Classification et Composition de Services Web : Une Perspective Réseaux Complexes

Chantal Cherifi

UNIVERSITE DE CORSE-PASCAL PAOLI
ECOLE DOCTORALE ENVIRONNEMENT ET SOCIETE
UMR CNRS 6134 (SPE)

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

---

Sous la direction du Professeur

Jean-François Santucci

Rapporteurs : M. Nacer BOUDJLIDA, *Professeur, Université Henri Poincaré, Nancy 1*
M. Hamamache KHEDDOUCIi, *Professeur, Université Claude Bernard, Lyon1*

Examinateurs : M. Jean-François SANTUCCI, *Professeur, Université de Corse, Corte*
Mme. Marie-Christine FAUVET, *Professeur, Université Joseph Fourier, Grenoble*
M. Dominique FEDERICI, *Professeur, Université de Corse, Corte*
Mme. Lynda TAMINE, *MCF-HDR, Université Paul Sabatier, Toulouse*
Mme. Evelyne VITTORI, MCF, *Université de Corse, Corte*

Remerciements

C’est avec beaucoup d’émotion que je rédige cette page parce qu’elle symbolise un lieu de rassemblement. Dans cet espace se trouvent réunies les personnes qui, à travers l’espace et le temps, ont partagé cette aventure.

En premier lieu, j’adresse toute ma gratitude à M. Jean-François Santucci qui a dirigé ces travaux. Grâce à sa disponibilité, sa réactivité et son efficacité, c’est en toute confiance et en toute quiétude que j’ai pu mener à bien ce projet. J’ai pleinement bénéficié de son soutien tout au long du parcours, et je l’en remercie vivement. Qu’il trouve ici l’expression de mon profond respect.

M. Nacer Boudjlida m’a fait l’honneur d’accepter de rapporter sur cette thèse. Ses travaux sur la représentation de services Web sémantiques sous forme de réseau ont été une source d’inspiration. Je lui suis reconnaissante d’avoir examiné mon travail avec la plus grande attention et lui adresse mes vifs remerciements.

J’adresse également de chaleureux remerciements à M. Hamamache Kheddouci. La classification des services Web à partir des réseaux d’interaction que je propose est en partie inspirée de ses travaux en la matière. J’ai eu l’occasion de profiter de ses conseils avisés et je suis très honorée qu’il accepte d'évaluer mon travail de thèse.

M^{me} Marie-Christine Fauvet a accepté de prendre part au jury. J’en suis flattée et l’en remercie vivement. L’attention qu’elle porte à mes travaux revêt une importance toute particulière et me sera d’une grande utilité pour enrichir ma réflexion.

La participation au jury de M^{me} Evelyne Vittori prolonge des échanges très fructueux. Son avis, ses conseils, son accompagnement, ses encouragements et son amitié m’ont beaucoup aidée, et ce tout spécialement pendant la période de rédaction. Merci de tout cœur.

J’adresse de même mes profonds remerciements à M^{me} Lynda Tamine pour avoir accepté de participer au jury de thèse. L’intérêt enthousiaste qu’elle a montré pour mes travaux me touche tout particulièrement.

M. Dominique Federici a également accepté d’examiner ce travail de thèse. Je suis très sensible à l’accueil qu’il a réservé à cette demande. Je lui en suis extrêmement reconnaissante et lui exprime toute ma gratitude.

Mes remerciements vont également à M^{me} et M. Jacqueline et Jean-Jacques Chabrier pour leur soutien constant.

Je remercie chaleureusement M^{me} Marie-Christine Rousset et M. Jean-Claude Fernandez dont j’ai également pu profiter des conseils précieux pour l’orientation de mes recherches.

Merci à MM. Mürat Egi de l’Université Galatasaray à Istanbul et Bih-Yaw Shih de NPUE à Taïwan, pour m’avoir accueillie au sein de leur laboratoire pendant mes séjours dans leurs établissements.

Je tiens à remercier vivement M. Marcel Grenard pour son accueil à l’IUT de Dijon et toute l’attention qu’il réserve en particulier aux nouveaux arrivants.

Mes chaleureux remerciements vont également à M. Alexandre Guidet. Il a toujours su être à l'écoute pour l'aménagement de mes enseignements au département informatique.

Merci à M. Vincent Labatut avec qui j'ai eu un immense plaisir à travailler pendant ces années de thèse, et avec qui les échanges ont été très constructifs.

Merci à Günce, Yvan, Nadin, Koray et Cihan qui ont pris part, à un moment, à ce travail de thèse et avec qui les collaborations ont été très fructueuses.

Une pensée émue pour Sylvain, mon fidèle compagnon de route du début et un clin d'œil à Sültan. Que de bons moments passés ensemble à GSU !

A tous mes amis relecteurs, Agnès, Alain, Jacqueline, Jean-Jacques, Juliette, Mady, Marie, Thierry, Sylvie et Vanessa, qui ont abordé la tâche avec grand sérieux, et pour chacun, avec l'humour que je leur connais, merci pour leur temps, merci de m'avoir fait rire ;)

Enfin, à mon mari, à mes parents, à ma famille et à tous les amis de qui je n'ai cessé de recevoir le patient soutien et les persévérants encouragements, merci pour leur amour :)

Résumé

Les services Web sont des briques de bases logicielles s'affranchissant de toute contrainte de compatibilité logicielle ou matérielle. Ils sont mis en œuvre dans une architecture orientée service. A l'heure actuelle, les travaux de recherche se concentrent principalement sur la découverte et la composition. Cependant, la complexité de la structure de l'espace des services Web et son évolution doivent nécessairement être prises en compte. Ceci ne peut se concevoir sans faire appel à la science des systèmes complexes, et notamment à la théorie des réseaux complexes. Dans cette thèse, nous définissons un ensemble de réseaux pour la composition sur la base de services décrits dans des langages syntaxique (WSDL) et sémantique (SAWSDL). L'exploration expérimentale de ces réseaux permet de mettre en évidence les propriétés caractéristiques des grands graphes de terrain (la propriété petit monde et la distribution sans échelle). On montre par ailleurs que ces réseaux possèdent une structure communautaire. Ce résultat permet d'apporter une réponse alternative à la problématique de la classification de services selon les domaines d'intérêts. En effet, les communautés regroupent non pas des services aux fonctionnalités similaires, mais des services qui ont en commun de nombreuses relations d'interaction. Cette organisation peut être utilisée entre autres, afin de guider les algorithmes de recherche de compositions. De plus, en ce qui concerne la classification des services aux fonctionnalités similaires en vue de la découverte ou de la substitution, nous proposons un ensemble de modèles de réseaux pour les représentations syntaxique et sémantique des services, traduisant divers degrés de similitude. L'analyse topologique de ces réseaux fait apparaître une structuration en composantes et une organisation interne des composantes autour de motifs élémentaires. Cette propriété permet une caractérisation à deux niveaux de la notion de communauté de services similaires, mettant ainsi en avant la souplesse de ce nouveau modèle d'organisation. Ces travaux ouvrent de nouvelles perspectives dans les problématiques de l'architecture orientée service.

Abstract

Web services are building blocks for modular applications independent of any software or hardware platforms. They implement the service oriented architecture (SOA). Research on Web services mainly focuses on discovery and composition. However, complexity of the Web services space structure and its development must necessarily be taken into account. This cannot be done without using the complex systems science, including the theory of complex networks. In this thesis, we define a set of networks based on Web services composition when Web services are syntactically (WSDL) and semantically (SAWSDL) described. The experimental exploration of these networks can reveal characteristic properties of complex networks (small world property and scale-free distribution). It also shows that these networks have a community structure. This result provides an alternative answer to the problem of Web services classification by domain of interest. Indeed, communities don't gather Web services with similar functionalities, but Web services that share many interaction relationships. This organization can be used among others, to guide compositions search algorithms. Furthermore, with respect to the classification based on Web services functional similarity for discovery or substitution, we propose a set of network models for syntactic and semantic representations of Web services, reflecting various similarity degrees. The topological analysis of these networks reveals a component structure and internal organization of the components around elementary patterns. This property allows a two-level characterization of the notion of community of similar Web services that highlight the flexibility of this new organizational model. This work opens new perspectives in the issues of service-oriented architecture.

