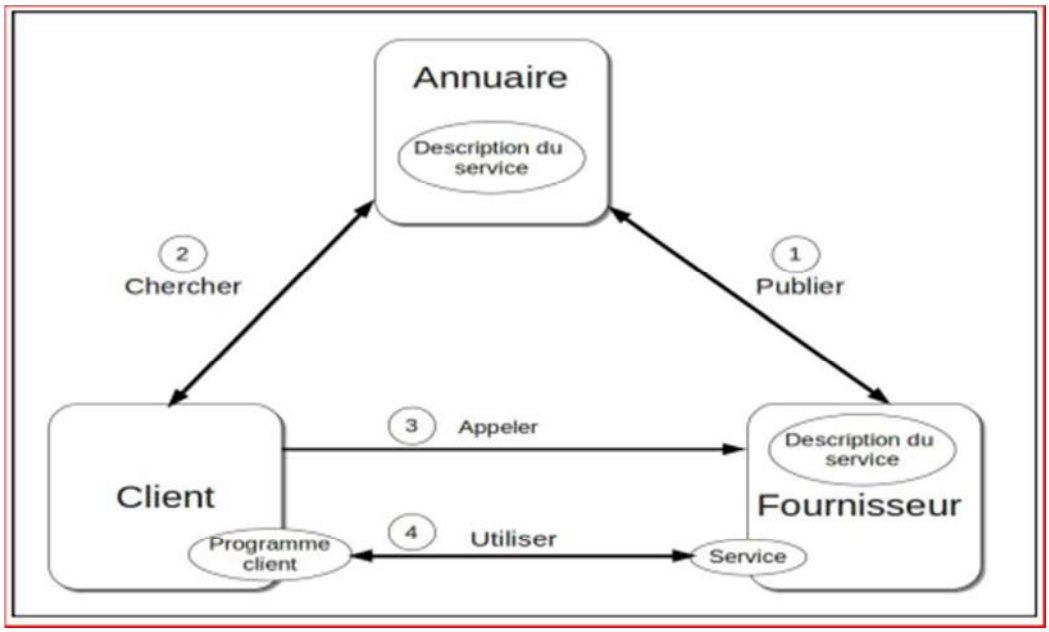



**Figure I.3:Principe général du fonctionnement des Services Web**

1. Le fournisseur de service web déploie son service et le rend accessible à travers internet. Ensuite, il le publie dans un annuaire en décrivant ses caractéristiques dans un fichier WSDL.

2. Ensuite, le client du service web envoie une requête SOAP à l'annuaire de service en spécifiant ses propres exigences de recherches. L'annuaire cherche les services demandés selon les critères spécifiés et retourne au client tous les URL des services disponibles.

3. Une fois le client sélectionne le service voulu, il établit un contrat avec le fournisseur pour invoquer le service web.

4. Le client interagit à partir de son application avec le service web à travers des requêtes SOAP en spécifiant les méthodes à appeler, et d'échanges (envoi et réception de messages). En outre, il s'agit d'une agrégation de services web élémentaires pour aboutir à un nouveau service web à valeur ajoutée.

## I.4 Conclusion

Un Service Web est un composant logiciel représentant une fonction applicative (ou un service applicatif). Il est accessible depuis une autre application (un client, un serveur ou un autre service Web) à travers les réseaux Internet en utilisant les protocoles de transports disponibles.

## II.1 Introduction

Les réseaux sociaux est un nouveau concept qui a toujours existé. En effet, depuis l'existence de l'être humain, que ce soit dans sa vie personnelle ou professionnelle, il se regroupe par des centres d'intérêts pour former des réseaux. Aujourd'hui, l'internet amplifie ce phénomène et les individus ont désormais la possibilité de se regrouper en ligne.

L'émergence des réseaux sociaux en ligne est liée aux révolutions technologiques et techniques. En effet, depuis quelques années maintenant, nous assistons à une métamorphose de l'internet. Ce changement a un nom le Web 2.0.

Le Web 2.0 est avant tout un changement de comportement social sur l'internet à l'inverse du Web 1.0 ou l'utilisateur était seulement passif, les nouvelles technologies déployées dans le Web 2.0 permettent à l'utilisateur d'être au centre de ce nouveau concept. Parmi les révolutions majeures du Web 2.0, nous pouvons citer l'apparition de la technologie AJAX qui a permis des interactions plus rapides avec les pages Internet. De ce fait, le nombre de membres des réseaux sociaux s'est allongé, d'une part car les interactions étant plus rapides, consulter internet est devenu plus confortable, et d'autre part car les utilisateurs prennent conscience de leur pouvoir d'interagir sur la toile, ils peuvent désormais créer leur propre Internet : ils partagent diffusent et échangent des informations.

## II.2 Définition d'un réseau social

Un réseau social est un ensemble d'identités sociales, telles que des individus ou encore des organisations, reliées entre elles par des liens, créés des interactions sociales. Le réseau social est représenté par une structure ou une forme dynamique d'un groupement social. Des réseaux sociaux peuvent être créés stratégiquement pour agrandir ou rendre plus efficient son propre réseau social (professionnel, amical).

Pour résumer, un réseau social est un moyen de conforter, retrouver, créer des relations avec des personnes et de pouvoir interagir avec eux à l'aide des messageries virtuelles

### II.2.1 Les Réseaux sociaux sur Internet

Les réseaux sociaux sur Internet se définissent comme un espace de partage de l'information entre « amis ». Leur taille, leur mode de fonctionnement quasi instantané via les publications sur les « murs » fait que l'information y circule extrêmement rapidement et peut générer des mobilisations très larges en un temps très court. Ces réseaux fonctionnement sur l'affect : en dehors des échanges traditionnels entre amis, les infos qui circulent le plus rapidement sont souvent les plus amusantes, étonnantes, choquantes. Ces sites s'avèrent d'efficaces caisses de résonnance pour des sujets sensibles. Bien utilisés, ces réseaux sociaux peuvent jouer un formidable rôle en complément d'un outil marketing.

Certains « réseaux sociaux » sur Internet regroupent des amis de la vie réelle. D'autres aident à se créer un cercle d'amis, à trouver des partenaires commerciaux, un emploi ou autres. Il s'agit de services de réseautage social, comme Facebook, Twitter, MySpace, Mupiz, Viadeo, Instagram, LinkedIn ou Pheed. [7]

### II.2.2 Avantages et inconvénients des réseaux sociaux

Un tableau simplifié concernant les avantages et inconvénients des réseaux sociaux, nous allons, par la suite, commenter ce tableau tout au long de cette partie. [7]

| **Avantages** | **Inconvénients** |
|---|---|
| Outils à faibles couts | Outils à confidentialité limitée |
| Fort pouvoir de communication | Actions de communication irrémédiables |
| Connexion permanente | Stress potentiel permanent |
| Permettent une fusion des technologies | Empêchent l'indépendance des outils |
| Forte interaction | Forte subjectivité |
| Relations humaines supplémentaires | Complexification des relations humaines |
| Innovations permanentes | Obligation de suivre les technologies |
| Automatisation de certains processus | Industrialisation des relations humaines |

## II.3 Les aspects de réseaux sociaux

Deux aspectes se côtoient quand on parle de réseaux sociaux : d'un cote, l'aspect sociologique et de l'autre l'aspect technologie [8].

D'un point de vue sociologique, un réseau social se définit par différents éléments que sont les individus, leurs liens, leurs affinités et l'environnement les entourant. Pour résumer, un réseau social est un ensemble de relations qu'entretiennent des unités sociales les uns avec les autres, directement, ou indirectement, à travers des chaînes de longueurs variables. Ces unités sociales peuvent être des individus, des groupes informels d'individus ou bien des organisations plus formelles, comme des associations, des entreprises, voire des pays. Les relations entre les éléments désignent des formes d'interactions sociales qui peuvent être elles aussi de natures extrêmement diverses : il peut s'agir de transactions monétaires, de transferts de biens ou d'échanges de services, de transmissions d'informations, de perceptions ou d'évaluation interindividuelles, d'ordres, de contacts physique et plus généralement de toutes sortes d'interaction verbales ou gestuelles, ou encore de la participation commune à un même événement, etc.

D'un point de vue technologie, un réseau social désigne l'ensemble des sites Internet permettant de faciliter l'établissement d'interrelation (relations sociales) entre d'individus à travers l'Internet. Il fournit à ses membres des outils et interfaces d'interaction, de présentation de communication. Les relations entre personnes se construisent autour d'intérêts partagés, d'activités communes, de connaissances dans la vie réelle, etc. Ils permettent à chaque utilisateur d'exposer ces idées, ses activités, ses intérêts, et de partager avec un nombre restreint de ses relations.

Les réseaux sociaux en ligne peuvent être classés selon deux catégories : les réseaux implicites et les réseaux sociaux explicites. Les réseaux sociaux implicites ne sont pas construits autour des utilisateurs mais des contenus. Par exemple, les réseaux de partage de vidéos. Les réseaux sociaux explicites eux, sont orientés utilisateurs. Par exemple, le contenu des réseaux professionnels n'est basé que sur les fiches des utilisateurs.

Les réseaux sociaux peuvent s'adapter à diverses thématiques : recherche d'emploi, rencontres entre individus échanges d'informations autour d'un centre d'intérêt commun, partage de contenus multimédias, etc.

## II.4 Analyse des réseaux sociaux

L'analyse des réseaux sociaux est définie comme étant l'étude des entités sociales (les personnes dans les organisations qu'on appelle acteurs) ainsi que leurs interactions et leurs relations. Ces interactions et relations peuvent être représentées par un graphe ou un réseau, dans lequel chaque nœud représente un acteur et chaque lien est une relation. Nous pouvons étudier les propriétés de la structure et son rôle ainsi que la position et le prestige de chaque acteur social [9]. Nous pouvons rechercher aussi les différents types de sous-graphes comme par exemple les communautés formées par des groupes d'acteurs ayant des intérêts communs, en isolant le groupe d'individus ayant une densité élevée. Les réseaux social peut être aussi une source permettant l'élaboration de recommandations : trouver un expert dans un domaine donné, suggérer des produits à vendre, proposer un ami, etc. Cette élaboration peut être fondée sur des algorithmes d'exploration de chemins, d'analyse de degrés, etc[10][11].

Les réseaux sociaux peuvent être modélisés par des graphes ayant des propriétés spécifiques, nous citons :

- **Le diamètre du réseau** : (le plus long chemin entre deux nœuds) est de longueur bornée. Cette propriété est appelée la propriété du petit monde et issue des études sociologiques annonçant qu'il existe 6 degrés de séparation selon l'expérience du petit monde de Stanley Milgram1967[1]
- **La transitivité dans le réseau :** qui est basée sur le postulat : « *l'ami de mon ami est mon ami* ». Cette propriété se traduit par le nombre de triangles dans le réseau d'où la mesure du taux de clustering suivant :

$$C = \frac{3 * \text{nombre de triangles dans le réseau}}{\text{nombre de triples connectés de noeuds}}$$

- **La distribution des degrés :** (nombre de nœuds connectés à un nœud donné) obéit à la distribution de degrés en loi de puissance. Nous avons évoqué la modélisation d'un réseau social par un graphe, or la description du réseau peut nécessiter d'éléments décrivant le contenu sémantique des acteurs et/ou des liens. Il est possible donc d'envisager une modélisation sémantique qui consiste à :

[1] 2 personnes choisies aléatoirement sont reliées au plus par 6 personnes intermédiaires
Rapport technique - LARIS-EISTI, octobre 2009.

- stocker au niveau des acteurs (nœuds) les informations concernant le profil utilisateur.
- stocker au niveau de lien une plusieurs informations : types de communication (email, etc), nombre ou fréquence d'interaction, la durée, nombre de participants, etc.

Les réseaux sociaux sont utilisés principalement pour décrire les interactions entre les entités sociales. Les réseaux sociaux sont encore utilisés pour modéliser d'autres types d'interaction comme les liens entre un ensemble de pages Web ou bien les citations bibliographiques dans un corpus de documents.

L'analyse des réseaux sociaux est utile pour la toile parce que le Web est une société virtuelle et donc un réseau social virtuel, où chaque page est considérée comme un acteur social et chaque hyperlien est une relation.

Un certain nombre de résultats issus des études des réseaux sociaux peuvent être adaptés et étendus dans le contexte des applications Web. Les idées développées dans le domaine de l'analyse des réseaux sociaux ont largement contribué au succès des moteurs de recherche [12]. De même, l'analyse des citations exprimées dans les références bibliographique permet de découvrir des similarités entre ces documents outre les mesures fondées sur le contenu, et de définir des centres d'affinités entre les auteurs, etc.

Par ailleurs, nous citons le terme *réseautage social* qui signifie la communication basée sur le partage et les commentaires des contenus. La problématique qui se pose ici sera l'extraction d'un réseau social dans un espace de travail collaboratif ou en analysant les interactions entre les internautes.

Dans les sections suivantes, nous détaillons les mesures utilisés dans des réseaux sociaux. Les mesures locales sont les indices permettant de donner des informations locales sur un acteur donné comme la centralité ou le prestige. Les mesures globales sont porteuses d'éléments d'information globale sur le réseau.

### II.4.1 Calculs en analyse des réseaux sociaux :

- **Les mesures locales :** Dans cette section, nous introduisons deux types de notions pour l'analyse des réseaux sociaux [13] : la centralité et le prestige qui sont bien liés à l'analyse des hyperliens et la recherche dans le Web. La centralité ainsi que le prestige sont des mesures de degrés de la pertinence d'un acteur dans un réseau social.

- **La centralité :** Les acteurs importants sont ceux qui sont liés et impliqués avec les autres acteurs d'une façon extensive. Dans le cadre d'une organisation, une personne qui a beaucoup de contacts et qui communiquent bien avec les autre personnes est considérée plus importante qu'une autre personne ayant moins de contacts. Ces contacts sont modélisés par des liens. Un acteur central est un acteur qui est impliqué dans plusieurs liens. (figure II.1) montre un exemple simple qui utilise un graphe non dirigé. Chaque nœud dans le réseau social est un acteur et chaque lien indique que les deux acteurs aux extrémités communiquent ensemble. Intuitivement, nous remarquons que l'acteur 1 est l'acteur le plus central parce qu'il/elle communique avec la majorité des autres acteurs. Il y a différents types de liens ou d'implications entre acteurs. Par conséquence, plusieurs types de centralités seront définis sur les graphes dirigés et non dirigés. Nous détaillons ces mesures dans la suite.
- **Degré de Centralité :** Les acteurs centraux sont les acteurs les plus actifs et qui ont le plus de liens avec les autres acteurs. Soit *n* le nombre total d'acteurs dans les réseaux.
- **Graphe non dirigé** : Dans un graphe non dirigé, le degré de centralité d'un acteur *i* noté *CD*(*i*) est le degré du nœud acteur (le nombre d'arêtes) noté *d*(*i*) normalisé par le degré maximal, *n* -1.

$$CD(i) = \frac{d(i)}{n-1}$$

La valeur de cette mesure varie entre 0 et 1 puisque $n_i1$ est la valeur maximale de *d*(*i*)

- **Graphe dirigé** : Dans ce cas, nous distinguons entre les liens entrants d'un acteur *i* (les liens pointant vers i) et les liens sortants (les liens pointant à partir de i). Le degré de centralité est définie à partir des liens sortants (le nombre des liens sortants : *do*(*i*)).

$$C'_D(i) = \frac{do(i)}{n-1}$$

- **Centralité de proximité** : Cette approche définit la centralité en utilisant la notion de proximité ou de distance. *Xi* est un acteur central s'il peut interagir facilement avec les autres acteurs. Par conséquence, sa distance avec les autres doit être courte. Nous utilisons donc la distance la plus courte pour calculer cette mesure. Soit *d* (*i*, *j*) la distance la plus courte à partir de l'acteur *i* vers l'acteur *j* (mesurée par le nombre de liens via le chemin le plus court)

- **Graphe non dirigé** : La centralité *CC*(*i*) de proximité d'un acteur *i* est définie par :

$$CC(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

La valeur de cette mesure varie entre 0 et 1 comme *n ¡* 1 est la valeur minimale du dénominateur, qui correspond à la somme de la distance la plus courte à partir de l'acteur *i* vers les autres acteurs. Noter bien que cette équation est valable seulement dans un graphe connecté.

- **Graphe dirigé** : La même équation est utilisée pour les graphes dirigés. Le calcul de la distance doit intégrer les directions des liens ou les arcs.
- **Centralité d'intermédiarité** : Si deux acteurs non adjacents *j* et *k* veulent communiquer et si l'acteur *i* se localise sur le chemin entre *j* et *k*, alors *i* a un certain contrôle sur leur interaction. L'intermédiarité mesure ce contrôle de *i* sur les deux autres acteurs. Par conséquent, si *i* se localise sur le chemin de plusieurs interactions alors *i* est un acteur important.
- **Graphe non dirigé :** Soit *Pjk* le nombre des chemins les plus courts entre les deux acteurs j et k. L'intermédiarité d'un acteur *i* est définie par le nombre des chemins les plus courts entre j et k passant par i notés par *pjk*(*i*) (avec *j* ≠*i* et *k*≠*i* ),w normalisé par le nombre total des chemins les plus courts entre tous les pairs d'acteurs qui n'incluent pas *i* :

$$C_B(i) = \sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}$$

Noter bien qu'il peut exister plusieurs chemins plus courts entre l'acteur *j* et l'acteur *k*. Quelques-uns passent par *i* et d'autres non. Nous supposons que tous les chemins ont la même chance d'être utilisés. *CB*(*i*) a un minimum de 0, atteint quand *i* ne figure sur aucun des chemins les plus courts. Le maximum atteint est de $\frac{(n-1)(n-2)}{2}$ qui correspond au nombre de pairs d'acteurs qui ne contiennent pas *i*.

Dans le réseau (figure II.1) l'acteur 1 est l'acteur le plus central). Il figure sur tous les 16 plus cours chemins qui lient les 6 autres acteurs.

*CB*(1) atteint la valeur maximale de 15 tandis que

$CB(2) = CB(3) = CB(4) = CB(5) = CB(6) = CB(7) = 0.$

Remarquer bien que la mesure de co-citation est symétrique.

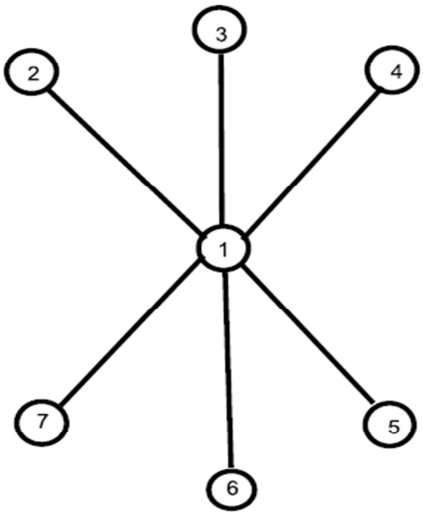



**Figure II.1:Exemple d'un réseau social centré sur un acteur**

Si nous souhaitons que la valeur de cette mesure varie entre 0 et 1, nous pouvons normaliser par $\frac{(n-1)(n-2)}{2}$ qui correspond à la valeur maximale de $CB(i)$. L'intermédiarité standard de l'acteur *i* est définie par :

$$C'_B(i) = \frac{2\sum_{j<k}\frac{p_{jk}(i)}{p_{jk}}}{(n-1)(n-2)}$$

A l'opposé de la mesure de proximité, l'intermédiarité peut être calculée même si le graphe n'est pas fortement connecté.

- **Graphe dirigé** : La même relation peut être utilisée mais doit être multipliée par 2 car nous considérons dans ce cas (*n* - 1)(*n* - 2) pairs puisqu'un chemin de j vers k est différent du chemin inverse allant de k vers j. De même, *pjk* considèrent les chemins dans les deux directions
- **Le prestige :** Le prestige est une mesure plus raffinée que la centralité. Nous distinguons entre les liens sortants et ceux entrants. Un acteur prestigieux est un acteur ayant beaucoup de liens entrants.

Le prestige d'un acteur est mesuré par le nombre des liens entrants. Nous citons Les mesures suivantes :

Le degré de prestige donné par la relation :

$$PD(i) = \frac{dI(i)}{n-1}$$

Avec *PI*(*i*) le degré entrant du nœud *i*, n'étant toujours le nombre d'acteurs (nœuds) dans le réseau

- **le prestige de proximité :** Soit *Ii* l'ensemble d'acteurs qui atteignent l'acteur *i* : soit j un acteur appartenant à *Ii* (*j* ∈ *Ii*), et soit *d* (*j; i*) le chemin le plus court de j vers i ; la distance moyenne est

$\frac{\sum_{j \in Ii} d(j,i)}{|Ii|}$ Où *Ii* étant la norme (le nombre d'éléments) de *Ii*. La proximité est définie comme étant le rapprochement des auteurs acteur de l'acteur i. Le prestige de proximité est défini comme étant :

$$P_p\,(i) = \frac{\frac{|Ii|}{n-1}}{\frac{\sum_{j \in Ii} d(j,i)}{|I|}}$$

Remarquer bien que *Pp*(*i*) ∈[0*;* 1] et que plus les autres acteurs sont prêts de l'acteur i plus cette mesure est élevée.

- **Le prestige de tri (rank prestige) :** Les mesures proposées jusqu'au là sont fondées sur les liens entrants et sortants d'un acteur donné. La mesure du prestige du tri considère la réputation et l'importance des acteurs choisissant l'acteur i ! Ce prestige est calculé selon la formule :

$$P_R(i) = A_{1i}P_R(1) + A_{2i}P_R(2) + \ldots.. + A_{ni}P_R(n)$$

où A étant la matrice d'adjacence ; et *PR*(*i*) étant la mesure de prestige de tri lié à l'acteur *i*. Remarquer bien que le prestige de tri de l'acteur i dépend des prestiges de tri des acteurs j ayant des liens sortants vers l'acteur i (ji).

Pratiquement, pour calculer le prestige de tri on procède ainsi :

– $A_{ij}$ vaut 1 si i pointes vers j, sinon 0

– On pose $P$ le vecteur contenant les valeurs de tri des n acteurs :

$$P = (P_R(1); P_R(2),\ldots\ldots, P_R(n))^T$$

A est la matrice exprimant la propriété adjacente, on a donc :

$$P = A^T P$$

– Recherche de valeurs propres d'$A^T$

Noter bien que cet algorithme est utilisé pour trié les résultats des moteurs de rechercher comme Google (algorithme PageRank).

- **Les mesures globales :** Il existe certaines mesures globales qui permettent d'avoir une idée sur la structure globale du réseau, nous citons :
- **La densité du graphe** : permet de décrire la connectivité à l'intérieur du graphe représentant le réseau :

$$D = \frac{m}{\frac{n*(n-1)}{2}}$$

Où m étant le nombre de liens dans le graphe, n'étant le nombre de nœuds, remarquer bien que le nombre maximal de liens dans un graphe est $\frac{n*(n-1)}{2}$

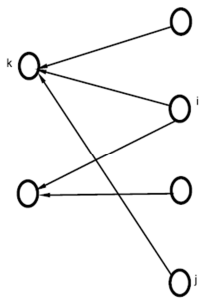


**Figure II.2 : Exemple de couplage bibliographique entre i et j**

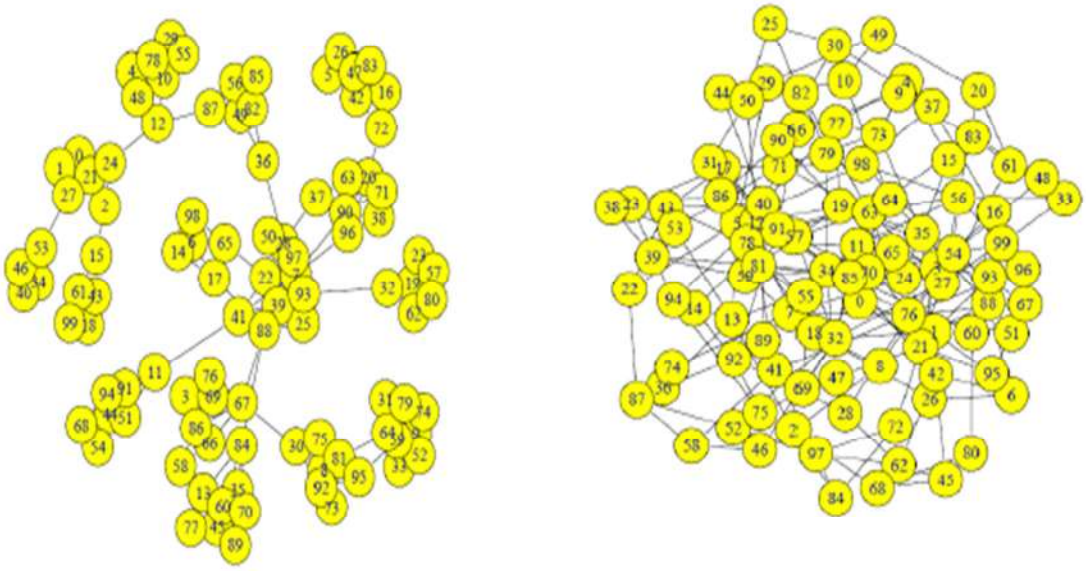
- La distance géodésique entre deux nœuds est le plus court chemin entre les deux nœuds.
- La distance moyenne d'un graphe connecté est égale à la moyenne des distances géodésiques entre tous les pairs d'acteurs.
- Le diamètre d'un graphe connecté est égal à la distance géodésique maximale au sein d'un groupe.

## II.5 Définition de la notion de communauté

L'existence de zones plus densément connectées que d'autres constitue une des caractéristiques non triviale que l'on retrouve dans de nombreux cas de réseaux réels. Ces zones sont appelées communautés par analogie avec les réseaux sociaux. La première tentative pour formaliser cette notion dans le cadre des réseaux apparaît d'ailleurs dans ce domaine [14] .Une communauté est ainsi définie comme un groupe d'éléments qui possède les propriétés suivantes :

- **Cohésion :** Les membres du groupe entretiennent de nombreux contacts entres eux. Dans le cadre des graphes, cela fait référence à la densité et à la notion de clique.
- **Compacité :** Les membres du groupe peuvent être atteints facilement. Ce qui se traduit par une faible distance entre nœuds dans un graphe.
- **Séparation :** Les membres du groupe ont plus de contacts entre eux qu'avec l'extérieur. Les nœuds d'une communauté sont donc plus densément connectés entre eux qu'avec l'extérieur.

Les communautés correspondent intuitivement à des groupes dans lesquels les nœuds sont plus fortement connectés entre eux qu'avec les nœuds des autres groupes [15]. La définition la plus stricte que l'on puisse d'ailleurs adopter est donc qu'une communauté est une clique qui possède peu de liens avec l'extérieur. Une structure communautaire fait référence à une structure topologique du réseau sous forme de communautés. Une telle structure est caractérisée par une distribution inhomogène des liens par opposition à une structure où les liens sont distribués de façon homogène. Ces deux situations sont représentées sur la figure (II.3) Le réseau de gauche possède une structure communautaire. Le réseau de droite est dépourvu de structure communautaire.



Réseau avec une structure communautaire Réseau sans structure communautaire

**Figure II.3:Réseaux artificiels générés avec le modèle de Lancichinetti et al. [16].**

Dans les grands graphes de terrain, les communautés peuvent être considérées comme des entités relativement indépendantes et peuvent avoir des interprétations différentes suivant le domaine d'application. Dans les réseaux sociaux, les groupes de nœuds identifiés sont souvent interprétés en tant qu'unités organisationnelles ; ils correspondent à des ensembles d'individus possédant des points communs et dont les liens sociaux sont naturellement plus forts. Pour les réseaux d'information, elles peuvent correspondre à des thématiques ; par exemple les pages Web traitant d'un même sujet se référencent mutuellement [17].

## II.6 Propriétés topologiques des communautés

La notion de communauté fait apparaître des caractéristiques structurelles à l'échelle même de la communauté. Ces propriétés expriment l'intuition selon laquelle une communauté est un ensemble de nœuds avec une meilleure connectivité intracommunautaire qu'extra communautaire. Elles révèlent ainsi la façon dont un nœud est connecté à sa communauté et la façon dont les communautés sont connectées entre elles. D'autre part, elles renseignent sur la façon dont un réseau est partitionné en communautés et sur les propriétés structurelles de chaque communauté. [18]

- **La distribution de la taille des communautés** est une caractéristique importante de la structure de communauté. Les études menées jusqu'à présent sur les réseaux du monde réel tendent à montrer que la distribution de la taille des communautés suit une loi de puissance avec un exposant variant entre 1 et 2. Autrement dit, la taille des communautés est hétérogène avec la présence de quelques grosses communautés et de nombreuses petites communautés.
- **L'enracinement** (embeddedness) **e**, évalue la proportion de liens d'un nœud avec les nœuds de sa communauté. Il est défini comme le rapport entre le degré interne et $k_{int}$ et degré total $k$ des nœuds considérés :

$$e = k_{int}/k$$

Le degré interne correspond au nombre de liens qu'un nœud partage avec les nœuds de la même communauté. Le degré externe mesure le nombre de liens avec des nœuds situés dans d'autres communautés. L'enracinement maximal est atteint lorsque tous les voisins d'un nœud considéré sont dans sa communauté. La valeur minimale de l'enracinement correspond au cas où tous les voisins appartiennent à des communautés différentes.

Si l'on s'intéresse à la distribution de l'enracinement dans les réseaux réels, une majorité de nœuds, généralement de faible degré, ont un enracinement très élevé. Pour le reste cela dépend de la nature des réseaux étudiés. Les réseaux de communication, Internet et les réseaux biologiques présentent un pic vers la valeur d'enracinement de 0,5. Dans les réseaux sociaux et les réseaux d'information, la distribution est plus uniforme. Dans tous les cas, tout l'éventail des valeurs de l'enracinement est représenté, y compris les petites valeurs.

Notons que « encastrement » est le terme consacré pour cette propriété en sociologie.

- **La densité de liens** $p$ d'une communauté est définie comme le ratio des liens qu'elle contient par rapport au nombre de liens possibles si tous les nœuds étaient connectés:

$$\tilde{P}\ (C) = p(C)n_c = \frac{2m_C}{n_C - 1}$$

La densité normalisée (scaled density) $\tilde{P}$, est obtenue en multipliant la densité par la taille de la communauté. Comparée à la densité globale du réseau, la densité de la communauté permet d'en évaluer la cohésion. Une communauté est censée être plus dense que le réseau auquel elle

appartient. Si la communauté considérée est un arbre, sa densité normalisée vaut 2. Si c'est une clique, c'est-à-dire un réseau complètement connecté, sa densité normalisée est égale au nombre de nœuds de la communauté. La densité normalisée permet donc de caractériser la structure de la communauté. Dans certains réseaux réels comme Internet ou les réseaux de communication, les communautés ont une structure arborescente. Au contraire, pour les réseaux sociaux ou les réseaux d'information, la densité normalisée augmente avec la taille de la communauté traduisant ainsi une structuration en cliques. Enfin, les réseaux biologiques présentent un comportement hybride, leurs petites communautés étant plutôt en forme d'arbre alors que les plus grandes sont plus denses et proches d'une organisation en cliques.