Table des matières

INTRODUCTION

Les services Web sont des composants logiciels distribués qui exposent un ensemble de fonctionnalités sur un réseau [1]. Leur mise en œuvre repose sur une architecture décentralisée. Cette architecture orientée services (Service Oriented Architecture - SOA) [2] est un style architectural fondé sur la description des services et de leurs interactions. Les services sont publiés dans des annuaires par des fournisseurs qui les développent et les hébergent. Ils sont accessibles via un réseau pour les clients qui les découvrent, les sélectionnent, les invoquent et les utilisent. Ces fonctionnalités distribuées que constituent les services Web sont conçues pour être utilisées conjointement dans des compositions. La composition de services fait référence au processus qui consiste à combiner les fonctionnalités de plusieurs services, simples ou eux-mêmes composés, au sein d'un même processus métier, dans le but de répondre à des demandes complexes qu'un seul service ne pourrait satisfaire [3]. La composition au sens large englobe plusieurs activités qui correspondent à différentes phases de son cycle de vie [4], depuis la description jusqu'à l'exécution. D'une façon générale, on peut dire que le cycle de vie de la composition de services se compose des activités de publication, de découverte, de synthèse de la composition, d'orchestration, de contrôle et de surveillance de la composition pendant son exécution. La publication englobe deux opérations élémentaires qui sont la description des services et l'inscription aux registres. La découverte consiste à trouver dans les annuaires le ou les services pouvant répondre à un besoin. L'activité de découverte peut s'accompagner d'une phase de sélection pendant laquelle un utilisateur ou un agent se trouve devant un choix à opérer entre plusieurs services. La synthèse de la composition aboutit à une spécification de la façon dont sont coordonnés les services afin de répondre à un besoin, c'est-à-dire l'ordre et les conditions dans lesquels ils s'enchaînent. L'orchestration réalise concrètement la synthèse en invoquant effectivement les services participants, en les exécutant, en supervisant et en gérant l'exécution de la composition. Pendant cette étape d'exécution, la substitution peut intervenir pour remplacer un service par un autre en cas de défaillance.

La description des services est une étape essentielle pour la découverte et la composition. Les informations qu'elle contient permettent de guider la recherche et le choix des services et conditionnent leur interaction. Une description précise la fonction d'un service, ses contraintes de fonctionnement et la façon d'interagir avec lui. Schématiquement, on peut classer ces informations en quatre catégories : les informations générales relatives à la nature du service (nom, description), les informations techniques relatives à la façon de l'utiliser, les informations non fonctionnelles qui concernent la qualité de service et enfin les informations fonctionnelles qui décrivent les fonctionnalités du service.

La description des services Web repose actuellement sur le standard Web Service Description Language (WSDL) [5] qui fournit une description syntaxique des services. Il spécifie les fonctionnalités d'un service en définissant des messages et des opérations. Les messages fournissent une définition abstraite des données qui doivent être échangées. Les opérations sont fournies par les services pour transformer les messages. Chaque message contient un ou plusieurs paramètres. WSDL permet une description centrée sur la fonction du service qui est représentée par les paramètres en entrée et en sortie des opérations.

Une deuxième façon de décrire les services est de faire appel aux ontologies [6] pour les décrire sémantiquement. L'objectif de la description sémantique est d'automatiser les activités du cycle de vie des services Web. Au niveau fonctionnel, les services Web sémantiques sont décrits de la même façon que les services syntaxiques, c'est-à-dire avec des opérations et des paramètres d'entrée et de sortie. La différence majeure est que les paramètres sont dans ce cas associés à des concepts ontologiques. Citons un des principaux langages qui est Ontology Web Language for Services (OWL-S) [7].

Une approche médiane consiste à enrichir les descriptions syntaxiques des services en utilisant des ontologies. Les deux principales propositions en ce sens sont WSDL-Semantic (WSDL-S) [8] et Semantic Annotation for WSDL (SAWSDL) [9]. Ces spécifications W3C établissent une correspondance entre certains éléments WSDL existants et des concepts ontologiques. Les opérations et les messages peuvent être annotés dans WSDL-S. SAWSDL prévoit une annotation seulement au niveau des paramètres. Le but est d'aboutir à une annotation automatique des descriptions WSDL existantes.

D'une façon générale, le passage d'une description syntaxique à une description sémantique des données est un élément de réponse à l'immense succès du Web. Initialement conçu pour le partage d'information entre scientifiques, le Web est devenu aujourd'hui un vaste et riche espace de source d'information. Cette information est caractérisée par son dynamisme et son aspect non structuré. Afin d'accompagner efficacement la croissance exponentielle des échanges, il est impératif d'utiliser un langage universel compréhensible tant par les humains que par les machines. Les services Web, en tant qu'applications disponibles sur le Web, suivent cette même tendance. De plus en plus d'organisations et d'entreprises externalisent leurs applications sous la forme de services Web. A titre d'exemple, Google renvoie 42 000 résultats pour une recherche de fichiers WSDL, ce qui est certainement loin du nombre réel de services existants.

Le développement des services Web soulève ainsi des difficultés similaires à celles qui ont accompagné la croissance du Web. En effet, ces services sont développés par différentes entités et il n'existe pas de consensus sur la manière de faire usage des descriptions de services. Les services sont créés, mis à jour ou supprimés à la volée ; ces changements peuvent, de plus, ne pas être reportés dans les annuaires. Pour de multiples raisons, les services peuvent également présenter des problèmes de dysfonctionnement au moment de leur utilisation. Ces caractéristiques conduisent à un environnement extrêmement dynamique et volatile. Publier, découvrir et composer des services dans un tel environnement pose un certain nombre de problèmes. La recherche de services doit se faire dans un ensemble de grande taille non structuré et changeant. La gestion de la dynamique doit être prise en compte au cours des processus de synthèse et d'exécution de la composition.
Toutes ces problématiques s'apparentent ainsi à celles rencontrées dans un grand nombre de systèmes du monde réel caractérisés par de grands ensembles d'entités en interaction. Une des caractéristiques communes à tous ces systèmes est qu'ils peuvent être représentés par des réseaux, c'est-à-dire des nœuds figurant des entités reliées par des liens qui représentent les interactions. L'étude de ces réseaux de grande taille issus d'une évolution décentralisée et non planifiée est aujourd'hui une discipline scientifique à part entière. Emergeant de la physique

statistique qui possède une longue tradition d'étude des systèmes complexes, la science des réseaux s'étend à des domaines scientifiques aussi variés que la biologie, l'informatique, la sociologie et les grandes infrastructures industrielles (transports, énergie, etc.). Ces réseaux qualifiés de grands graphes de terrain ou réseaux complexes, possèdent pour la plupart un certain nombre de propriétés topologiques caractéristiques. Leur caractère auto-organisé et évolutif les distingue aussi bien d'une organisation régulière que d'une organisation purement aléatoire. Ils sont notamment caractérisés par la propriété « petit monde ». Celle-ci traduit le fait qu'il y a souvent une liaison courte entre deux entités du réseau. Ces caractéristiques communes permettent ainsi de traiter ces réseaux avec des outils spécifiques. Les avancées en la matière ont des applications potentielles dans les nombreux domaines concernés. A titre d'exemple, on peut citer la propagation des virus dans les réseaux informatiques, la propagation d'épidémies dans les réseaux sociaux, la résistance aux pannes dans les grandes infrastructures technologiques ou encore les perturbations de l'écosystème, comme autant de champs d'application de l'étude des réseaux complexes.