- **La distance moyenne** d'une communauté permet également d'en évaluer sa cohésion. Dans les réseaux réels, les petites communautés, de taille inférieure à 10, sont supposées avoir la propriété petit monde. Ceci signifie que la distance moyenne doit augmenter de façon logarithmique avec la taille de la communauté. Pour les grandes communautés, la distance moyenne augmente, mais plus lentement, ou bien se stabilise pour certaines catégories de réseaux, comme les réseaux de communication. Une petite distance moyenne peut être expliquée par une forte densité dans le cas des réseaux sociaux, par la présence de hubs dans les réseaux de communication et Internet, ou même par les deux dans les réseaux biologiques, ou les réseaux d'information.
- **La dominance de hub** (hub dominance) $h$ , révèle la présence d'un hub central dans une communauté *C*. Elle peut être évaluée en utilisant la relation suivante :

$$h(C) = \max_c (k_{int})/(n_c - 1)$$

Où le numérateur est le degré maximal interne trouvé dans une communauté et le dénominateur est le degré maximal théoriquement possible étant donné la taille de la communauté. Cette mesure vaut 1 quand au moins un nœud est connecté à tous les autres nœuds dans la communauté. Elle est nulle si aucun des nœuds n'a de connexion interne à sa communauté, ce qui est peu probable pour une communauté. Dans les réseaux réels, on observe différents comportements. Ainsi, pour les réseaux de communication, on observe une valeur élevée dans pratiquement toutes les communautés et ceci indépendamment de leur taille. Ceci traduit donc la présence de concentrateurs dans toutes les communautés.

Considérant que leur structure est peu dense et sous forme arborescente, on peut en déduire que les communautés ont plutôt une structure en étoile. Ce phénomène est beaucoup moins

marqué pour les autres types de grands graphes de terrain. On peut même remarquer que la dominance de hub diminue lorsque la taille des communautés augmente.

## II.7 Algorithmes de détection de communautés

La découverte de communautés peut s'apparenter au partitionnement de graphe ou à L'apprentissage non supervisé. Néanmoins, contrairement au partitionnement de graphe, le nombre de communautés et leur taille n'est pas connue. De plus, contrairement aux techniques de clustering, on ne veut pas nécessairement mettre en avant une structure communautaire si le réseau en est dépourvu. Il faut néanmoins remarquer que la notion de communauté reste floue. Comme le fait remarquer Fortunato [19] *« The first problem in graph clusteringis to look for a quantitative definition of community. No definitionis universally accepted »*. Pour formaliser le problème, on peut définir la détection des communautés comme la partition des sommets du graphe en un nombre fini de sous-ensembles à partir d'un critère de qualité. Dans ces conditions, la détection de communautés consiste alors à trouver la partition qui optimise le critère de qualité. Ce problème étant NP-difficile, les techniques de partitionnement visent à trouver les meilleures partitions possibles tout en minimisant les coûts en temps et en espace.

De nombreux algorithmes de détection de communautés ont été proposés ces dernières années. Etablir une taxonomie de ces méthodes n'est pas chose aisée. Nous présentons par la suite quelques algorithmes qui ont reçu le plus d'attention par la communauté scientifique afin de donner une vue d'ensemble des méthodes proposées et d'en illustrer la diversité. Nous avons essayé de les classifier en fonction de leurs principes sous-jacents et des techniques utilisées. Bien que certains algorithmes puissent appartenir à plusieurs classes, nous avons choisi de les regrouper en quatre catégories différentes. La première classe contient les algorithmes hiérarchiques issus des approches classiques de clustering. La deuxième classe contient des algorithmes basés sur les processus de marche aléatoire. La troisième classe contient les algorithmes utilisant les propriétés spectrales des réseaux. Enfin, la dernière classe contient tous les autres algorithmes qui ne pouvaient être placés dans une des classes précédentes. Ceci permet par ailleurs de dégager quelques grandes lignes directrices parmi les techniques retenues. Pour une présentation plus exhaustive de l'éventail des solutions à la détection de communautés,. A la suite de la classification proposée pour les algorithmes, nous donnons la définition de la modularité et nous abordons une discussion sur son utilisation [18].

### II.7.1 Approches hiérarchiques

Les algorithmes hiérarchiques fournissent une structure de communauté qui peut être représentée sous une forme arborescente appelée dendrogramme. On peut distinguer deux types d'approches:

Les approches agglomératives dans lesquelles on se donne un critère pour fusionner des partitions. On peut ainsi fusionner les partitions qui sont les plus proches au sens d'une distance prédéfinie ou celles qui optimisent un critère de qualité du partitionnement. Au départ, chaque sommet est une communauté et on fusionne les deux communautés les plus proches. Ce processus est réitéré jusqu'à l'obtention d'une seule communauté ou jusqu'à ce qu'un critère prédéfini soit atteint.

- **L'algorithme glouton de Newman (FastGreedy)** est un algorithme typique de cette approche. Les sommets sont regroupés itérativement en partant d'une partition où chaque sommet représente une communauté jusqu'à obtenir une seule communauté regroupant tous les sommets. L'algorithme fusionne à chaque étape la paire de communautés qui permet d'avoir la plus grande augmentation de la modularité. La modularité est une fonction de mesure de la qualité d'une partition basée sur la proportion d'arêtes internes à chaque communauté. Nous présentons cette mesure en détail par la suite. La complexité de l'algorithme glouton est $(mn)$ [20].
- **L'algorithme de Bondel et al. (Louvain)** est aussi une approche gloutonne de clustering hiérarchique agglomérative. Il comprend deux étapes qui sont répétées de manière itérative. Au départ, chaque sommet représente une communauté. On calcule le gain de modularité obtenu en plaçant chaque nœud dans la communauté de ses voisins. Ensuite, les nœuds est placé dans la communauté pour laquelle ce gain est maximum, mais seulement si ce gain est positif. Si aucun gain positif n'est possible, le nœud reste dans sa communauté d'origine. On applique cette procédure à plusieurs reprises et de façon séquentielle pour tous les nœuds jusqu'à ce qu'il n'y ait plus d'augmentation possible de la modularité. La deuxième étape consiste à construire un nouveau réseau à partir de communautés identifiées à l'étape précédente et de reprendre la première phase sur le réseau ainsi formé. Les nœuds sont alors les communautés. Les liens intra communautés sont représentés par des boucles et les liens externes sont traités comme des liens entre voisins.

Les approches divisives qui scindent le graphe en plusieurs communautés en retirant les arêtes reliant des communautés distinctes. Les arêtes sont retirées une à une jusqu'à ce que tous les nœuds soient isolés. A chaque étape, les composantes connexes représentent les communautés. On obtient ainsi une structure hiérarchique de communautés. L'idée principale des algorithmes de division est donc de trouver les liens inter communautaires et de les retirer afin de séparer les communautés [21].

- **L'algorithme de Girvan et Newman (EdgeBetweenness)** est basé sur le nombre de plus courts chemins passant par une arête communément appelée la centralité d'intermédiarité (EdgeBetweenness). Cela suppose que peu d'arêtes relient les communautés et que les plus courts chemins entre deux communautés passent par ces arêtes. A chaque étape de l'algorithme on calcule la centralité d'intermédiarité des arêtes et on retire celle qui possède la valeur la plus élevée. La complexité de l'algorithme est en$O(m^2n)$ [22] .
- **L'algorithme de Radicchi et al.** (**Radetal**) est basé sur le coefficient de clustering d'arête qui mesure le nombre de cycles de longueur *g* passant par une arête, divisé par le nombre cycles possibles. Cela suppose que les arêtes inter communautaires sont peu clustérisées. A chaque étape on retire l'arête de plus faible *coefficient de clustering. Cet algorithme est néanmoins moins coûteux que le* précédent car la suppression d'une arête ne demande qu'une mise à jour locale des coefficients de clustering. Sa complexité est en $O(m^2)$ [23] .

### II.7.2 Algorithmes basés sur les marches aléatoires

Une marche aléatoire dans un graphe est un processus aléatoire dans lequel un marcheur est positionné sur un sommet du graphe et peut à chaque étape se déplacer aléatoirement vers un des sommets voisins avec une probabilité qui peut être différente pour chaque arête. Les probabilités de transition d'un sommet *i* à un sommet *j* définissent la matrice de transition de la chaîne de Markov associée au processus de marche aléatoire. La suite des sommets visités constitue une marche aléatoire. L'idée sous-jacente à cette classe d'algorithmes est qu'une marche aléatoire courte partant d'un sommet tend à rester dans la communauté de ce sommet.

Plus précisément, lorsqu'un marcheur est dans une communauté, il possède une forte probabilité de rester dans la même communauté à l'étape suivante (grâce à la forte densité de liens internes et la faible densité de liens externes). Ainsi, un marcheur possède de grandes chances

de rester lors d'une marche de courte distance dans sa communauté d'origine. Notons que la plupart des algorithmes dans ce type sont aussi des algorithmes hiérarchiques agglomératifs.

- **L'algorithme de van Dongen (Markov cluster)** repose sur la matrice de transition des marches aléatoires, d'où son appellation « Markov Cluster Algorithm » (MCL). Deux opérations sont utilisées de façon séquentielle pour simuler les marches aléatoires, l'expansion et l'inflation. L'opération d'expansion élève la matrice à la puissance *l*. Ce qui correspond aux probabilités de transition entre deux nœuds quelconques pour une marche de longueur *l*. L'opération d'inflation consiste à modifier les valeurs des probabilités de transition de façon à accentuer les différences. Autrement dit, on augmente les valeurs des transitions les plus probables afin de les favoriser et on diminue celle des moins probables afin de les pénaliser encore plus. Un paramètre permet de contrôler la vigueur de ces modifications. Après un certain nombre d'itérations de ce processus, on atteint une situation ou seules les transitions dans une même communauté sont possibles. La complexité de l'algorithme est en $O(n^3)$ [24].
- **L'algorithme de Pons et Latapy (Walktrap)** est basé sur une méthode hiérarchique agglomérative. Il utilise une distance entre communautés basée sur des marches aléatoires pour identifier les communautés les plus proches. L'hypothèse sous-jacente est que deux marches qui partent d'une même communauté vont avoir des comportements similaires et se concentrer sur les mêmes zones du graphe. L'idée pour comparer la proximité de deux sommets est alors de comparer les distributions de probabilité des marches aléatoires. La distance entre deux sommets *i* et *j* pour une marche de longueur *l* est donnée par la distance quadratique entre les deux vecteurs colonnes correspondant de la matrice de transition.

  Selon cette définition, deux sommets sont d'autant plus proches que le comportement des marches aléatoires partant d'eux est similaire. Une procédure agglomérative permet de générer une suite de partitions en fusionnant successivement des communautés deux à deux. Le choix des communautés à fusionner repose sur les distances entre les communautés adjacentes de la partition courante. Pour mesurer la proximité entre communautés, on généralise la notion de distance entre sommets en considérant des marches qui partent uniformément de l'ensemble d'une communauté plutôt que d'un sommet unique. La complexité de cet algorithme est en$O(mn\log(n))$[25].
- **L'algorithme de Rosvall et Bergstorm (Infomap)** exploite aussi le fait qu'un marcheur suivant aléatoirement les arêtes d'un graphe a tendance à rester bloqué dans sa

communauté. Dans cette approche, la détection de communautés se ramène à un problème de codage optimal au sens de la théorie de l'information. Le but est d'attribuer des codes décrivant les chemins sur le réseau qui découlent de processus de marche aléatoire en utilisant un codage de longueur minimale. Deux niveaux de codage sont utilisés. L'un distingue les communautés dans le réseau, l'autre distingue les nœuds dans une communauté. Les codes sont attribués à partir du codage de Huffman en utilisant les fréquences à laquelle les nœuds et les communautés sont traversés par les marches aléatoires. On associe ainsi des codes courts aux nœuds et aux communautés les plus visités et des codes plus longs à ceux qui le sont moins. Si l'on décrit une marche aléatoire par une séquence indiquant le code de la communauté de départ et les codes des sommets traversés hors de sa communauté, alors un bon partitionnement doit permettre de réduire la taille de cette séquence. En effet, les changements de communauté doivent être rares. La meilleure partition est trouvée en minimisant la quantité d'information nécessaire pour représenter une marche aléatoire dans le réseau en utilisant cette nomenclature. La détection de communautés proprement dite est obtenue par une approche gloutonne de clustering hiérarchique agglomérative similaire à celle utilisée dans Louvain [26].

### II.7.3 Algorithmes utilisant les propriétés spectrales des réseaux

Les algorithmes spectraux utilisent les représentations matricielles des réseaux pour détecter les communautés. La décomposition en vecteurs propres et valeurs propres est exploitée dediverses façons afin d'aboutir à une partition adaptée. [18]

- **L'algorithme de Newman (LeadingEigenvector)** est inspiré par les méthodes traditionnelles de partitionnement spectral. Ces algorithmes visent à trouver la coupe du graphe minimisant le « nombre » d'arêtes coupées entre les différentes parties. Minimiser ce coût de coupe revient à calculer le vecteur propre correspondant à la plus petite valeur, propre non nulle de la matrice Laplacienne du graphe. Le graphe est alors séparé en deux parties en fonction du signe de leur composante selon ce vecteur propre. Cette méthode est particulièrement adaptée lorsque les partitions sont de taille comparable. Ceci n'est ni approprié ni réaliste pour des problèmes de détection de communautés. Afin d'adapter cette technique à la détection de communautés, Newman a proposé de substituer la maximisation de la modularité à la minimisation du coût de coupe. Il a montré que le problème peut s'exprimer à partir de la matrice de

modularité. Son rôle est identique à celui joué par la matrice Laplacienne pour la minimisation du coût de coupe. La partition obtenue permet de trouver simplement deux communautés. L'algorithme utilise ensuite une approche hiérarchique divisive en n'acceptant pas les divisions qui ne permettent pas un gain de la modularité [27].

- **L'algorithme de Donettiet Muñoz (Comfind)** utilise les propriétés spectrales de la matrice Laplacienne du graphe conjointement à des techniques de clustering. La décomposition spectrale est utilisée pour projeter les nœuds du réseau dans un espace des vecteurs propres de dimensionnalité variable. Chaque nœud est ainsi représenté par un point dans un espace D-dimensionnel. Une métrique (euclidienne ou angulaire) est utilisée afin de calculer les distances entre nœuds pour pouvoir appliquer une technique de classification hiérarchique agglomérative. Le nombre de vecteurs propres à utiliser est inconnu. Plusieurs calculs sont donc effectués avec des nombres différents de vecteurs propres. Le meilleur résultat au sens de la modularité est retenuv [28].

## II.8 Conclusion

Les réseaux sociaux ont toujours été présents dans la vie sociale de l'être humain. Aujourd'hui, la puissance des réseaux sociaux a augmenté grâce à l'Internet et les enjeux dans ce domaine sont multiples tant pour les utilisateurs que pour les acteurs et les décideurs.

Du coté des utilisateurs, les réseaux regroupent toutes sortes d'individus provenant de tous les secteurs. Il est possible de retrouver quasiment n'importe quelle personne quelque soit sa situation géographique. En plus de pouvoir visualiser des informations sur les personnes, les réseaux sociaux offrent l'opportunité d'entrer en contact avec toutes ces personnes. Les liens entre les membres d'un réseau social sont les profils personnalisés, les identités numériques.

Cependant, dans un monde numérique ou les utilisateurs ne sont pas en présence physique l'un de l'autre, si on veut atteindre le même niveau de sécurité que celui du monde réel, dans lequel nous devons nous identifier, le passage par le mécanisme sécurisé d'identité numérique est essentiel. Un tel mécanisme peut être assuré par la carte d'identité numérique.

## III.1 Introduction

Dans ce chapitre, nous proposerons une nouvelle approche de détection de communautés. Cette approche nécessite la connaissance détaillée des utilisateurs et des services web. Pour cela, notre travail va être devisé en deux tâches principales à savoir la définition des éléments composants les profils utilisateurs et services web dans un premier lieu, ensuite, essayer de concevoir une méthode générale pour détecter les communautés. Cependant, la première tâche reste un travail sensible et extensible qui nécessite une réflexion approfondie pour définir les éléments adéquats des profils afin de bien concrétiser la tache de détection de communautés.

## III.2 L'idée proposée:

Rappelons que dans l'architecture AOS, un service web est offert par un fournisseur de service qui publie la description de son service dans le répertoire d'enregistrements UDDI construit à cet effet. Lorsqu'un utilisateur cherche un service web, il prend connaissance des services disponibles dans UDDI grâce aux définitions WSDL diffusées par les fournisseurs de service. Dès qu'un service web est choisi par l'utilisateur, le lien entre ces deux entités est établi et l'utilisateur peut accéder au service.

Notre méthode de découverte de communautés nécessite de connaitre les informations concernant l'accès des utilisateurs aux services web. Cependant, ces informations ne sont disponibles que sur des plateformes payantes. De ce fait, nous avons fait une simulation des services et des utilisateurs ainsi que de leur description et nous les avons modélisés par un graphe biparti[2] dans lequel le premier sous ensemble de sommets représente les profils des utilisateurs et le deuxième sous ensemble les profils des services. Le sous ensemble des profils des services représente en réalité l'UDDI de l'architecture AOS, Chaque arête du graphe qui lie un service à un utilisateur veut dire que cet utilisateur a eu accès à ce service.

Une fois les profils utilisateurs et services crées, nous avons modélisé le processus de découverte des services web adéquats aux besoins de l'utilisateur. Ensuite, nous avons modélise le

[2] C'est un graphe dont l'ensemble des sommets peut être partitionné en deux parties X et Y telles que toute arête a une extrémité dans X et l'autre dans Y. [Graphes bipartis - affectation, transport, mariages stables, Pauline Sarrabezolles, 11/12/2013]

processus d'accès des clients aux services web. Au terme de ces deux processus, nous avons implémenté le processus de découverte de communautés. Dans ce qui suit, nous détaillerons chacun de ces trois processus.

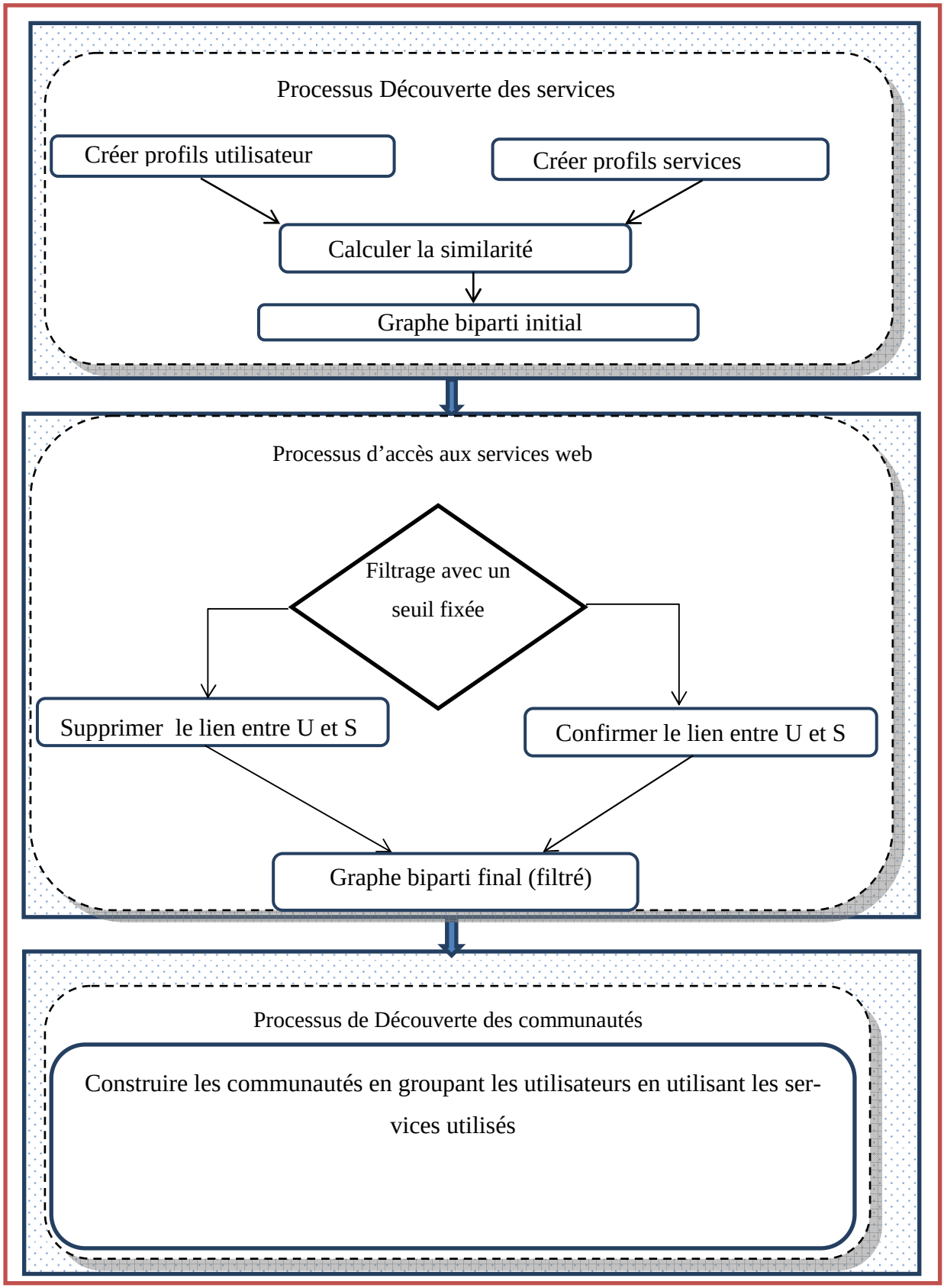



**Figure III.1 : les processus itérâtes de notre approche**

### III.2.1 Processus de découverte des services web

Le processus de découverte a pour objectif d'offrir aux utilisateurs les services web les plus similaires à leurs besoins.

Dans AOS, les descriptions des services web ne sont faites qu'au niveau syntaxique. Leur découverte est donc essentiellement basée sur ce paramètre. Cependant, vu la diversité des utilisateurs et des conditions dans lesquelles ils accèdent aux services web, d'autre paramètre doivent être considérés lors de la découverte, tels que le type de dispositif utilise (PDA, ordinateur portable etc), les préférences de l'utilisateur, sa localisation, etc. Tous ces paramètres forment un contexte d'utilisation particulier.

Notre but est de modéliser un processus de découverte de service, qui réponde le mieux aux exigences de l'utilisateur en termes de description de service, tout en étant en adéquation avec les diverses caractéristiques qui constituent son contexte d'utilisation .Pour cela, il est nécessaire de fournir une description enrichie des services web qui permettra la prise en compte du contexte de l'utilisateur. La découverte des services utilisera une mesure de similarité qui se base sur l'interaction des utilisateurs à travers les services web.

Avant de décrire le processus de découverte proposé et la mesure de similarité utilisée, nous allons définir les composants constituant les profils des utilisateurs et des services web que nous avons considérés.

#### III.2.1.1 Profils utilisateur et service

Les deux entités entrant en jeu lors de la découverte de services web sont les utilisateurs et les services web eux-mêmes. Chaque utilisateur du web possède un profil ou sont consignées un ensemble d'informations le concernant (nom, localisation, préférences ……..). Toutes ces informations fortement son identité numérique. Cependant le processus de découverte sa basera uniquement sur une partie de ces caractéristiques, communes à tous les utilisateurs, et qui sont rassemblées sous le terme contexte. Ainsi, le contexte utilisateur est formé par divers paramètres tels que : son identité, les services demandés, ses préférences, le type de terminal utilisé, localisation etc. Le contexte du service quant à lui est caractérise par les paramètres suivants : le coût d'utilisation du service, les restrictions géographiques du service, les types de terminaux pris en charge, la qualité de service etc. Dans ce qui suit, nous définissons un format de profil universel qui permet de modéliser aussi bien le contexte de l'utilisateur solli-

citant le service que celui de service web offert, au moyen d'une structure général permettant le partage d'un vocabulaire commun pour la représentation des données relatives aux deux entités.

Notons qu'un profil, qu'il s'agisse d'un profil utilisateur ou d'un profil service, est susceptible de changer dans le temps suite à des évènements .Par exemple, un utilisateur peut vouloir modifier un attribut de son profil ou peut être amène à ajouter une valeur à un attribut. C'est pour cette raison qu'une représentation linéaire est inappropriée pour la représentation du profil. Notre choix s'est donc porté sur la modélisation du profil par un arbre. Ce choix se justifie par le fait qu'il permet une description du profil, au moyen d'une structure générale qui prend une forme standard en faveur de tous les usagers. La structure ainsi définie est flexible, dons le sens ou différentes caractéristiques peuvent être étendues grâce au caractère arborisant de la description proposée qui permet son extension.

L'arbre contient un ou plusieurs composants, qui contiennent à leur tour un ou plusieurs attributs. Le schéma suivant une représentation générale du contexte utilisé .C'est sur ce contexte que le processus de découverte de service va s'appuyer.

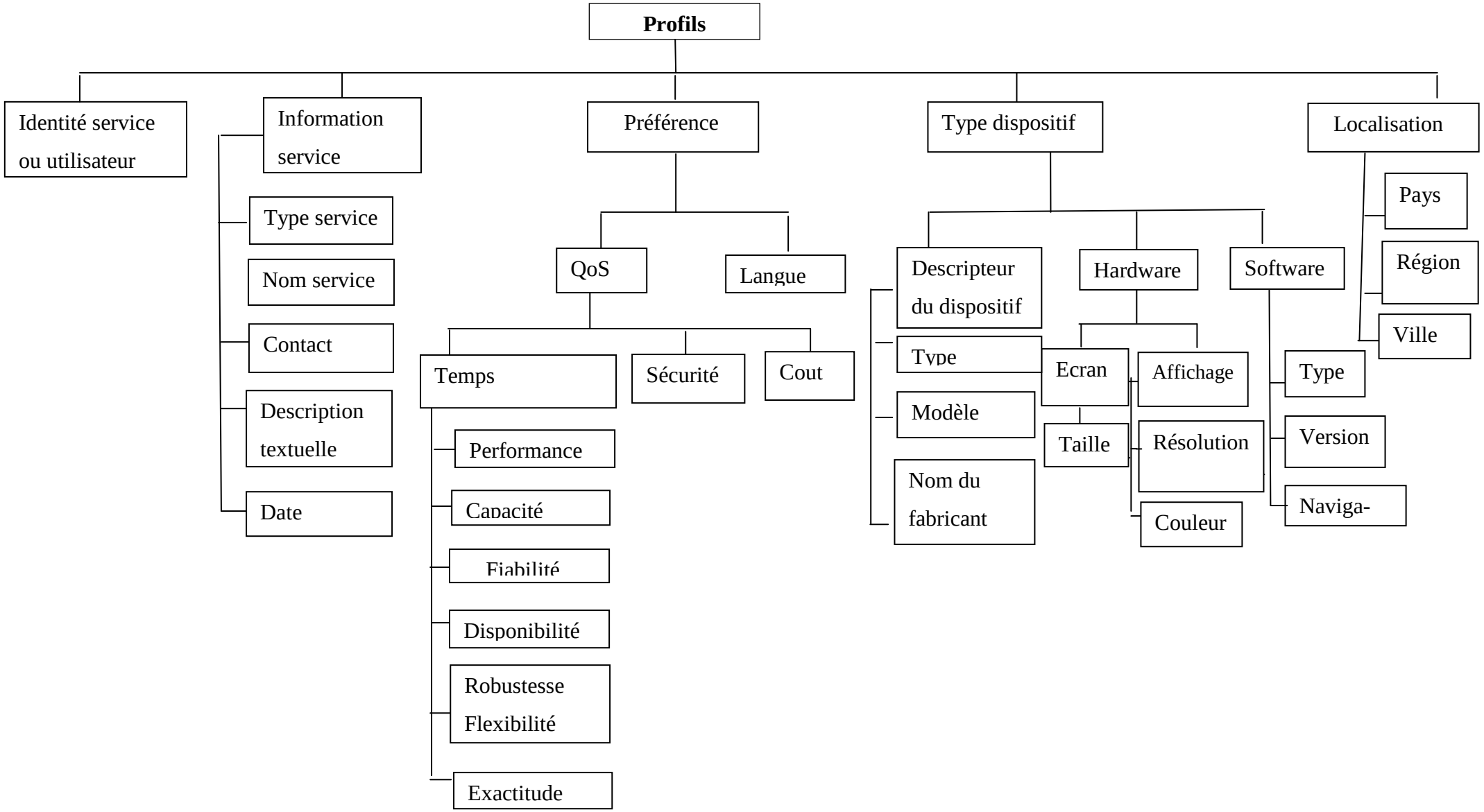



**Figure III.2:Représentation graphique du contexte**

Dans ce qui suit, nous détaillons les paramètres représentant le contexte schématisé préalablement, selon les deux points de vue (utilisateur et service). Nous distinguons cinq composants selon le type d'information fournit :

**1. Identité service/utilisateurs**

Cet attribut spécifie pour chaque profil service et utilisateur un indentant unique.

**2. Information service**

Ce composant spécifie quatre informations :

**Nom de service :** Pour le service, il correspond au nom du service offert. Pour l'utilisateur, cet attribut réfère au nom du service recherché ;

**Contact :** Il offre pour le service au mécanisme de référence à l'organisation fournissant le service. Pour l'utilisateur, cet attribut réfère au nom de l'entreprise à laquelle le service recherché appartient ;

**Description textuelle du service :** Du point de vue service, il décrit les fonctionnalités qu'offre celui-ci. Pour l'utilisateur, cet attribut fournit une brève description de ce que doit réaliser le service ;

**- date :** pour le service cet attribut représente la date de publication du service web. Pour l'utilisateur, il spécifie la date de publication du service recherché.

**3. préférence :**

Ce paramètre englobe deux aspects :

**a) Les paramètre de qualités de service (QoS)**

Pour l'utilisateur, ces paramètres représentent la qualité de service web recherchée. Pour le service, ils représentent la qualité de service web fournie.

On distingue :

Les attributs de QoS liés au temps d'exécution

**- Capacité :** c'est le nombre maximum de requêtes simultanées que le système prend en charge avec des performances garanties ;

- **Performance** *:* c'est la vitesse d'exécution d'une requête ;
- **Fiabilité :** capacité d'un service à exécuter ses fonctions requises dans des conditions indiquées pendant une période donnée ;
- **Disponibilité :** c'est la probabilité que le système soit actif ;
- **Robustesse /Flexibilité :** représente le taux lequel un service peut s'exécuter correctement dans le cas de données d'entrées inadmissibles, incomplètes ou contradictions ;
- **Exactitude :** représente le taux d'erreur produit par le service.