Si l'on considère le domaine informatique, le Web et Internet constituent les deux exemples les plus représentatifs de grands graphes de terrain. Internet est un réseau d'infrastructure où les nœuds représentent les « machines » (routeurs, système autonomes, ordinateurs) et les liens représentent les liaisons physiques entres elles. Le Web est quant à lui un système d'information où les nœuds représentent les pages Web et les liens correspondent aux liens hypertextes entre les pages. Ces deux types de réseaux complexes ont fait l'objet de nombreuses études. En ce qui concerne les services Web, le nombre de travaux est extrêmement limité. Pourtant, à l'instar du Web, l'espace des services Web constitue un système d'information à partir duquel on peut concevoir différents modèles de graphes pour représenter les services et les relations qu'ils entretiennent. Tout comme pour le Web, il est nécessaire à terme de tenir compte du facteur d'échelle et le paradigme des réseaux complexes s'avère être adapté dans ces situations.

Nos travaux s'inscrivent dans ce contexte. Ils visent à définir et à valider des modèles de réseaux complexes répondant aux problématiques soulevées par les services Web. En effet, nous pensons que la structuration de l'espace des services Web peut être abordée dans ce cadre. Comme dans les grands graphes de terrain, l'espace des services Web est un système où de nombreuses entités entretiennent des relations plus ou moins complexes. Il présente un caractère fortement dynamique qui tient au fait que des services sont continuellement ajoutés, supprimés, relocalisés ou mis à jour. Ces entités peuvent entretenir différents types de relations. Ainsi, dans le cadre de la composition, les services interagissent afin de satisfaire les buts d'un utilisateur. Cette problématique peut être abordée à travers des réseaux d'interaction. Si l'on se place dans la perspective de la découverte et de la substitution, les relations entre services qui sont à considérer sont des relations de similitude. Certains services peuvent par exemple avoir les mêmes fonctionnalités, appartenir au même domaine d'application, avoir les mêmes exigences de qualité de service. On peut là aussi définir des réseaux traduisant cette notion de similitude.

Dans ces travaux, nous nous intéressons essentiellement à l'aspect fonctionnel des services Web. Un service est alors défini comme un ensemble d'opérations qui possèdent des paramètres en entrée et en sortie. Ces paramètres peuvent être décrits de façon syntaxique ou

de façon sémantique. Nous nous intéressons plus particulièrement à la notion de relation d'interaction intervenant dans le cadre de la composition et à celle de similitude caractéristique de la classification des services.

Rappelons que la classification est l'opération qui consiste à organiser les services Web publiés dans les annuaires. Elle a pour objectif d'améliorer les activités de publication, de découverte, d'invocation ou encore de maintenance. C'est une tâche qui s'avère complexe et coûteuse si elle est réalisée manuellement, étant donné le nombre important et croissant de services. Les travaux sur la classification visent à proposer des solutions pour l'automatisation de ce processus. La plupart des méthodes de classification proposées sont basées sur la similitude entre les services. Pour notre part, la notion de similitude porte sur les opérations contenues dans les services et plus particulièrement sur les paramètres d'entrée/sortie de ces opérations.

Dans le cycle de vie de la composition, nous focalisons notre attention sur l'aspect modélisation de la synthèse. Une composition est ainsi formée d'un enchaînement de services reliés par leurs paramètres d'entrée/sortie. A titre d'exemple, supposons qu'un utilisateur souhaite obtenir la date de publication d'un livre (figure 1). Cet utilisateur connaît le nom de l'auteur et le titre du livre. Si sa requête ne peut être satisfaite par un service Web atomique, on peut envisager une combinaison d'autres services. Supposons que les services suivants soient disponibles figure 1(b): le premier, `AuthorNameBookTitle_ISBN`, fournit l'ISBN d'un livre contre le nom de l'auteur et le titre du livre; le second, `ISBN_PubliDate`, fournit la date de publication d'un livre contre son ISBN. Un service composite satisfaisant la requête peut être synthétisé. Il est obtenu en combinant les services `AuthorNameBookTitle_ISBN` et `ISBN_PubliDate` comme indiqué dans la figure 1(c). La composition ainsi formée fournit l'information attendue par l'utilisateur, c'est-à-dire la date de publication du livre figure 1(a).

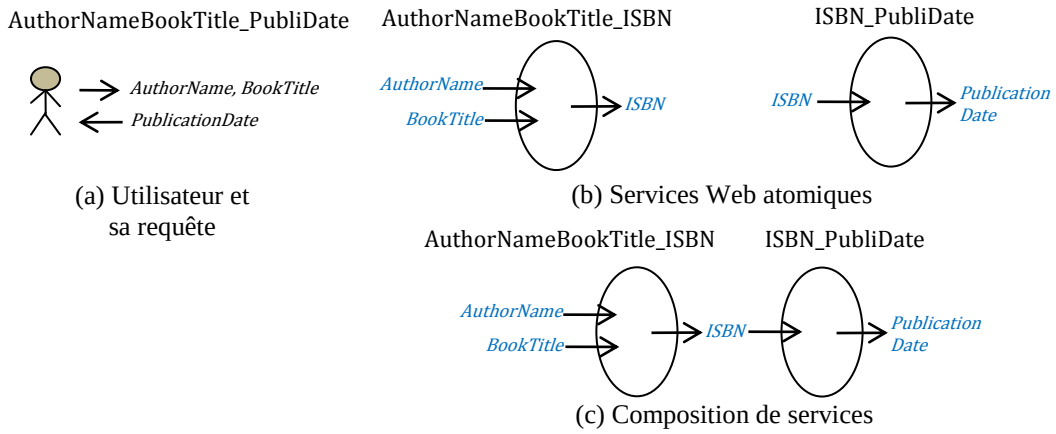


FIG. 1 - Exemple d'une interaction de services Web (c) élaborée à partir de deux services atomiques (b) et d'une requête (a).

Le principal objectif de ces travaux est de montrer que la théorie des réseaux constitue un cadre formel adapté aux problématiques de la classification et de la composition des services Web. Tout d'abord, en nous appuyant sur les travaux existants présentés au premier chapitre, nous définissons des modèles de réseaux pour chacun de ces problèmes dans le second chapitre. Après avoir présenté les fondements sur les réseaux complexes dans le chapitre trois, nous étudions la structure de ces réseaux dans les chapitres quatre et cinq. Le chapitre six introduit les éléments théoriques nécessaires à l'étude de la structure communautaire des réseaux que nous utilisons dans le chapitre sept, pour mettre en évidence cette caractéristique dans les réseaux d'interaction de services Web.

Nous précisons ci-après le contenu de chacun des chapitres.

Le premier chapitre permet de délimiter le périmètre de nos travaux dans le contexte des services Web. Nous présentons ainsi un ensemble des travaux relatifs à la classification afin de dresser un éventail des principales méthodes trouvées dans la littérature. Nous considérons d'une part les travaux qui proposent des modèles d'organisation et d'autre part ceux qui abordent la problématique par le biais de la classification automatique. En ce qui concerne la composition, nous considérons uniquement les travaux qui abordent cette problématique sous l'angle des réseaux en faisant le distinguo entre les approches graphes et les approches réseaux complexes. Quel que soit le type de relation considéré (similitude ou interaction), il est fait appel à des fonctions de mise en correspondance pour déterminer la similitude entre un ensemble de paramètres. Nous présentons les travaux sur lesquels nous nous appuyons pour définir nos fonctions de mise en correspondance.

Le second chapitre est consacré à la présentation des modèles de réseaux que nous proposons d'utiliser pour décrire les relations d'interaction et de similitude de services Web. Nous introduisons quatre fonctions qui permettent de construire des réseaux de similitude d'opérations de services Web. Ces réseaux se déclinent par ailleurs en fonction du type de description (syntaxique et sémantique). Ainsi, huit réseaux de similitude sont définis. Les réseaux d'interaction sont définis à trois niveaux de granularité (service, opération, paramètre). Deux fonctions de mise en correspondance syntaxique et quatre fonctions de mise en correspondance sémantique sont utilisées pour apparier les paramètres. Les réseaux d'opérations se distinguent par ailleurs en fonction du mode d'invocation (totale, partielle). Ceci permet de définir seize types de réseaux d'interaction.