**b) La langue :**

Pour l'utilisateur ce paramètre réfère à la langue dans laquelle il souhaite recevoir le service pour le service, il représente la langue avec laquelle le service peut être fourni.

**4. Type de dispositif :**

Ce paramètre procure un vocabulaire commun aux fournisseurs de services pour décrire les différents types de dispositifs pouvant supporter les données transmises par leurs services web et aux utilisateurs afin de décrire les contraintes matérielles et logiciel du terminal avec lequel ils accèdent aux différents services web.

Ce paramètre englobe trois aspectes :

**a) La description du dispositif**

- Type
- Modèle
- nom de fabricant

**b) Le hardware**

-Taille : taille de l'écran

-Résolution

-type d'affichage

**c) Le software :**

Ce paramètre définit le type et la version du système d'exploitation et du navigateur utilisés.

-Type : type du système d'exploitation.

-Version : version du système d'exploitation.

-Navigateur.

**5. Localisation**

Ce composant permet à l'utilisateur de décrire sa localisation et au fournisseur d'introduire les restrictions d'usage géographique du qu'il propose. Ce composant spécifie les trois attributs suivants :

- Pays, Région, ville.

Les profils utilisateurs et services présent définis, il s'agit maintenant de les comparer, le but étant de trouver les services web les plus adéquats à ceux recherchés par chaque utilisateur, c'est- à-dire les plus similaires à chaque profil.

### III.2.1.2 Mesure de la similarité

La mesure de similarité est le calcul de ressemblance entre un profit utilisateur et un profil service afin de construire le graphe initial de simulation. Chaque profil est modélisé par un tableau d'enregistrement contenant la position de l'attribut dans l'arbre et sa valeur.

Alors beaucoup de descriptifs ont des valeurs manquantes que ce soit pour un profil utilisateur ou service, il faut une mesure de similarité qui prend en charge ce fait. Donc, la mesure de similarité quantitative **[29]** répond parfaitement à ce problème.

QSIM : P * P à [0,1] est définie comme suit :

$$\text{QSIM (X,Y)} = \frac{a \sum_{i=1}^{i=n} Wi \; X \; ASIM\;(Xi,Yi)}{a \; X \; \sum_{i=1}^{i=a} Wi \; + \; b \; X \; \sum_{i=1}^{i=b} Wa{+}i \; + c \; X \; \sum_{i=1}^{i=c} Wa{+}b{+}i}$$

_ X et Y sont deux profils appartenant à P.

_ a est un ensemble des caractéristiques communes de X et Y.

_ b est un ensemble des caractéristiques existantes en X et non existantes en Y.

_ c c'est l'ensemble des caractéristiques existantes en Y et non existantes en X.

_ ASIM est la similitude atomique entre chaque caractéristique de X et Y.

_ $w_i$ poids.

Dans notre cas on a affecté pour les paramètres b et c la valeur 0 et pour $w_i$ la valeur 1

La similarité atomique (ASIM) est une application définie de R*R et à valeur dans [0,1] :

$$ASIM : R*R \longrightarrow [0,1]$$

$$Sima(x_i, y_i) = \begin{cases} 1 & Si \quad x_i = y_i \\ 0 & Si \quad x_i = 0 \quad ou \quad y_i = 0 \\ \frac{\min(x_i, y_i)}{\max(x_i, y_i)} & sinon \end{cases}$$

-x et y : deux profils à comparer

-$x_i$ (resp.$y_i$) : valeur associée à l'attribut i dans le profil x (resp.y).

-$x_i$=0 (resp.$y_i$=0) signifie qu'aucune valeur n'été affecté à l'attribut i dans le profil x (resp.y).

### III.2.2 Processus d'accès aux services web

Le processus de découverte des services web génère un graphe dont les arêtes indiquent que les contextes utilisateurs et service sont similaires, mais pas que l'utilisateur a véritablement accéder à un service.

Le processus d'accès vise à déterminer pour chaque utilisateur, quels sont les services auxquels il a accédé parmi ceux qui lui sont similaires. L'objectif de ce processus est donc de créer un sous graphe du graphe précédent dans lequel une arête qui lie un utilisateur à un service signifie que l'utilisateur a accédé à ce service.

Cependant, rappelons que le but final de notre approche est la détection de communautés basée service, qui reposera sur le graphe obtenu lors du processus en cours. Or, un nombre limité d'accès à un service ne peut pas faire d'un utilisateur un membre de la communauté relative à ce service. En effet, il peut y a voir accéder par erreur ou simplement par curiosité. Ainsi, un paramètre important à prendre en considération lors de la construction du graphe d'accès est le nombre d'accès d'un utilisateur à un service d'où la nécessité de sauvegarder cette information au niveau de chaque service. Chaque arrête du graphe précédent sera donc pondérée

par la fréquence d'accès de l'utilisateur au service. Dans ce processus, on ne gardera que les arrêtes dont le poids est supérieur à seuil fixé.

### III.2.3 Processus de découverte des communautés

Avant de présenter notre méthode de découverte des communautés basée service, nous allons dans un premier lieu présenter une méthode existante de détection de communautés dans les réseaux sociaux (voire le chapitre 2). Nous évoquerons ensuite l'ensemble des faiblesses détecté dans celle-ci, puis nous présenterons les solutions apportées par notre méthode.

#### III.2.3.1 Faiblesses de la méthode existante

- **Type d'interaction inconnu**

Notons qu'une arête qui lie deux nœuds indique que les utilisateurs associés à ces nœuds partagent au moins un service commun, mais qu'aucune information ne fournit l'identité du service partagé. Notons également que deux clients, qu'ils partagent un ou plusieurs services communs, seront liés par une seule arête : nous ignorons donc le nombre de services qu'ils sont partagés. Au terme de la détection, les communautés générées seront formées de nœuds ayant une forte interaction certes, mais cela est sans grand intérêt puisque nous ne connaissons pas l'objet des interactions.

- **L'utilisateur : au centre des interactions**

Cette méthode base son analyse sur un graphe ou la seule entité représentée est l'utilisateur. Ces dernières années, avec l'explosion des réseaux sociaux, le nombre d'utilisateurs n'a cessé d'augmenter et les informations les concernant aussi. Or, il n'existe pas de base de données pour les utilisateurs comme il peut en exister web (UDDI). Il est ainsi difficile de connaître au préalable la communauté de tous les utilisateurs à analyser. Cette méthode n'est donc approprie aux réseaux complexes actuels.

#### III.2.3.2 Notre approche de détection des communautés basé service

Notre approche de détection de communautés a pour objectifs de décomposés les utilisateurs en communautés .Chaque communauté fait référence à un /plusieurs service(s)et sera composée des utilisateurs utilisant /ces service(s)

**Etape1** : Le traitement de notre approche commence par détecter les communautés relatives à un seul service .C'est à ce niveau que l'on réalise l'intérêt de la modélisation du problème par un graphe biparti .En effet, considérant le graphe d'accès obtenu dans le processus précédent, il suffit pour chaque service, de déterminer les arêtes incidentes au nœud correspondant à ce service. Les nœuds utilisateurs adjacent à ces arêtes formeront la communauté relative à ce service.

Cependant, la détection de communautés relatives à un seul service n'est pas suffisante. Le but est de découvrir des communautés relatives à un groupe de services.

**Etape2 :** La décomposition du réseau à l'étape précédente se caractérise par des chevauchements entre communautés. Cela résulte du fait qu'un utilisateur peut accéder à plusieurs services web, et peut par conséquent appartenir à plusieurs communautés.

Dans cette étape, on considère deux à deux les communautés obtenues à l'étape précédente et qui se chevauchent. Pour chaque paire de communautés considère on crée une nouvelle communauté relative à la fusion de leur services. Elle sera formée de plusieurs extraits de leur intersection. On réitère cette étape jusqu'à ce qu'il n'y ait plus de chevauchement entre les communautés.

Cette méthode de détection est récursive : chaque itération nécessite le résultat de l'itération précédente.

## III.3 Diagramme de classe de l'application

La figure ci-dessous présente le diagramme de classe de notre idée de base :

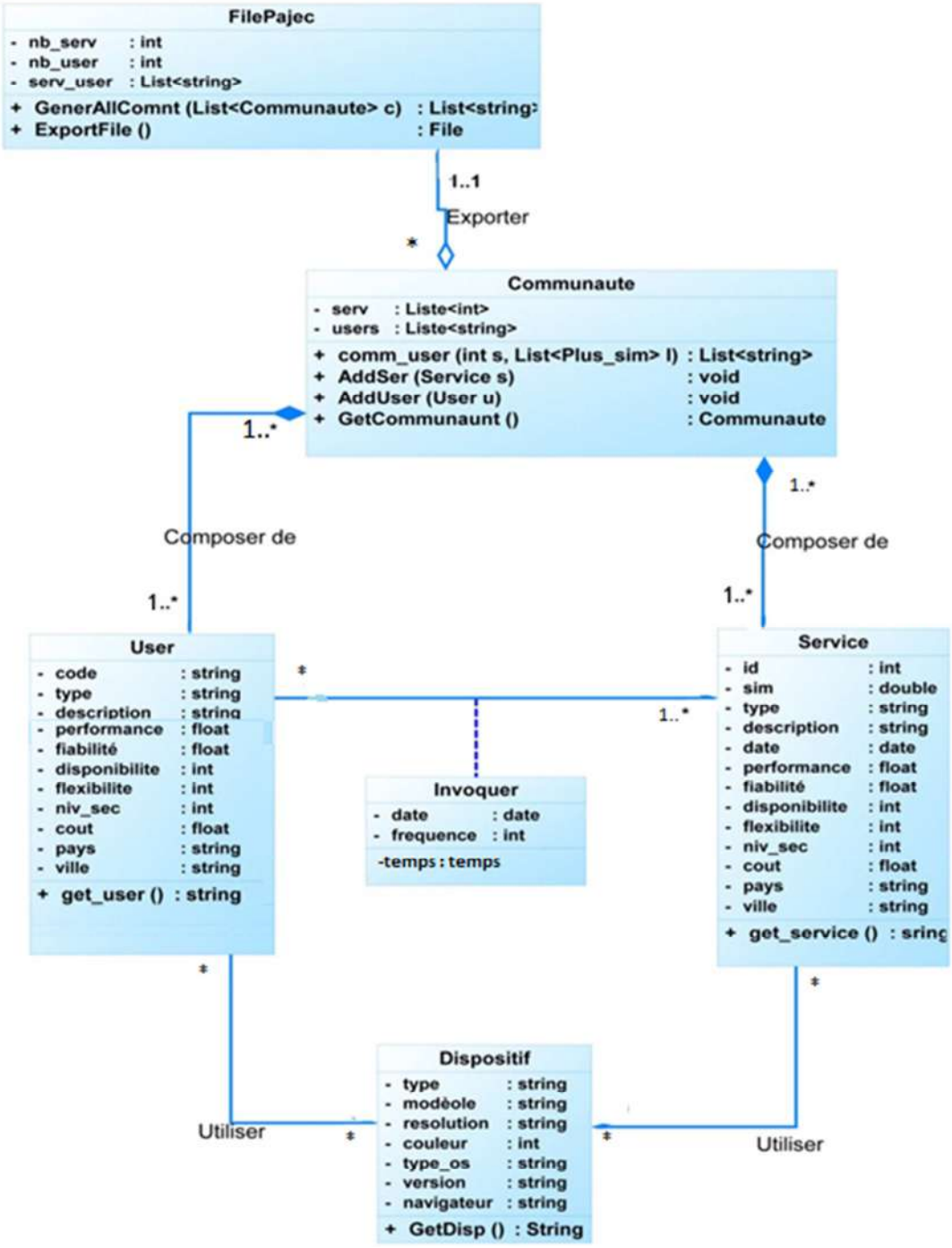



Figure III.3 : Diagramme de classe de l'application

Notre conception utilise plusieurs utilisateurs qui peuvent accéder à plusieurs services. Cependant, Chaque utilisateur (service) peut utiliser un ou plusieurs dispositifs. Après calcule de la similarité entre les profils services et les profils utilisateurs, et à la fin de la détection des communautés, nous devront sauvegarder la structure dans un fichier afin de le visualiser dans un logiciel de visualisation des graphes (pajek).

## III.4 Algorithmes proposés

Cette partie couvre les algorithmes nécessaires pour la mise en œuvre (l'implémentation) de notre approche « découverte les communautés basé services ». La première phase concerne de découvrir des services web, la découverte utilise une mesure de similarité,( voir Algorithme01 Page 51). La deuxième phase concerne de l'accès aux services web, c'est-à-dire de déterminer pour chaque utilisateur, quels sont les services auxquels il a accédé réellement parmi ceux qui lui sont similaires (Algorithme 02), La troisième phase concerne la découverte des communautés (Algorithme 03).

### III.4.1 Algorithme de calcule de similarité

#### III.4.1.1 Principe de fonctionnement

Rappelons que notre problème consiste, à calculer la similarité entre les utilisateurs et services afin de détecter les communautés. Pour cela, nous avons basé sur des formules mathématiques. L'algorithme développé dans cette partie consiste à associer les services aux les utilisateurs.

L'algorithme prend en paramètre deux tableaux (utilisateurs et services), le résultat de cet algorithme est un graphe biparti dont l'ensemble des liens entre les utilisateurs et les services, si leur taux de similarité est supérieur à un seuil fixé ,Cependant le contexte utilise contient aussi bien des valeurs numériques que des chaines des caractères. Donc, nous soumettons d'abord les profils généré au traitement de la conversion numérique (voir Annexe) a fin d'utilisé la similarité quantitative.

Pour calculer la similarité entre un utilisateur et un service on utilise la fonction « mesure similarité», cette fonction fais la somme des similarités de chaque paire d'attribut de profil (utilisateur/service), qu'elle est devisé par le nombre totale des attributs, le pourcentage obtenu

correspond au taux de similarité des deux profils, on applique cette fonction pour tous les services et les utilisateurs.

### III.4.1.2 Présentation de l'algorithme

**Algorithme01 similarité**

**Input** : (utilisateurs, services) ;
**Output** : Graphe ;

**Var**

d : réel

**Début**

d= 0 ;

ps = {} // chaque élément de cette liste compris un utilisateur avec ses services les plus similaire

**Pour** tous $U_i$ ∈ utilisateurs **faire**

ss = {} ; //liste pour compris les services les plus similaire

**Pour** tous $S_j$ ∈ services **faire**

d = mesure_similarité ($U_i$,$S_j$) ;

**Si** (d >50) **alors**

Ajouter ($S_j$, ss) ; // ajouter le service Sj dans la liste ss

**Finsi**

**Finpour**

Ajouter (ps,$U_i$, ss) ; //ajouter l'utilisateur $U_i$ avec ses service ss dans la liste ps

**Finpour**

**Retourner** ps ;

**Fin**

## III.4.2 Algorithme d'accès au service

### III.4.2.1 Principe de fonctionnement

Dans cette phase nous avons déterminé pour chaque utilisateur, quels sont les services auxquels il a accédé parmi ceux qui lui.