Dans le chapitre trois, nous introduisons les réseaux complexes. Nous présentons brièvement les différents types de grands graphes de terrain. Nous rappelons les définitions des propriétés topologiques de base utilisées pour caractériser ces réseaux. Nous présentons ensuite des propriétés caractéristiques des grands graphes de terrain : la propriété petit monde ainsi que la propriété sans échelle. Cette dernière traduit les inégalités de la distribution des liens dans le réseau, à savoir une majorité faiblement connectée et une minorité fortement connectée. En dernier lieu, nous donnons les propriétés topologiques de grands graphes de terrain caractéristiques des divers domaines d'application.

Le chapitre quatre est consacré à l'analyse topologique des réseaux d'interaction de services Web. Dans un premier temps, nous présentons la méthodologie d'expérimentation. Nous

détaillons notre cheminement pour le choix d'une collection de services Web. Nous présentons brièvement les fonctionnalités de l'extracteur de réseaux mis au point pour générer les réseaux à partir des descriptions de services. Nous présentons ensuite successivement l'analyse des propriétés topologiques des réseaux de paramètres et d'opérations issus des descriptions syntaxiques et sémantiques des services de la collection retenue. Nous établissons par ailleurs une comparaison des réseaux en termes de granularité (opération, paramètre) et de description (syntaxique, sémantique). Finalement, nous intégrons les réseaux de services Web dans un comparatif de grands graphes de terrain en confrontant les résultats obtenus sur leurs propriétés topologiques.

Nous adoptons la même démarche dans le chapitre cinq pour l'analyse topologique des réseaux de similitude. Nous justifions le choix d'une collection et nous présentons les huit réseaux extraits de cette collection. Nous menons une analyse comparative des propriétés topologiques de ces réseaux. Le but visé à ce niveau n'est pas tant de confronter les réseaux de similitude aux grands graphes de terrain, mais plutôt de mieux appréhender la structuration des communautés de services formées dans les différents réseaux. A cette fin, nous observons la structuration ainsi que la répartition des opérations dans les composantes des réseaux. Nous établissons le lien entre les composantes, la notion de communauté et les domaines dont sont issus les services.

Dans le chapitre six, nous introduisons la détection de communautés dans les réseaux complexes. Cette problématique primordiale vise à trouver un niveau intermédiaire entre les aspects microscopiques et les aspects macroscopiques des réseaux complexes. Nous rappelons la définition des communautés et introduisons les mesures qui permettent de les caractériser. Nous présentons un échantillon représentatif des diverses solutions présentées dans la littérature pour la découverte de communautés. Les mesures de performance, qui permettent de qualifier ces algorithmes, sont définies à la suite. Finalement, nous analysons les travaux portant sur la comparaison des algorithmes de détection de communautés.

Dans le chapitre sept, le but visé est de mettre en évidence une structuration communautaire dans les réseaux d'interaction de services Web. Nous justifions tout d'abord le choix des collections utilisées ainsi que la méthodologie d'analyse. Une première série d'expérimentations porte sur des réseaux de paramètres et d'opérations extraits d'une collection artificielle de services dédiée à l'étude de la découverte de compositions. La connaissance de la structure communautaire du réseau nous permet ainsi d'évaluer les performances respectives des algorithmes utilisés dans une situation maîtrisée. La seconde série d'expérimentations porte sur la collection utilisée dans les chapitres précédents, afin d'apporter des éléments de réponse sur l'existence et la nature des communautés dans les réseaux d'interaction.

Enfin, nous concluons ce manuscrit par une synthèse de nos contributions. Nous précisons les améliorations envisageables et les perspectives pour la poursuite de nos travaux.

1. TRAVAUX CONNEXES

Le but de ce chapitre est de présenter les travaux connexes qui permettent de situer notre travail. Nous faisons ainsi le point sur les approches de la composition et de la classification de services Web qui ont inspiré notre travail.

La *classification* induit la notion de communauté. Le Larousse donne comme définition de la communauté « Ensemble de personnes unies par des liens d'intérêts, des habitudes communes, des opinions ou des caractères communs ». En ce qui concerne les services Web, ce terme peut prendre des significations distinctes en fonction des activités visées. Ainsi dans le cadre de la publication qui vise à permettre et à accroître la visibilité et la découverte des services, les communautés représentent plus particulièrement des domaines d'intérêts. Lorsque les préoccupations concernent la substitution de services, les communautés sont formées de services aux fonctionnalités similaires.

La *composition* est le fait de grouper plusieurs services pour répondre à un besoin donné. En effet, les nombreux services atomiques disponibles sur Internet ne peuvent pas toujours satisfaire individuellement des requêtes spécifiques. Ces services doivent alors être intégrés pour créer des services composites à valeur ajoutée. La recherche de compositions est l'extension du problème de la découverte à un ensemble de services liés par des relations d'interaction. Il fait donc lui aussi appel à la notion de similitude et peut tout aussi bien profiter d'une organisation sous forme de communautés. Dans ce cadre les communautés regroupent alors les services selon leur capacité à interagir.

Pour évaluer le degré de *similitude* ou la possibilité d'interaction entre deux services, une opération de mise en correspondance est mise en œuvre. Dans les deux cas, il s'agit de déduire des correspondances entre des éléments constitutifs des descriptions de services. La mise en correspondance est donc un concept fondamental commun aux deux types de relations étudiées. C'est pourquoi nous commençons ce chapitre par un récapitulatif des différentes façons d'aborder le problème de la mise en correspondance. Nous nous intéressons ensuite aux approches de classification de services. Enfin, avant de conclure le chapitre, nous dressons un état de l'art des travaux relatifs à la composition de services utilisant une approche réseau.

1.1 Mise en correspondance de services Web

La recherche de similitude est une classe générale de problèmes dans laquelle un objet donné, l'objet de la requête, est comparé à l'ensemble des objets d'une collection dans le but de récupérer ceux qui lui ressemblent le plus [10]. La recherche de similitude s'opère par le biais d'une fonction de mise en correspondance.

Un aspect central de la mise en correspondance de services Web est de trouver une « bonne » notion de similitude. La façon dont la similitude est définie apparaît comme un élément fondamental pour déterminer la façon dont les services correspondent à une requête, comment

ils peuvent être composés ou encore comment ils peuvent être classifiés. Les mesures de similitude ont été largement utilisées dans plusieurs domaines. Nous pouvons notamment citer les systèmes d'information [11] [12], la science cognitive, les bases de données [13] [14], le génie logiciel [15] et l'intelligence artificielle [16] [17].

Dans le paradigme SOC (Service Oriented Computing) [18], la recherche de similitude entre des services a des applications dans trois activités principales de leur cycle de vie, la découverte, la composition et la publication :

- La *découverte* exploite les informations liées aux services Web dans le but de localiser des services capables de répondre à une requête particulière avec la meilleure adéquation possible. Dans le processus de découverte, la mise en correspondance consiste à rechercher les similitudes potentielles entre la requête et les services Web publiés dans les annuaires. Elle s'opère par une comparaison des propriétés de la requête avec les propriétés des services Web disponibles. Plus particulièrement, les services recherchés doivent produire les mêmes buts que le service demandé et exiger les même entrées.

- Dans le processus de *composition*, une fonction de mise en correspondance intervient pour mettre à jour des similitudes entre deux services qui vont s'enchaîner. Elle s'applique à déterminer la compatibilité entre les buts d'un premier service et les entrées d'un second service.

- Pendant la *publication*, les services Web peuvent être organisés selon des critères de similitude. L'organisation qui découle de ce processus de classification peut être ensuite utilisée pour améliorer la découverte, ou à des fins de substitution. La substitution intervient pour le remplacement d'un service par un autre service équivalent pour des raisons d'indisponibilité ou d'insatisfaction du service proposé. La substitution peut être un processus automatique mis en œuvre pendant l'exécution de la composition. Elle peut aussi prendre la forme d'une recommandation. Un client peut passer du temps à chercher manuellement un service qui correspond à ses besoins en auscultant les catégories dans un registre et invoquer un service qui finalement se révèle indisponible. Dans cette situation, des services similaires peuvent lui être suggérés [19].