Cet algorithme a été développé pour filtrer les services invoquer par un utilisateur, il prend en paramètre un graphe bipartie obtenu dans le premier algorithme. Il teste le nombre le nombre d'accès de chaque utilisateur aux ses services similaire s'il trouve ce nombre est supérieur ou égale à un seuil fixé (10 dans notre cas) il confirme le lien, dans le cas contraire supprimer le lien. Le résultat obtenu est un graphe biparti final

### III.4.2.2 Présentation de l'algorithme

**Algorithme02 accé_service**

**Input**: Graphe (ps) // le graphe obtenu dans la première phase
**Output** : Graphe // le graphe après l'optimisation
**Var**
Fréquence : entier ;

**Début**

**Pour** i = 1 à taille (ps) **faire**

**Pour** tous $S_j \in$ (ps[i]$\rightarrow$ servs) **faire**// liste de services plus similaire pour l'utilisateur i

Fréquence = get_fréquence (ps[i] $\rightarrow$user,$S_j$); //obtenir le nombre
//d'utilisation du service Sj par l'utilisateur i

**Si** (Fréquence < 10) **alors**

Supprimer le lien entre (ps[i]$\rightarrow$ user, $S_j$);

**Finsi**

**Finpour**

**Finpour**

**Retourner** ps

**Fin**

### III.4.3 Algorithme de détection de communautés

#### III.4.3.1 Principe de fonctionnement

L'algorithme implémenté dans cette partie consiste à découvrir tous les communautés, à partir du graphe 1 de l'algorithme précèdent en plus un graphe 2 vide. Il fait les étapes suivantes :

- intersections entre tous les communautés du graphe 1
- ajouté les nouveaux intersections dans le graphe 2
- fusionner les deux graphes
- faire un appelle récursive
- l'algorithme arrêté quand il n y a pas des nouveaux intersections

### III.3.2 Présentation de l'algorithme

**Algorithme Communautés**

**Input** : Liste de communautés ($c_1$, $c_0$)

**Output** : Liste de Communautés ($c_1$)

**Début**

Communautés $c_2$ = {} ; //la liste qu'il va contenir les nouveaux communautés détecté

**Pour** toute $c_i \in c_0$ **faire**

Communauté $c$ = Intersection **($c_i$, $c_j$);// j ≠i**

**Si** (non_vide($c$)) **alors**

**Si** ($c \notin c_2$) **alors**

Ajouter**($c$, $c_2$) ;**

**Finsi**

**Finsi**

**Finpour**

**Si** (est_vide($c_2$)) **alors**

Fusionner ($c_0$, $c_1$) ; //Fusionner la liste de communautés $c_0$ avec $c_1$

**Retourner** $c_1$ ;

**Sinon**

Sup_s_com ($c_1$) ; //Supprimer les sous communautés inclus dans les nouveau communautés
Fusionner ($c_0$, $c_1$) ;

**Retourner** Communauté ($c_1$, $c_2$) ; //Calculer les nouveaux communautés à partir de C2

**Finsi**

**Fin**

## III.5 Conclusion

Dans ce chapitre, nous avons présente une nouvelle méthode de détection de communautés basée service. Comme nous l'avons vu, notre approche a donné à la détection de communautés une autre dimension dans le sens où ce n'est plus l'utilisateur qui est au centre des interactions, mais le service. Cela veut dire que nous pouvons donner un sens aux communautés détectées et voir les sujets du regroupement. Aussi, le faite de présenter les contextes d'utilisateurs et de services par un graphe biparti facilite énormément la détection dans les réseaux complexes. C'est un autre point fort de notre démarche de découverte des communautés.

Dans le prochain chapitre, nous allons maitre en œuvre notre idée de conception avec les outils nécessaires.

## IV.1 Introduction

Dans ce chapitre nous allons présenter l'implémentation et la mise en œuvre de notre approche par des outils convenables, puis nous allons présenter l'interface de notre application en expliquant les différentes fonctionnalités disponibles, nous terminons ce chapitre par un test pratique pour montrer et valider l'efficacité de l'approche.

## IV.2 Les outils et les langages utilisés en développement

### IV.2.1 visuel studio.Net

Est un environnement de développement intégré (EDI) de Microsoft qui regroupe :

- Plusieurs langages de programmation
- Un ensemble de bibliothèque pour le développement

Il permet ainsi un développement plus rapide, plus performant et facile à maintenir. Il est utilisé pour développer les applications Windows, application web, site web, services web, le .NET contient aussi un « moteur » appelé le CLR qui permet d'exécuter votre application indépendamment de l'OS de l'architecture sur lequel tourne votre application

Les principaux langages

- C #
- VB.Net
- C++.Net
- J#

Ci-dessous nous voyons l'environnement de développement intégré (EDI) de visuel studio :

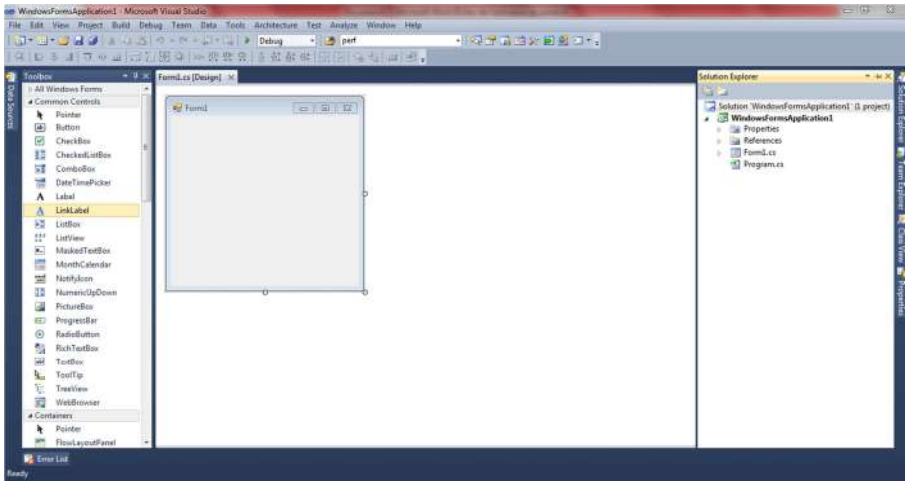


**Figure IV.1: L'interface principale de l'environnement de développement**

### IV.2.2 le langage C#

Pour la programmation de notre application, nous avons utilisé le langage C#. C# est un langage apparu en 2001 et spécialement crée par Microsoft pour être utilisé avec le Framework .NET. Microsoft a également sorti durant cette même année Visual Studio, son IDE (Integrated Development Environement) dédié au .NET [30].

### IV.2.3 SQL Server 2008

Produit par Microsoft, SQL Server 2008 est un système de gestion de bases de données relationnelles. Le stockage, la manipulation et l'analyse de ces données se font au sein de son moteur de bases de données. Ce service permet la réalisation de nombreuses applications, requêtes, et transactions, notamment grâce au langage T-SQL (Transact-SQL). Pour plus de détails sur le moteur de SQL Server 2008, des cours sont disponibles dans la partie « SQL Server 2008 Implémentation and Maintenance » [31]

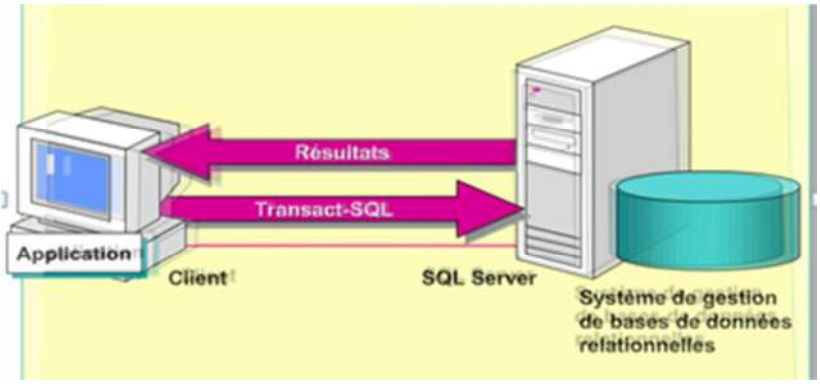



**Figure IV.2: Architecture de fonctionnement SQL Server 2008**

### IV.2.4 L'éditeur de graphes « pajek »

C'est un logiciel gratuit - mais non open source, tourne sous Windows, et Linux, il permet d'étudier des graphes de grande taille (plusieurs milliers de nœuds[3]), il est compatible avec les autres logiciels existants (notamment Ucinet et R).

[3]. Pour les mathématiciens, la taille d'un graphe revoit au nombre de liens, le nombre de sommets désigne l'ordre du graphe.

Pajek utilise des menus déroulants à la mode Windows ou Mac. Il convient d'abord de sélectionner les objets à étudier dans la fenêtre principale, puis de dérouler les menus de la barre supérieure pour trouver les commandes appropriées. Les trois principaux objets manipulables avec Pajek sont les réseaux, les partitions et les vecteurs [32].

**Le réseau** c'est une liste de sommets et une liste de liens (éventuellement signés ou values).

**Une partition** permet de classer les sommets en différents groupes et, plus généralement, d'attribuer un nombre entier à un sommet. Ainsi, je peux créer une partition codant des villes en 1, 2 et 3 selon leur statut administratif.

**Les vecteurs** attribuent à chaque sommet des réels (entiers ou décimaux).

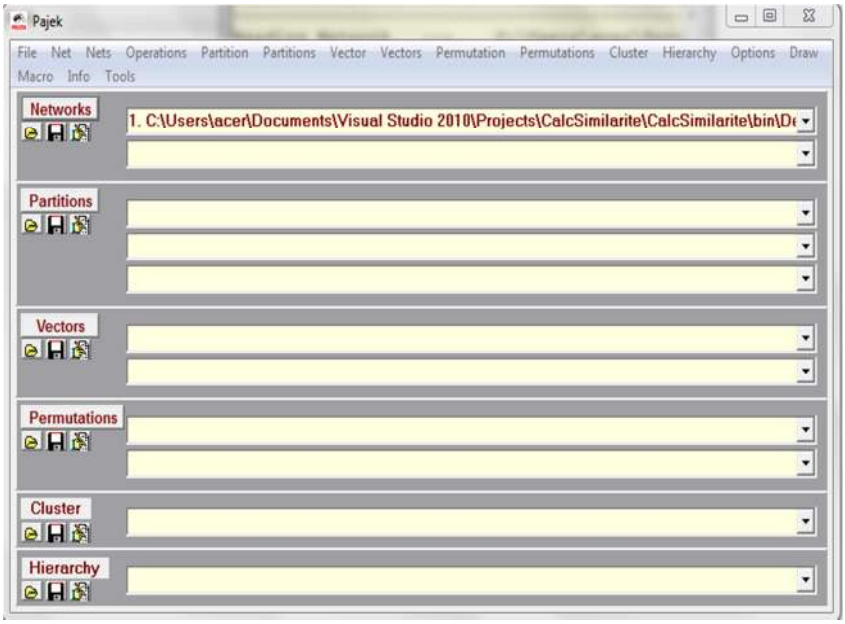



**Figure IV.3:L'interface principale du pajek**

## IV.3 Présentation de notre application

On a programmé notre application en utilisant C#, et on a utilisé une base de données implémenter par SQL Server contient les utilisateurs et les services web. L'application est utilisée pour calculer la similarité entre les utilisateurs et services web afin de détecter les communautés. La figure ci-dessous montre l'interface de l'application.

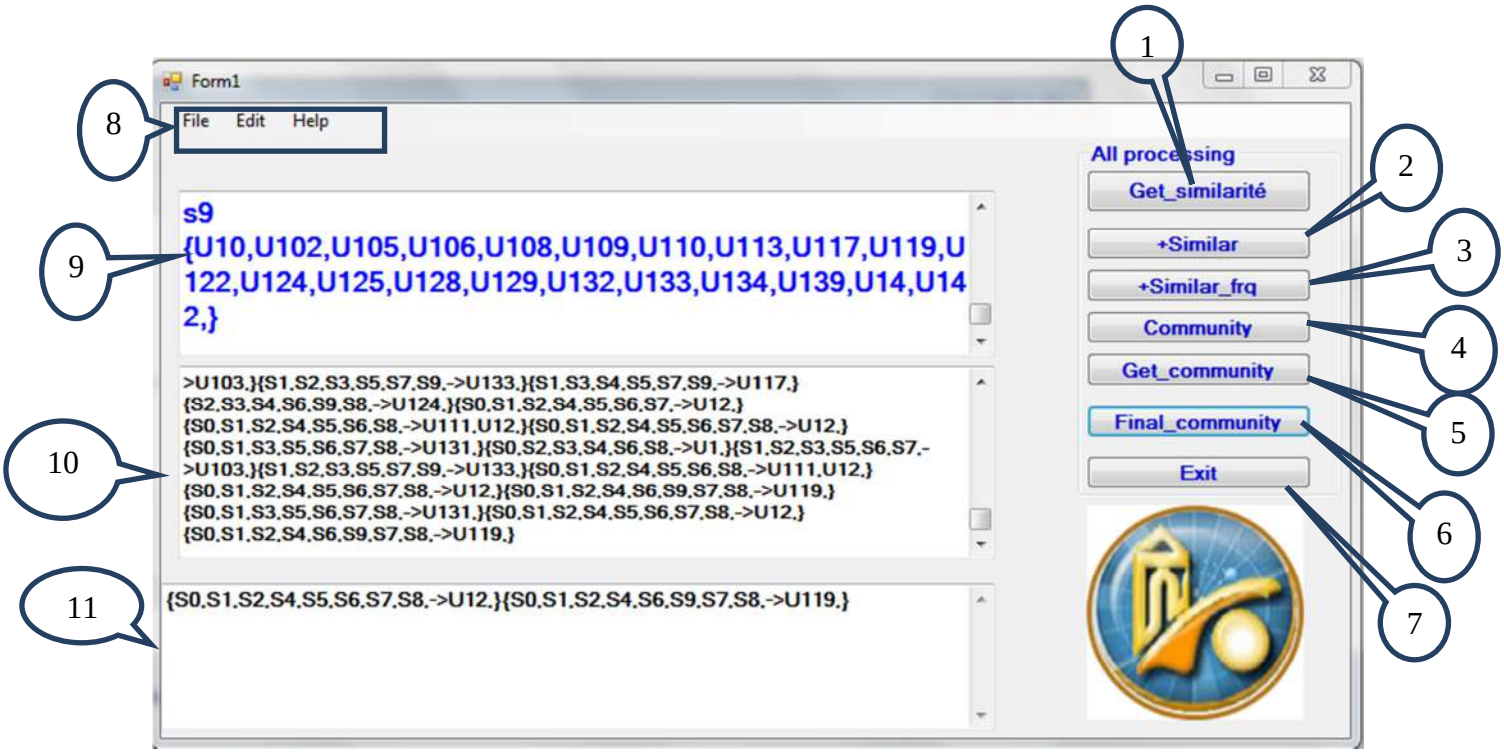



**Figure IV.4: L'interface de l'application**

## Description de l'interface

On va présenter les différentes fonctionnalités de notre l'application

1. Calculer la similarité entre les profils utilisateurs et profils services
2. Trouver les utilisateurs les plus similaires avec ses services utilisés.
3. Filtrer les utilisateurs selon la fréquence de l'utilisation des services.
4. Récupérer chaque service avec ses utilisateurs.
5. Détecter tous les communautés.
6. Récupérer les dernières communautés trouvées.
7. Quitter l'application.
8. Le menu de l'application.
9. Affichage des communautés relatives à un seul service.
10. Affichage des communautés relatives à plusieurs services.
11. Affichage des communautés finales.

## IV.4 Expérimentation

Dans cette section, nous décrivons les expériences permettant de montrer le résultat de l'approche proposée. Pour mettre l'expérimentation en œuvre on a utilisé dans un première partie 20 utilisateurs et 6 services pour faciliter la compréhension, après on augmente le nombre des utilisateurs et services, l'application peut générer plusieurs fichiers de structure bien définie qui contiennent les communautés, finalement on utilise le logiciel pajek pour bien éditer le graphe biparti de communautés en utilisant des fichiers .net générés précédemment.