1.1.1 Fonctions de mise en correspondance

Les approches pour la mise en correspondance dépendent de *l'information considérée*, c'est-à-dire de la partie de la description utilisée. Une première catégorie d'approches se concentre sur l'aspect dynamique, c'est-à-dire les processus. Une deuxième catégorie prend en compte les aspects non fonctionnels comme la qualité de service, des informations générales comme la catégorie, le fournisseur, la description textuelle des services. Enfin une troisième catégorie s'intéresse aux aspects fonctionnels en considérant les interfaces avec le nom des opérations, les noms, types et concepts des paramètres. Dans ce travail nous nous restreignons à cette

vision du problème. Nous considérons la mise en correspondance des services Web par le biais des noms et des concepts associés aux paramètres des opérations.

Considérons deux objets *x* et *y*, deux types de correspondances peuvent être envisagés entre ces deux objets, la correspondance syntaxique et la correspondance sémantique.

Correspondance syntaxique

La mise en correspondance syntaxique vise à apparier des paramètres à partir de leurs orthographes respectives. On peut distinguer la correspondance égale et la correspondance approximative.

- La correspondance ***égale*** utilise l'équivalence syntaxique stricte. Deux objets sont dit similaires si et seulement si ils possèdent une orthographe identique. Cependant, deux objets ayant des représentations sensiblement différentes, comme `GOVERNEMENT` et `GOVERNMENT` ne peuvent pas être appariés par ce type de correspondance stricte.

- La correspondance ***approximative*** utilise des fonctions de distance *d(x, y)* pour quantifier la similitude entre deux chaînes de caractères *x* et *y*. Si la distance entre deux objets est au dessus d'un certain seuil, ces objets sont dits similaires. Dans [20] et [21], les auteurs rapportent différentes fonctions de distance. La mise en correspondance approximative est plus flexible que la mise en correspondance égale. En effet, deux termes avec des orthographes différentes peuvent avoir la même sémantique et par conséquent être interchangeables. Les vocabulaires utilisés peuvent comporter des erreurs typographiques comme par exemple dans les deux noms de paramètres `GOVERNEMENT` et `GOVERNMENT`, comporter des abréviations comme par exemple `PhysicianPassword` et `PhysicianPwd`. Ce n'est cependant pas suffisant pour détecter que deux termes comme `PRIX` et `TARIF` peuvent être interchangeables. Ce problème est résolu avec le deuxième type de mise correspondance grâce à la comparaison des concepts ontologiques.

Correspondance sémantique en utilisant des ontologies

La mise en correspondance sémantique vise à apparier des concepts organisés dans une relation hiérarchique de subsomption. On peut distinguer deux types d'approches pour évaluer la similitude sémantique entre concepts dans une ontologie: les approches basées sur la distance, c'est-à-dire sur la structure de l'ontologie [17], et les approches utilisant le contenu informatif des concepts [22] [23]. Dans le premier cas, la similitude est évaluée par la distance qui sépare les concepts dans l'ontologie. Dans le second cas, on associe aux concepts l'information véhiculée par ce concept au sens de la théorie de l'information. La similitude entre deux concepts est alors mesurée par la quantité d'information qu'ils partagent. Notons que des approches hybrides ont été proposées. Dans le cadre des services Web, la similitude de concept s'apparente aux approches basées sur la distance.
Pour comparer les sorties d'une requête aux sorties d'un service publié, quatre degrés de mise en correspondance sont utilisés dans [24]. Ces quatre degrés de mise en correspondance sont

nommés *exact*, *plug-in*, *subsumes* et *fail*. Nous en donnons la signification ci-après en notant pour des questions de commodité C_R le concept associé à la requête et C_{SW} le concept associé au service publié. Les exemples cités sont basés sur le fragment d'ontologie ci-dessous (cf. figure 2).

- La correspondance ***exact*** comporte deux clauses. La première clause est l'équivalence des concepts. Si C_R et C_{SW} sont deux concepts équivalents, alors la correspondance est dite *exact*. La deuxième clause est une relation de subsomption. Si C_R est une sous classe de C_{SW}, alors les deux concepts sont également dits en correspondance exacte. Cela suppose qu'en publiant C_{SW}, le fournisseur s'engage à fournir des sorties compatibles avec la sous classe immédiate de C_{SW}. A titre d'exemple, en reprenant l'extrait d'ontologie de la figure 2, ceci équivaut à dire qu'en publiant `roman historique`, un fournisseur s'engage à fournir des romans historiques de type `classique, médiéval et moderne`.

- La correspondance ***plug-in*** correspond à la situation où le concept du service est une généralisation du concept de la requête. Autrement dit si C_{SW} subsume C_R, on parle de correspondance plug-in. Dans ce cas, C_{SW} est un ensemble qui généralise C_R. Par exemple, un service qui fournit des romans (`roman`) pourrait être utilisé par une requête qui attend des romans historiques modernes (`moderne`). Dans ce cas, la relation entre C_R et C_{SW} est plus fragile que dans le deuxième cas de figure de la relation exact. On peut en effet attendre qu'un service qui publie une sortie comme étant un roman (`roman`) fournisse certains types de romans, mais on ne peut en revanche pas attendre par exemple qu'il fournisse tous les types de romans médiévaux (`médiéval`).

- La correspondance ***subsumes*** correspond à la situation où le concept du service est une spécialisation du concept de la requête. Autrement dit si C_R subsumes C_{SW}, on parle de correspondance subsumes. Dans ce cas, le service ne satisfait pas complètement la requête. Ce service peut être utilisé pour atteindre partiellement le but de la requête. Un ou plusieurs services supplémentaires devront éventuellement être utilisés pour satisfaire l'intégralité des buts de l'utilisateur. Supposons que la requête concerne le concept `roman historique`, le service quant à lui peut fournir des romans historiques `classique et` l'utilisateur devra donc faire appel à d'autres services pour obtenir les deux autres spécialisations de `roman historique (médiéval, moderne)`.

- Le cas ***fail*** traduit le fait qu'il n'existe aucune relation de subsomption entre C_R et C_{SW}. Il n'existe par exemple aucune relation de subsomption entre `roman de fiction` et `roman historique`.

En terme de satisfaction de la requête, ces quatre degrés de mise en correspondance sémantique peuvent être ordonnés selon une échelle de préférence comme suit : *Exact* > *Plug-in* > *Subsumes* > *Fail*.

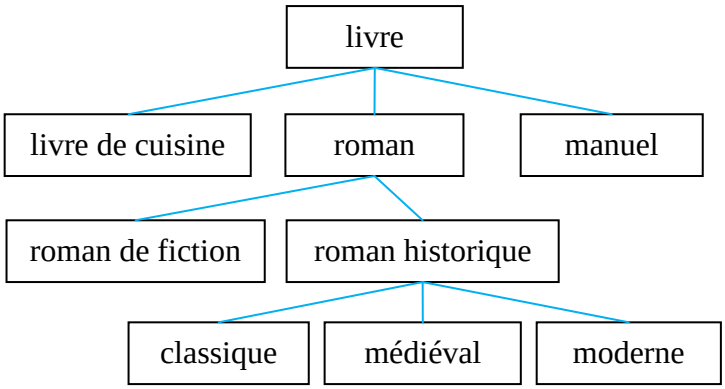


FIG. 2 – Fragment d'ontologie relative aux livres.

1.1.2 Correspondance dans la découverte de services Web

La mesure de similitude joue un rôle important dans la recherche de services pour la découverte et la composition ainsi que dans la classification. Pour la découverte, la mise en correspondance peut se faire sur des informations textuelles et sur les paramètres en entrée et les paramètres en sortie par une mise en correspondance verticale. Pour la composition, la mise en correspondance est horizontale ; elle s'applique sur les paramètres en sortie du premier service comparé et sur les paramètres en entrée du deuxième service comparé. Par ailleurs, selon la nature des objets à traiter, la mise en correspondance peut se décliner selon une mise en correspondance syntaxique ou une mise en correspondance sémantique. Notons néanmoins que la mise en correspondance sémantique peut être utilisée sur des descriptions syntaxiques en enrichissant les descriptions pour le traitement. Diverses solutions ont été proposées dans la littérature telles que l'utilisation d'outils comme la base de données lexicale WordNet [25] ou des méthodes comme l'analyse sémantique latente [26] [27].

Dans ce travail, nous nous intéressons plus particulièrement à la similitude et à l'interaction des services Web. Toutefois, nos travaux sont en partie inspirés de recherches conduites dans le cadre de la découverte de services. Nous présentons brièvement ci-après les contributions dans le domaine de la découverte sur lesquels nous nous sommes appuyés.

On distingue principalement trois catégories de correspondances. La première catégorie concerne la mise en correspondance des entrées et sorties. Cette catégorie est adoptée dans [24] ; la mise en correspondance est opérée sur les paramètres des requêtes et des services publiés en inférant des relations entre leurs concepts. La deuxième catégorie concerne la mise en correspondance des pré-conditions et effets. La troisième catégorie concerne à la fois les entrées et sorties et les pré-conditions et effets. Elle est adoptée dans [28] [29].

Dans la mise en correspondance proposée par [28], les auteurs définissent des relations ensemblistes entre l'ensemble des buts recherchés dans une requête et les offres d'un service. Pour lever l'éventuelle ambigüité des descriptions [30], des intentions sont attribuées à la fois aux buts recherchés O_R et aux offres de services O_{SW}. Ces intentions reflètent le caractère obligatoire ou facultatif d'un paramètre. Les relations ensemblistes sont alors interprétées en fonction de ces intentions. La première intention, notée ∀, exprime le fait que *tous* les

éléments du but sont obligatoires ou que *tous* les éléments annoncés par un service publié sont produits. La deuxième intention, notée ∃, exprime le fait que seulement *certains* des buts recherchés sont obligatoires ou que seulement *certains* des éléments promis par un service sont fournis. Les interactions entre les relations ensemblistes ***Match***, ***ParMatch***, ***PossMatch***, ***PossParMatch***, ***NonMatch*** et les intentions sont présentées dans le tableau 1.

TAB. 1 – Interaction entre les relations ensemblistes et les intentions relatives aux buts d'une une requête et aux éléments fournis par un service Web [28].

Intention de *R/SW*	$I_{SW} = \forall$		$I_{SW} = \exists$	
$I_R = \forall$	$O_R = O_{SW}$	Match	$O_R = O_{SW}$	PossMatch
	$O_R \subseteq O_{SW}$	Match	$O_R \subseteq O_{SW}$	PossMatch
	$O_R \supseteq O_{SW}$	ParMatch	$O_R \supseteq O_{SW}$	ParMatch
	$O_R \cap O_{SW} \neq \emptyset$	ParMatch	$O_R \cap O_{SW} \neq \emptyset$	PossParMatch
	$O_R \cap O_{SW} = \emptyset$	NonMatch	$O_R \cap O_{SW} = \emptyset$	NonMatch
$I_R = \exists$	$O_R = O_{SW}$	Match	$O_R = O_{SW}$	Match
	$O_R \subseteq O_{SW}$	Match	$O_R \subseteq O_{SW}$	PossMatch
	$O_R \supseteq O_{SW}$	Match	$O_R \supseteq O_{SW}$	Match
	$O_R \cap O_{SW} \neq \emptyset$	Match	$O_R \cap O_{SW} \neq \emptyset$	PossMatch
	$O_R \cap O_{SW} = \emptyset$	NonMatch	$O_R \cap O_{SW} = \emptyset$	NonMatch

Pour illustrer la lecture du tableau, considérons à titre d'exemple le cas suivant. D'une part, un utilisateur souhaite obtenir tous les paramètres spécifiés dans son but ($I_R = \forall$) et le fournisseur affirme pouvoir délivrer tous les paramètres spécifiés par son service publié ($I_{SW} = \forall$). D'autre part, nous avons la relation ensembliste $O_R \subseteq O_{SW}$. Dans ce cas, les besoins de l'utilisateur seront totalement satisfaits par le service publié puisque celui-ci peut fournir tous ses paramètres. On dit alors que l'on a un Match.

Si l'on fait abstraction des intentions, **Match** signifie que le service satisfait totalement la requête. **Parmatch** traduit la situation où l'offre satisfait partiellement la requête. **NonMatch** correspond à la situation où l'offre est non pertinente par rapport aux buts recherchés.

Dans [29], les auteurs étendent le travail précédent avec deux notions supplémentaires nommées *RelationMatch* et *ExcessMatch*. Ces deux notions présentent l'avantage de considérer des offres qui seraient exclues par les relations précédentes.

> ***RelationMatch*** signifie que l'offre ne peut pas satisfaire la requête directement mais peut fournir des fonctionnalités connexes. Un tel service peut alors être utilisé en combinaison avec d'autres. A titre d'exemple, un service proposant la réservation de billets d'avions par le biais des codes d'aéroports peut être retenu par la fonction RelationMatch pour un utilisateur qui souhaite réserver un billet en fournissant le nom des villes. Un service tiers peut alors produire un code d'aéroport sur la base d'un nom de ville.

ExcessMatch signifie que le service offert est en mesure de satisfaire la requête mais son utilisation pourrait se traduire par des effets supplémentaires indésirables non demandés par le client. Cette fonction permet par exemple d'informer un client qui souhaite seulement acheter un téléphone par le biais d'un service, que ce service lie l'achat de téléphones à un contrat téléphonique.

1.1.3 Discussion et Conclusion

Nous avons identifié les différentes approches pour la correspondance en distinguant les approches syntaxiques et sémantiques. Nous avons ensuite présenté les propositions pour la découverte de services Web qui ont inspiré notre recherche. Suite à ces présentations, nous dressons les conclusions suivantes.

Un de nos objectifs étant d'établir le paysage topologique des réseaux de services Web selon leur déclinaison syntaxique et sémantique, nous avons besoin d'un éventail de mesures de similarité pour définir nos modèles d'interaction et de similitude. Tout d'abord nous tissons un lien entre découverte, composition et classification. En effet, même si les relations de subsomption basées sur la logique sont développées et utilisées dans le cadre de la découverte des services Web pour inférer des relations sémantiques, ces notions de subsomption peuvent aussi être adaptées à la réalisation de compositions à un niveau fonctionnel comme il est précisé dans [31]. Nous définirons donc nos fonctions de mise en correspondance pour l'interaction et la similitude des services Web sur la base des relations sémantiques introduites par [24] et de la notion de relations ensemblistes introduites par [28] et [29]. Pour la correspondance syntaxique dans les relations d'interaction et de similitude, nous utiliserons des métriques de comparaison de chaînes de caractères.

Dans les réseaux de services Web, les entités mises en correspondance sont les paramètres des services. Cette information que nous utilisons pour l'évaluation de la similitude est une information ciblée. Nous nous concentrons en effet sur l'aspect fonctionnel qui est exprimé par les paramètres des opérations. Bien que d'autres propriétés puissent être utilisées, la fonctionnalité d'un service qui est représentée par ses paramètres reste le cœur du système. Il est cependant envisageable que notre schéma soit étendu à ces autres propriétés des services.