Par notre application produit un fichier textuel .net (figure IV.5) à partir des profils (utilisateurs/services) ces dernies sont crées avec tous leurs descriptifs aléatoirement et sont stockés dans une base de donnée dans SQL server.

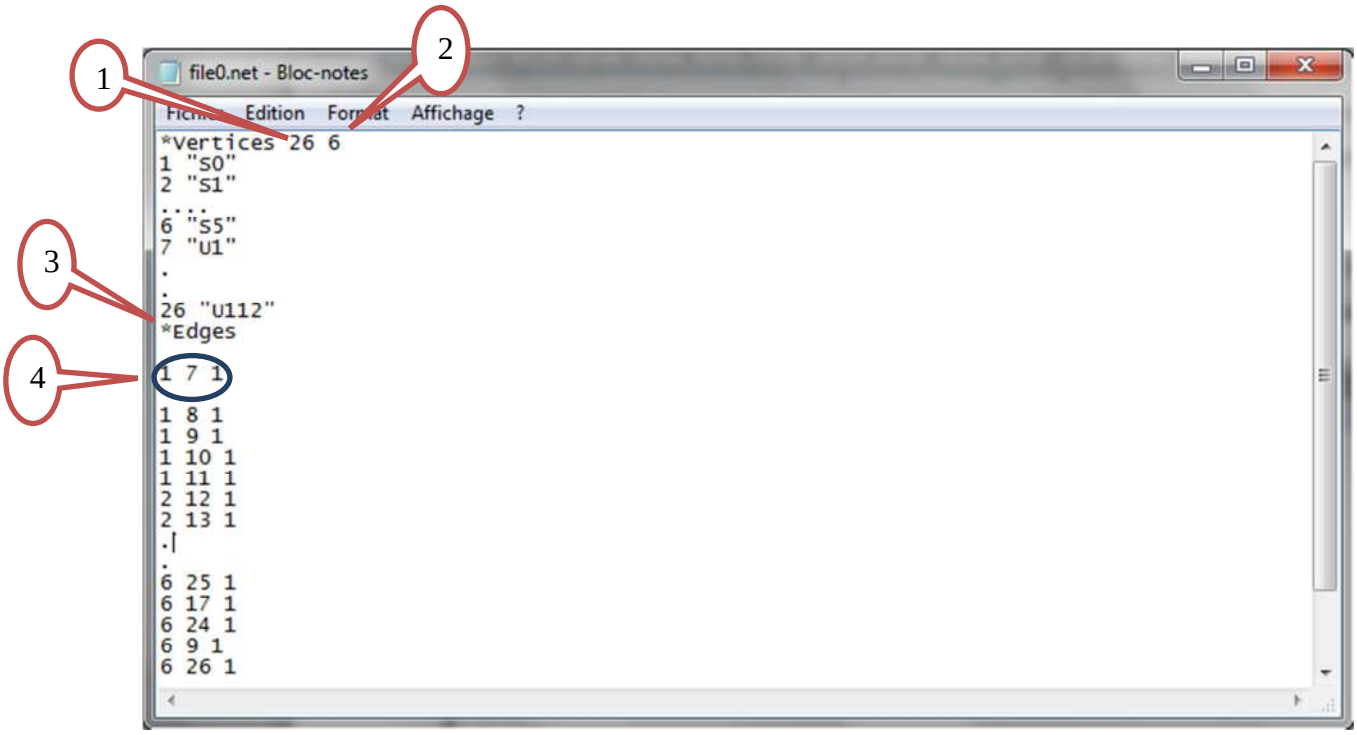



**Figure IV.5: La structure générale du fichier génère**

1. nombre totale de sommet (services + utilisateurs)
2. nombre premier sous-ensemble de sommets(les services)
3. c'est-à-dire le graphe non orienté (arêtes)
4. le service ($S_0$) plus similaire avec l'utilisateur ($U_1$)

$S_i$ : service i
$U_i$ : utilisateur i

Nous montrons dans ce qui suit les étapes suivies pour construire les différents fichiers (.net) et pour chaque étape on l'affiche dans le logiciel pajek jusqu'à atteindre le fichier qui contient les communautés finales.

**Les étapes**

- récupérer les profils (utilisateurs/services) de la base de données.
- Calculer la similarité entre ces profils (utilisateurs/services).

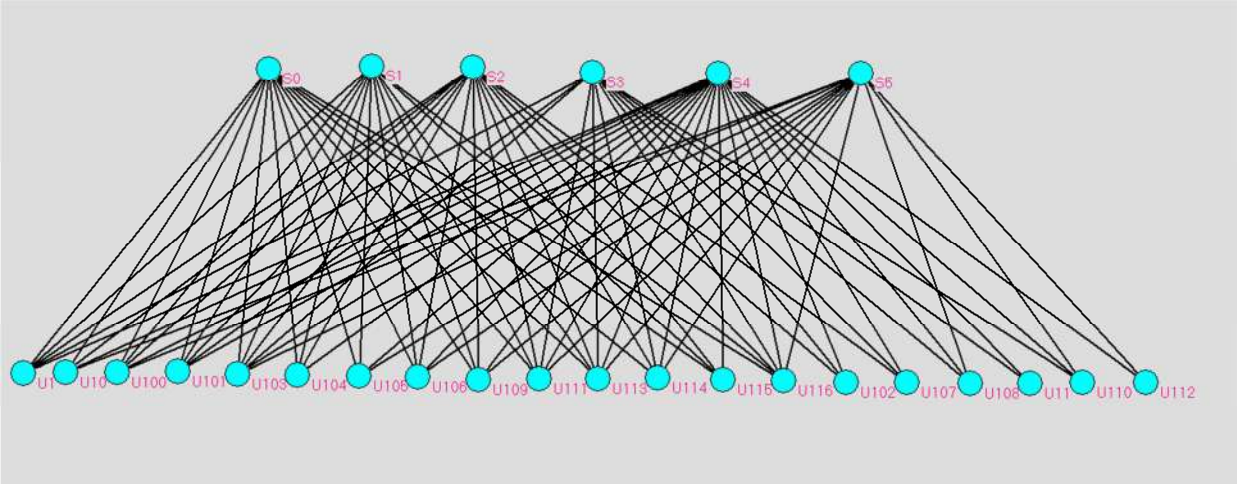



**Figure IV.6: Graphe de similarité (services/utilisateurs)**

- Filtrer le graphe initial en supprimant les liens entre utilisateurs et les services qui ont nombre d'utilisation inférieur à la limite.

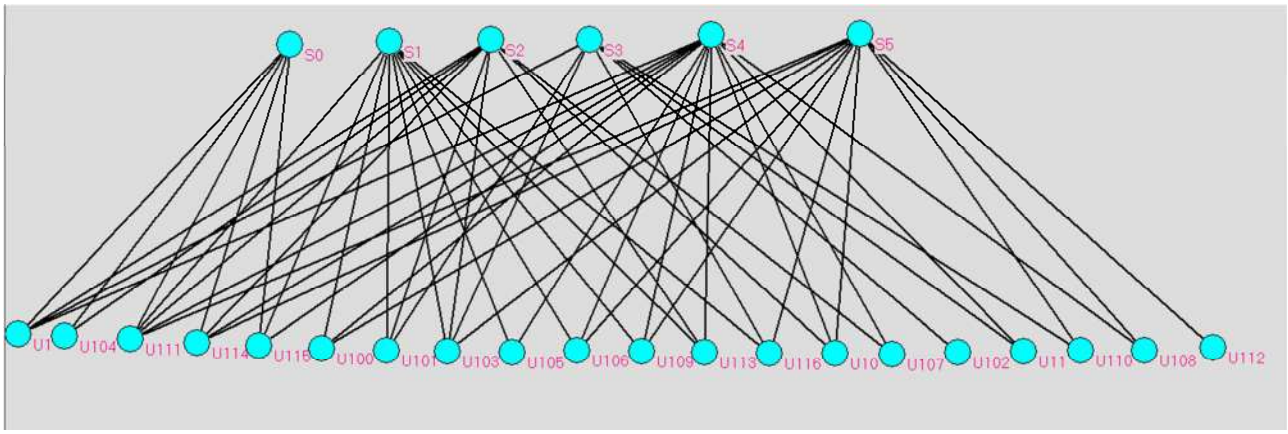



**Figure IV.7: Graphe d'accès au service au service web**

Nous constatons que certains liens présents dans le graphe des similarités n'existent plus dans le graphe d'accès. Le lien $S_0$-$U_{113}$ en est la preuve. Cela signifie que le service $S_0$ répondait aux besoins de l'utilisateur $U_{113}$, mais que ce dernier n'a pas eu accès à ce service.

- Découvrir les communautés relatives à un seul service (figure IV.8).

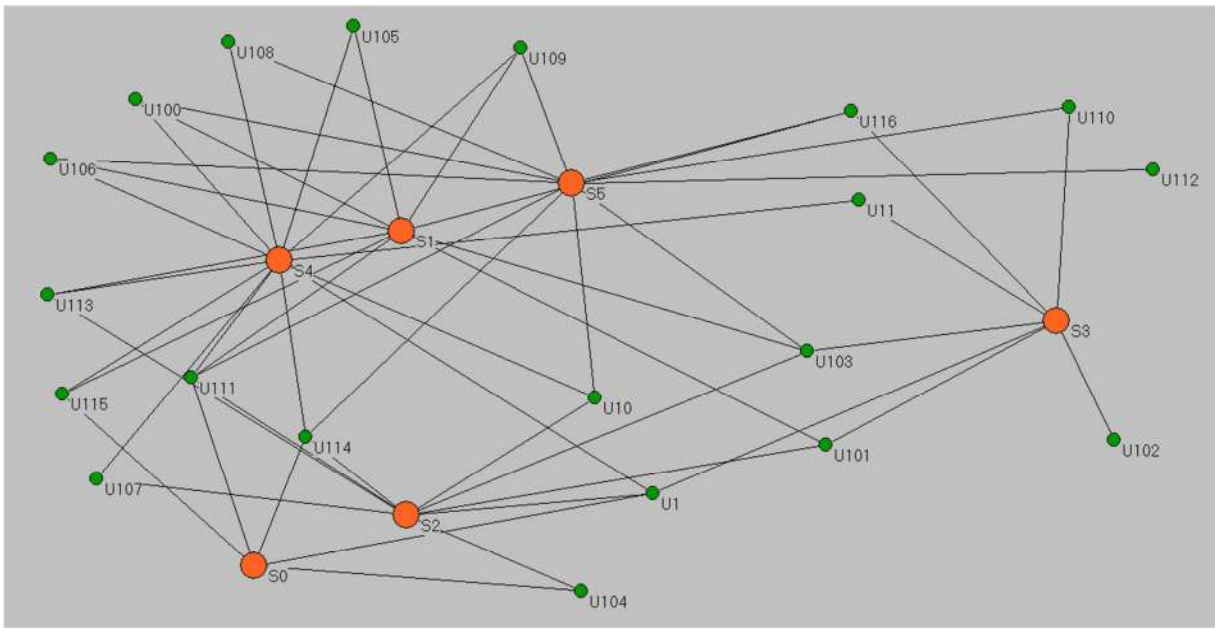



**Figure IV.8 : Découverte des communautés relatives à un service**

- Découvrir les communautés relatives à 2 services (figure IV.9).

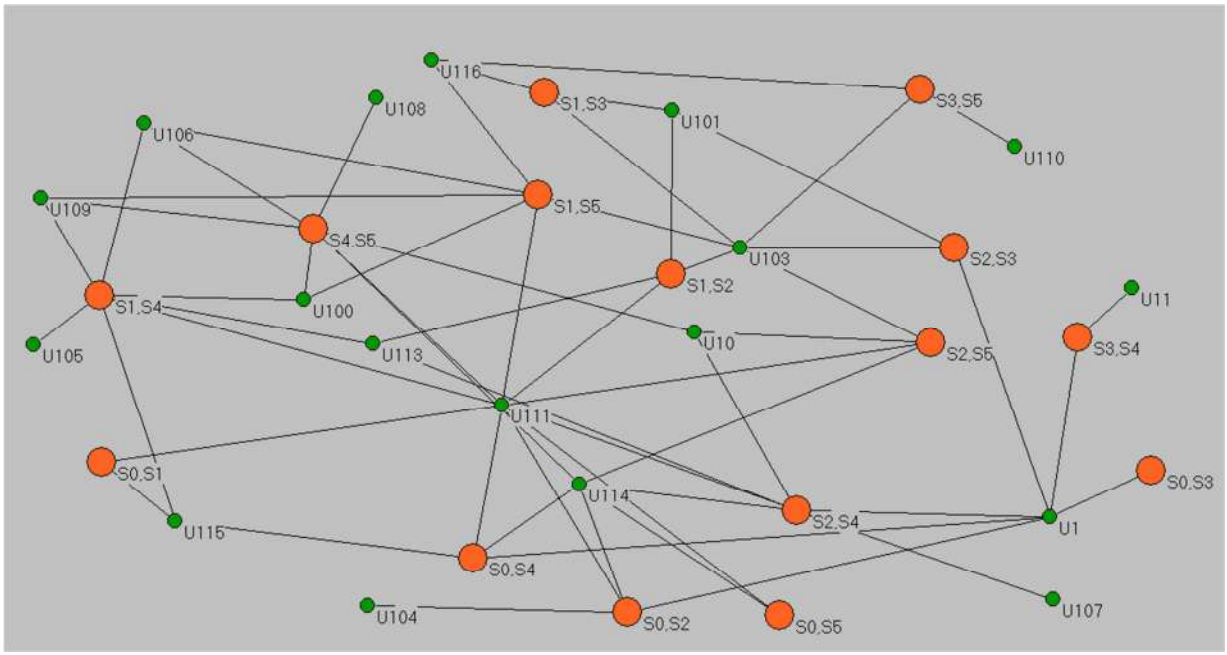



**Figure IV.9 : Découverte des communautés relatives à deux services**

De la même manière, les graphes correspondants aux communautés relatives à trois services et à N services sont générer jusqu'au que nous aurons plus d'intersections entre services et utilisateurs.

Si nous augmentons le nombre des utilisateurs à 200 et le nombre de service a 15, affiche le graphe des communautés sera comme suit (figure IV.10).