1.2 Similitude et classification de services Web

Dans ce paragraphe, nous présentons divers travaux ayant trait à la classification de services en vue de la publication, de la découverte et de la substitution. Les répertoires de services Web constituent des espaces qui sont des éléments clés dans les systèmes où l'information partagée doit être gérée de façon efficace. La classification a pour but d'organiser ces espaces. Au sens courant et général du terme, la classification fait référence à un système de classement, à l'action de distribuer par classes, par catégories. Une seconde définition plus spécifique et celle de la classification automatique. Cette notion fait référence à un ensemble de méthodes automatiques permettant de répartir les éléments d'un ensemble en groupes, c'est-à-dire d'établir une partition de cet ensemble. Dans cette partie, nous serons amenés à parler de classification selon les deux définitions précédentes. Les caractéristiques des

services utilisées pour calculer la similitude qui sert de base à la catégorisation diffèrent selon les approches. Parmi les critères utilisés, nous pouvons citer selon une granularité décroissante : le domaine d'application, la description textuelle, l'interface qui comprend le nombre et le nom des opérations, et la signature des opérations qui comprend le nombre, le nom, le type et le concept des paramètres.

La classification peut être bénéfique pour plusieurs activités du cycle de vie des services Web. Elle peut permettre de faciliter, d'optimiser, d'automatiser l'efficacité et l'efficience des processus de découverte, de composition, d'exécution et de gestion des services Web [32]. Dans le cadre de la découverte, l'efficacité et l'efficience des algorithmes peuvent être améliorées en utilisant une classification des services basée sur le domaine d'intérêt, sur les propriétés fonctionnelles ou sur les propriétés non fonctionnelles. De la même façon, la composition peut tirer parti de la classification des services en sélectionnant les services relativement à un domaine d'intérêt à chaque étape de la synthèse de la composition. Le processus de substitution peut faire appel à la catégorisation en organisant les services en ensembles de services aux fonctionnalités similaires. Lors de l'exécution, si certains services deviennent indisponibles, alors le processus d'exécution ne peut aboutir. La classification offre ainsi la possibilité de trouver, en temps réel, des services similaires remplaçants. La publication d'un grand nombre de services est plus efficace quand les services sont organisés en catégories. L'enregistrement des services dans un registre peut se faire directement selon des catégories.

1.2.1 Eléments d'analyse

Lors de l'étude des travaux existants sur la similitude et la classification, nous avons pu identifier un certain nombre d'éléments qui nous ont aidés à les classifier. Pour présenter ces travaux de façon claire et selon nos préoccupations, nous avons retenu les quatre éléments d'analyse suivants:

- L'approche utilisée pour aborder la problématique de la classification
  On peut distinguer les travaux sur les *modèles d'organisation* avec en vue des problématiques de publication ou de substitution de services, de ceux qui se focalisent plus particulièrement sur la *classification automatique* de l'espace des services et l'algorithmique associée. Ces derniers sont également liés aux problématiques de publication.

- La démarche adoptée pour la constitution des catégories
  Dans les travaux que nous considérons, deux démarches se dégagent. La première consiste à définir des catégories de services similaires selon *une approche descendante*. La deuxième démarche consiste à définir les catégories selon *une approche ascendante*. Une approche descendante impose de concevoir et de modéliser les catégories à priori. Les services sont ensuite classés en fonction de leur profil ou bien peuvent être profilés pour correspondre à ces catégories. Dans une approche ascendante, les catégories sont extraites des services considérés. Elles sont définies à partir de l'existant.

- La description des services considérés
  Certains auteurs considèrent une description *syntaxique* des services. D'autres s'attachent à étudier des services décrits de façon *sémantique*.

- L'information utilisée pour évaluer la similitude et effectuer la classification
  Nous parlerons *d'information élargie* et *d'information ciblée.* L'information élargie fait référence à la description textuelle, aux noms des services et des opérations. L'information ciblée fait référence uniquement aux paramètres des opérations**.**

L'évaluation de la similitude et la classification de services Web peuvent ainsi être abordées selon des critères extrêmement variés. Afin de dresser un panorama ciblé des travaux connexes sur cette problématique, nous les avons regroupés selon notre premier élément d'analyse. Dans chaque cas, nous spécifions les informations relatives aux trois autres éléments d'analyse.

1.2.2 Modèles d'organisation

Dans ces travaux, le fil conducteur est d'organiser un ensemble de services Web selon des critères de similitude catégorielle et fonctionnelle. Il est fait appel à la notion de communauté qui vise à définir une entité pour le regroupement de services. Ce regroupement peut s'opérer de deux façons. Dans le premier cas, une communauté désigne un ensemble de services qui partagent le même domaine d'intérêt. Autrement dit une communauté rassemble un ensemble de services répondant à un même ensemble de besoins fonctionnels. Il s'agit là d'un niveau d'abstraction intermédiaire entre des applications clientes et les services que celles-ci désirent utiliser. Cette organisation vise principalement à accroître l'efficacité de la publication et de la découverte. Dans le second cas, une communauté rassemble un ensemble de services fonctionnellement similaires. L'objectif principal de ce modèle de communautés est de fournir un cadre à la substitution d'un service Web par un autre. La substitution peut intervenir en cas de non réponse d'un service, ou être guidée par la préférence de l'utilisateur. Une communauté regroupe alors un ensemble de services substituables.

Communauté et domaine d'intérêt

Dans cette première approche, les services Web qui partagent le même domaine d'intérêt sont regroupés dans des communautés prédéfinies.

Dans le contexte de la description des services Web et de la conception d'un cadre ontologique répondant au besoin d'une organisation sémantique, les auteurs de [33] utilisent le concept de communauté. Tous les services Web qui partagent le même domaine d'intérêt appartiennent à une même communauté. Les communautés sont des descriptions abstraites. Elles fournissent les descriptions des services souhaités, sans qu'il ne soit fait référence à aucun service Web réel. Chaque communauté est formellement définie par quatre éléments. Ces éléments sont son identifiant, sa catégorie, un ensemble d'opérations génériques et ses membres. L'identifiant est composé du nom de la communauté et d'une description textuelle.

La catégorie décrit le domaine d'intérêt de la communauté. Les opérations génériques sont des opérations abstraites qui résument les principales fonctions de la communauté. Les membres sont les fournisseurs de services. Les opérations génériques sont définies par un ensemble d'attributs fonctionnels et non fonctionnels. Les opérations génériques fournies par la communauté peuvent être utilisées "telles quelles" ou bien être personnalisées. Les services Web sont enregistrés dans une communauté après que les prestataires aient identifié la bonne communauté. Un service Web, composite par exemple, peut être enregistré auprès de plusieurs communautés dans la mesure où les différents services atomiques le composant proviennent de divers domaines d'intérêt.
Il s'agit là d'une approche descendante dans le sens où les communautés abstraites sont définies à priori. L'aspect sémantique est introduit par une ontologie de communauté.

Dans les travaux de [34], l'idée dominante est également de grouper les services selon leur domaine ontologique. Les communautés sont des ensembles de pairs fournissant des services dans le même domaine. Elles sont structurées de façon hiérarchique autour des concepts ontologiques qu'elles partagent. Ainsi, à chaque ontologie on peut associer un ensemble de communautés structurées de façon hiérarchique autour des concepts. Chaque communauté possède un « pair maître » qui connaît tous les autres « maîtres » d'un réseau pair à pair. Il a aussi connaissance des fonctionnalités offertes par chacun des membres de sa communauté. La communication dans le réseau passe par les pairs « maîtres » qui ont pour rôle d'orienter une requête vers les membres de leur communauté à même de la satisfaire. Lorsqu'un nouveau service souhaite intégrer le réseau, il contacte le pair maître d'une quelconque communauté. Il rejoint cette communauté s'il correspond au même domaine ontologique, sinon, il peut être redirigé vers sa communauté si celle-ci est préalablement peuplée à l'aide des informations sur la structure globale du réseau maintenu dans les pairs maîtres. Dans la négative, il crée sa propre communauté dont il devient le pair « maître ». Charge à lui d'informer l'ensemble des pairs maîtres de l'existence d'une nouvelle communauté. Par ailleurs, chaque maître maintient pour chacun des membres de sa communauté une information concernant les interactions possibles avec les membres d'autres communautés sous forme d'un graphe orienté. Cette structuration hiérarchique et pair à pair auto-organisante, fournit une réponse adéquate au passage à l'échelle. Elle est utilisée pour la recherche de compositions de services.
L'aspect sémantique est réalisé par le domaine ontologique auquel chaque service est relié. L'approche est ascendante.