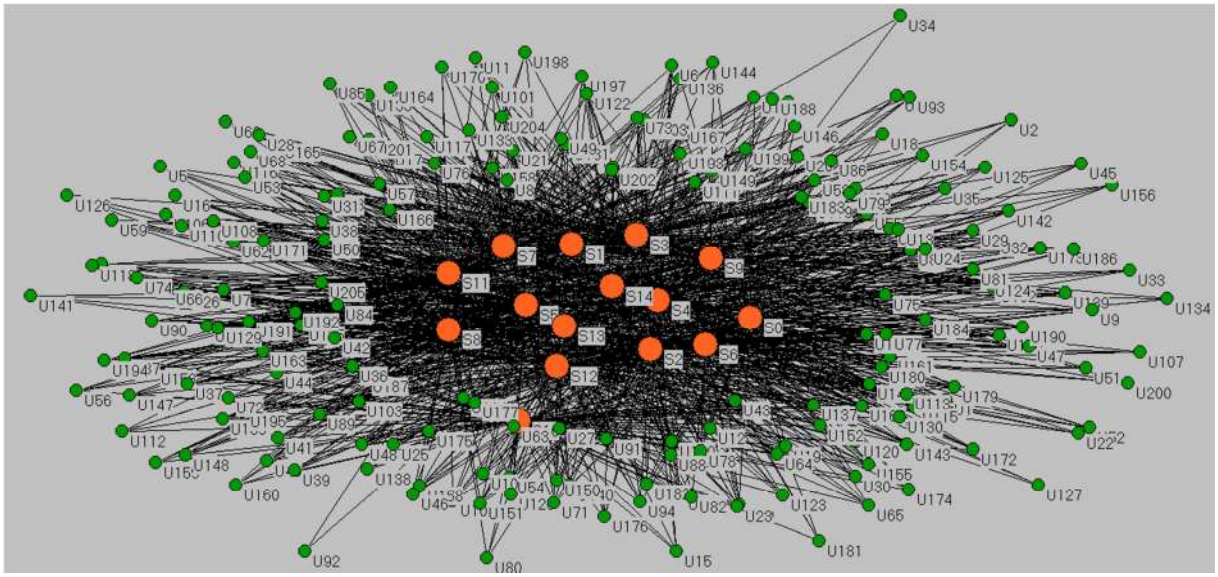


**Figure IV.10: Le nuage de communautés avec 200 utilisateurs et 15 services**

Après détection nous aurons le graphe suivant :

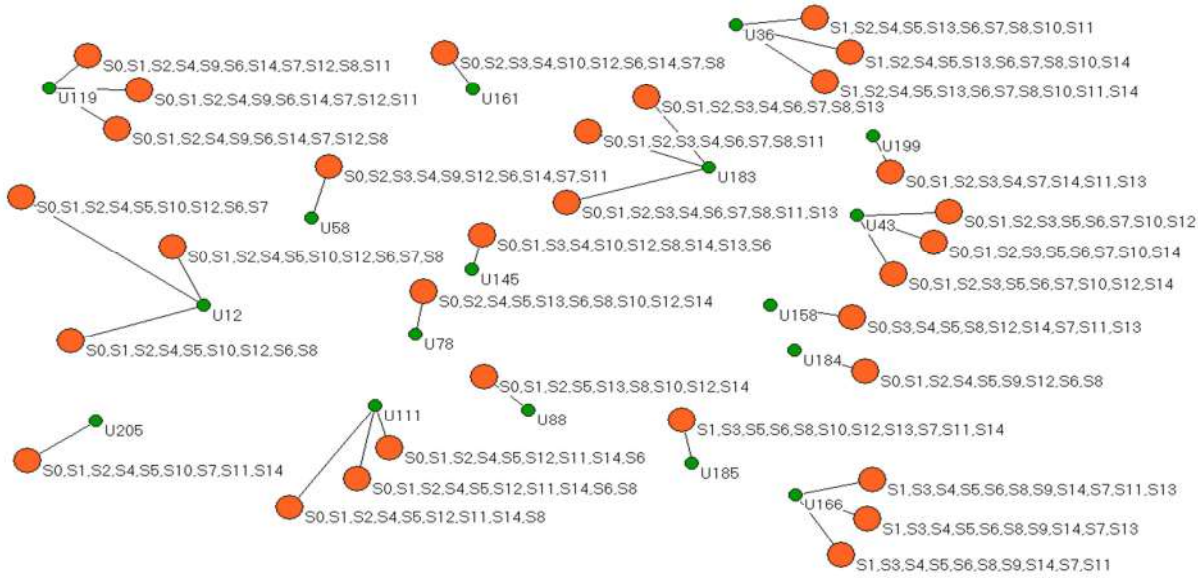



**Figure IV.11: Graphe de communautés finale**

## IV.5 Conclusion

A travers une vue pratique, cette partie est consacrée à une présentation de notre application, nous avons présenté les algorithmes nécessaires pour la mise en œuvre de la découverte des communautés dans les réseaux sociaux basé service, puis les différents outils et le langage de développement.

Enfin, il faut mentionner que la phase d'implémentation nécessite une étape d'évaluation de performances de la solution proposée, pour tester notre application, on a terminé notre chapitre par une expérimentation.

## Conclusion générale et perspectives

Dans les grands réseaux, la découverte de communautés est un problème que l'on retrouve dans plusieurs disciplines. En particulier, dans les réseaux sociaux, ces communautés jouent un rôle important dans l'organisation et la structuration du réseau. Jusqu'à présent, l'objectif des méthodes de détection de communautés était de former des communautés d'utilisateurs fortement connecté entre eux.

Mais à l'heure du web 2 .0, les utilisateurs échangent de plus d'informations et lissent constamment des traces de leurs activités en ligne. De ce fait, le web est devenu une véritable mine d'information dont l'exploitation peut s'avérer très utile. Notre intérêt s'est porté sur le traitement et l'analyse des informations concernant les accès des utilisateurs aux services web. En effet, les services web auxquels ont accès les utilisateurs sont souvent révélateurs de leur centre d'intérêts et de leurs activités. Ces informations peuvent par exemple servir à obtenir une meilleure idée des services qui les intéressent, ce qui peut orienter les activités d'une organisation et améliorer une stratégie globale de marketing. Elles peuvent également servir à dresser un portrait de l'ensemble des utilisateurs d'un service à des fins de mise en marché et de personnalisation des services offerts. Pour toutes ces raisons nous avons jugé pertinent de regrouper en communauté des utilisateurs utilisant les mêmes services web. Nous avons proposé dans ce sens une méthode de détection de communautés basée sur les interactions des utilisateurs avec les services web. Notre méthode de détection utilise une mesure de similarité qui se base sur l'interaction des utilisateurs a travers les services web. L'objectif était de détecter les communautés relatives à un seul service, puis d'affiner la découverte en considération les communautés relatives à toutes les combinaisons de service qui existent.

Toute fois notre travail reste susceptible à une amélioration certaine en y intégrant une détection de communauté multicritère, telle que la variation selon le temps et la localisation de chaque utilisateur et service, dans le but d'améliorer la détection de communauté. Comme il travail ouvre de nombreuses perspectives au sein de ce domaine à savoir l'utilisation de benchmark des réseaux sociaux tels que Facebook, LinkedIn. . . .etc, ou bien l'ajout d'autres contraintes telles que la tranche d'âge, niveau d'étude. . . .etc.

Enfin, nous espérons que ce modeste travail fera l'objet d'une suite afin d'améliorer et compléter notre idée.

## Bibliographies

[1]: Barry Douglas K. The Savvy Manager's Guide to Web Services and Service Oriented Architectures. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[2] : WaelSELLAMI. Une approche formelle pour la vérification des propriétés non fonctionnelles d'orchestration des services web. 2011.

[3] : Olivier Carton, L'essentiel de XML - Cours XML. 2014.

[4]: baronm. 02 - Web Service – WSDL. 2012

[5]: M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, and H.F. Nielsen. Simple object access protocol (soap) 1 . 2. http ://www.w3.org/TR/soap, 2004.

[6] : Karim Bouchahma, Conception et développement d'un Service Web Pour l'échange d'information dans le domaine humanitaire, 2007.

[7] : Mémoire de fin d'études présenté pour l'obtention du grade de master «Les réseaux sociaux, un levier de fidélité pour les entreprises. Exemple concret de l'enseigne Décathlon » Présenté par « Aurélien HILT » Juin 2013

[8] : Philipe Torloting Enjeux et perspectives des réseaux sociaux, 2006.Mémoire.

[9]: M. E. J. Newman. The structure and function of complex networks.

Mar 2003.

[10] : Ecole d'été web intelligence. In *WI09*. Université de Lyon, 2009.

[11]: G. Ereteo, F Gando, M. Buffa, and P. Grohan. Analyse des réseaux sociaux et web sémantique : un état de l'art. Technical report, Projet ISICIL (ANR), 2009.

[12]: S. P. Borgatti, M. G. Everett, and L. C. Freeman. *Ucinet for Windows : Software for Social Network Analysis*, 2002.

[13]: M. G. Everett and S. P. Borgatti. The centrality of groups and classes. *Journal of Mathematical Sociology*, 23(3) :181–201, 1999.

[14]: S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge : Cambridge University Press, 1994.

[15]: M. E. J. Newman, “Detecting community structure in networks,” *The European Physical Journal B Condensed Matter*, vol. 38, no. 2, pp. 321-330, 2004.

[16]: A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms.,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 78, no. 4 Pt 2, p. 6, 2008.

[17]: G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, “Self-organization and identification of Web communities,” *Computer*, vol. 35, no. 3, pp. 66-71, 2002.

[18] : Thèse présentée pour l’obtention du grade de Docteur EN INFORMATIQUE Soutenue publiquement par « CHANTAL CHERIFI » Le 9 décembre 2011 « Classification et Composition de Services Web : Une Perspective Réseaux Complexes»

[19]: V. Satuluri and S. Parthasarathy, “Scalable graph clustering using stochastic flows: applications to community discovery,” in *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 737-746.

[20]: V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, Oct. 2008.

[21] : M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821-6, Jun. 2002.

[22]: A. Lancichinetti, M. Kivelä, J. Saramäki, and S. Fortunato, “Characterizing the Community Structure of Complex Networks,” *PLoS ONE*, vol. 5, no. 8, p. 8, 2010.

[23]: F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, “Defining and

identifying communities in networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658-2663, 2004.

[24]: S. Van Dongen, “Graph Clustering Via a Discrete Uncoupling Process,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 121-141, 2008.

[25]: P. Pons and M. Latapy, “Computing communities in large networks using random walks,” *Journal of Graph Algorithms and Applications*, vol. 10, no. 2, pp. 191-218, 2005.

[26]: M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118-1123, 2008.

[27]: M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 74, no. 3 Pt 2, p. 036104, 2006.

[28]: L. Donetti and M. A. Munoz, “Detecting network communities: a new systematic and efficient algorithm,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2004, no. 10, p. P10012, 2004.

[29]: Faycal M’hamed Bouyakoub; Abdelkader Belkhir, A Service Discovery Approach Based on a Quantitative Similarity Measure for M-Tourism Platforms, 2013.

[30] : C# et .NET Auteur :Gerarg LEBLANC  Date :2006 ÉDITIONS EYROLLES 61, bd Saint-Germain 75240 Paris Cedex 05.

[31] : jravaille, Introduction à la B.I. avec SQL Server 2008, 2009.

[32] : Laurent Beauguitte, Initiation à Pajek Logiciel pour l'analyse des réseaux sociaux, 2014.

## Annexe : Conversation numérique des profils

| Caractéristique | Interprétation Commune User Publisher | Interprétation coté User | Interprétation Coté Publis-her | Valeurs Quantifiée utilisateur | Valeurs quantifiée par celui qui a publié |
|---|---|---|---|---|---|
| Capacité | Faible : valeur $\in [0,40[$<br>Moyenne : valeur $\in [40,70[$<br>Elevée : valeur $\in [70,100[$ | Faible<br>Moyenne<br>Elevée | La valeur réelle de la capacité du service | La bonne supérieure des inter-valles<br>40<br>70<br>100 | Valeur saisis par celui qui a publié |
| Performance | Faible : valeur $\in [115, plus[$<br>Moyenne : valeur $\in [7,15[$<br>Elevée : valeur $\in [2,7[$ | Faible<br>Moyenne<br>Elevée | La valeur réelle de la vitesse d'exécution du service | La bonne inferieure des inter-valles<br>2<br>7<br>15 | Valeur saisis par celui qui a publié |
| Fiabilité | Faible : valeur c'est 2<br>Non Faible : valeur c'est 1 | Faible<br>Non faible | Faible<br>Non faible | 2<br>1 | 2<br>1 |
| Disponibilité | Disponible : valeur c'est 2<br>Non Dispo-nible : valeur c'est 1 | Disponible<br>Non Disponible | Disponible<br>Non Disponible | 2<br>1 | 2<br>1 |
| Robustesse/<br>Flexibilité | Flexible : va-leur c'est 2<br>Non Flexible : va-leur c'est 1 | Flexible<br>Non flexible | Flexible<br>Non flexible | 2<br>1 | 2<br>1 |
| Sécurité | Faible : valeur c'est 3<br>Moyenne : valeur c'est 2<br>Elevée : valeur c'est 1 | Faible<br>Moyenne<br>Elevée | Faible<br>Moyenne<br>Elevée | 3<br>2<br>1 | 3<br>2<br>1 |

| Caractéristique | Interprétation Commune User Publis-her | Interprétation coté User | Interprétation Coté Publis-her | Valeurs Quantifiée utilisateur | Valeurs quantifiée par celui qui a publié |
|---|---|---|---|---|---|
| Langue | On a pris en considération 80 langues. Et on a pris leurs classement de la+ parlée à la - parlée | la plus parlée: valeur c'est la valeur +grande. Moins parlée Valeur c'est la valeur la + petite. | la plus parlée: valeur c'est la valeur +grande. Moins parlée Valeur c'est la valeur la + petite. | La valeur Est prise dans inter-valle [1,80] selon la langue saisis | La valeur Est prise dans inter-valle [1,80] selon la langue saisis |
| Cout | On a pris trois unités monétaires. Pour la compa-raison on fait la conversion vers le dinar | Le cout en dinar | Le cout en dinar | Valeur sai-sis par l'utilisateur convertis au dinar | Valeur saisis par celui qui a publié, convertis au dinar |
| Modèle | La capacité de mémoire gra-phique | La capacité de mémoire graphique par modèle de dispositif | La capacité de mémoire graphique par modèle de dispositif | Valeur nu-mérique | Valeur nu-mérique |
| Couleur | Couleur : va-leur c'est 2 noir &blanc : valeur c'est 1 | couleur noir &blanc | couleur noir &blanc | 2 1 | 2 1 |
| Type Software | Classement selon la (qualité prix) Qualité : (faci-lité d'installation, performances, confort d'utilisation) | Classement selon la (qualité prix) Qualité : (fa-cilité d'installation, performances, confort d'utilisation) | Classement selon la (qualité prix) Qualité : (fa-cilité d'installation, performances, confort d'utilisation) | La valeur est prise dans cet intervalle [12,20] se-lon l'OS saisis | La valeur est prise dans cet intervalle [12,20] selon l'OS saisis |
| Navigateur | Classement selon le temps de démarrage, Utilisation mémoire. temps charge-ment des pages | Classement selon le temps de dé-marrage, Utilisation mémoire. temps char-gement des pages | Classement selon le temps de dé-marrage, Utilisation mémoire. temps char-gement des pages | La valeur est prise dans cet intervalle [1..6] selon le naviga-teur saisis | La valeur est prise dans cet intervalle [1..6] selon le navigateur saisis |

| Caractéristique | Interprétation Commune User Publisher | Interprétation coté User | Interprétation Coté Publisher | Valeurs Quantifiée utilisateur | Valeurs quantifiée celui qui a publié |
|---|---|---|---|---|---|
| Pays | Selon les valeurs de la longitude et latitude de la paye donnée par le GPS | Selon les valeurs de la longitude et latitude de la paye donnée par le GPS | Selon les valeurs de la longitude et latitude de la paye donnée par le GPS | Valeur numérique | Valeur numérique |
| Région | Selon les valeurs de la longitude et latitude de la région donnée par le GPS | Selon les valeurs de la longitude et latitude de la région donnée par le GPS | Selon les valeurs de la longitude et latitude de la région donnée par le GPS | Valeur numérique | Valeur numérique |
| Ville | Selon les valeurs de la longitude et latitude de la ville donnée par le GPS | Selon les valeurs de la longitude et latitude de la ville donnée par le GPS | Selon les valeurs de la longitude et latitude de la ville donnée par le GPS | Valeur numérique | Valeur numérique |