Communautés et services alternatifs

Nous retrouvons le concept de communauté dans [35] où ce sont essentiellement des containers de services Web alternatifs. Les communautés fournissent des descriptions de services génériques en décrivant les capacités attendues, comme par exemple une interface de réservation de vols. Les interfaces contiennent un ensemble d'opérations définies sans référence à des services réels. Les communautés sont peuplées par l'enregistrement de leurs services par les fournisseurs. L'enregistrement requiert une mise en correspondance entre les opérations du service et celles de la communauté. Grâce à un niveau de flexibilité, les services peuvent s'enregistrer avec seulement un sous ensemble des opérations de la communauté. Cette approche descendante est réalisée par SELF-SERV (compoSing wEb accessibLe

inFormation & buSiness sERVices), un système pour la gestion dynamique de services Web. Le dispositif est utilisé dans le cadre de la découverte et de la composition. Au moment de l'exécution, lorsqu'une requête correspond à une fonctionnalité donnée, la communauté correspondante est identifiée et la requête est déléguée à un de ses membres. Une police de sélection de critères non fonctionnels permet de choisir un membre parmi ceux d'une communauté.

Dans [36], les auteurs proposent également de grouper les services Web dans des communautés. Les services Web d'une communauté répondent à un même besoin. Ils sont définis comme étant fonctionnellement similaires. Chaque communauté est donc associée à une fonctionnalité spécifique. Cette fonctionnalité est représentée par un concept ontologique. La communauté est définie par un triplet contenant un service Web abstrait, un ensemble de services Web concrets et un module de mise en correspondance entre les opérations abstraites et les opérations des services Web concrets. Les services Web abstraits sont des interfaces contenant un ensemble d'opérations abstraites. Une communauté est un ensemble de services Web substituables. Cette approche a pour objectif de répondre à la problématique de la substitution de services Web. La substitution est définie comme l'opération de remplacement d'un composant par un autre. Le remplaçant doit produire les mêmes sorties et satisfaire les mêmes exigences que le composant remplacé. Les communautés sont publiées à des fins de découverte. Les services Web concrets les mieux adaptés à une demande sont sélectionnés dans la communauté par l'utilisateur. Les incompatibilités structurelles, incompatibilités de nom, de type et d'ordre des paramètres entre une opération concrète et une opération abstraite, sont gérées au moment de l'exécution par des fonctions d'adaptations. Cette approche descendante intègre un aspect sémantique par le biais du concept associé à chaque fonctionnalité.

1.2.3 Classification automatique

Un effort considérable a été fait pour développer des méthodes automatiques ou semi-automatiques pour classer les services Web selon leur domaine d'application. Les solutions proposées peuvent faire appel à des techniques d'apprentissage supervisé ou d'apprentissage non supervisé communément appelées techniques de « clustering ».

Classification automatique par apprentissage supervisé

Dans ces techniques, le nombre de catégories est prédéterminé. On dispose d'un échantillon de services étiquetés i.e. dont on connaît la catégorie. L'algorithme de classification peut alors trouver ou approximer la fonction qui permet d'affecter la bonne « étiquette » à ces exemples. Suite à cet apprentissage, le classifieur peut alors catégoriser de nouveaux services.

Dans le cadre des descriptions syntaxiques de services, de nombreuses approches utilisent des éléments de texte comme par exemple la description textuelle, le nom des opérations, comme entrée aux méthodes de classification. Ainsi, dans [37], les éléments textuels des descriptions sont utilisés afin d'effectuer la classification automatique à l'aide de machines à vecteurs de support (en anglais Support Vector Machine, SVM). Dans ce travail, les auteurs utilisent une

approche descendante en considérant une information élargie issue de descriptions syntaxiques.

Dans [38], la représentation des services fait appel au paradigme du sac de mots. Dans ce dernier, un document particulier est représenté par l'histogramme des occurrences des mots le composant. Cela suppose de disposer d'un dictionnaire. Dans notre cadre, le dictionnaire est construit à partir des noms des opérations et des noms des paramètres des services. Le dictionnaire est estimé à partir d'un corpus de descriptions de services. Un service est alors représenté par un vecteur dont chaque élément représente la fréquence d'apparition d'un mot particulier dans le fichier WSDL correspondant. Un classifieur naïf de Bayes est ensuite entrainé pour apprendre le vocabulaire et les fréquences d'occurrences relatives à un domaine particulier. Suite à cette phase d'apprentissage, la classe d'un nouveau service est prédite en utilisant la règle de la classe la plus probable. Dans ce travail, les auteurs utilisent une approche descendante en considérant une information élargie issue de descriptions syntaxiques.

Dans [32], les services Web sont classés selon une similitude relative à leur domaine. Les auteurs présentent une méthode pour la classification automatique basée sur les profils OWL-S des services Web. Cette classification est opérée sur la base des descriptions textuelles et des signatures. Dans la signature, le traitement porte plus précisément sur les concepts des paramètres. L'information textuelle et la signature du service Web sont combinées dans un vecteur de caractéristiques. La stratégie utilisée pour construire le vecteur s'apparente à la représentation utilisée dans les sacs de mots. Elle en diffère néanmoins car ce n'est pas la fréquence d'apparition d'un mot qui est utilisée, mais la présence ou non d'un mot dans la description. Un mot présent est codé 1 alors que l'absence de ce mot est codée 0. Afin d'observer l'impact de la signature sur la classification, deux façons différentes de traiter cette signature sont exploitées. La première consiste à traiter le nom du concept comme du simple texte. La seconde consiste à le traiter en tant qu'annotation en appliquant un raisonneur. Ainsi, les auteurs considèrent à la fois l'information syntaxique et l'information sémantique. Dans le but de s'affranchir de l'influence de la technique de classification utilisée, les expérimentations sont menées en considérant cinq algorithmes issus de différents paradigmes d'apprentissage. Les résultats obtenus sur la collection OWLS-TC [39] démontrent l'avantage qu'il y a à combiner l'information syntaxique et sémantique pour accroître les performances des classifieurs.

Classification automatique et clustering

Dans les techniques d'apprentissage non supervisées ou clustering, le nombre de classes et leur nature ne sont pas prédéterminés. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le processus de classification dispose d'exemples sans étiquette qu'il doit classer en groupes homogènes. Pour ce faire, la similitude est généralement calculée selon une fonction de distance entre paires d'exemples.
Dans [40], la proposition est le regroupement en clusters de services Web similaires sur la base de leurs interfaces. La mesure de similitude est faite par la méthode de calcul de similitude d'interfaces proposée dans [41]. La similitude entre les services Web est calculée en identifiant la correspondance paire à paire de leurs opérations qui maximise la somme

totale du score de mise en correspondance des paires individuelles. De la même façon, la similitude entre les opérations est calculée en identifiant la correspondance paire à paire des listes de leurs paramètres en entrée et en sortie qui maximise la somme totale des scores de mise en correspondance des paires individuelles des entrées et sorties. Les calculs sont faits sur les noms des opérations et des paramètres par similitude lexicale. Les groupes ou clusters sont déterminés à partir de la matrice de similarité et d'une méthode de clustering hiérarchique. Chaque cluster est représenté par un ensemble d'opérations caractéristiques. Cette représentation est la base pour la classification de nouveaux services qui est opérée par la technique du plus proche voisin. Elle est aussi conçue pour être utilisée lors de la découverte et pour la récupération de services similaires. Dans cette approche ascendante, la dimension sémantique est apportée par l'utilisation de WordNet pour associer les noms de paramètres à des concepts.

La méthode dénommée Semantic Web services Clustering utilisée dans [42] est relativement proche de la précédente. Les services Web sont groupés en clusters sur la base de leur similitude par une méthode agglomérative de clustering hiérarchique. La mesure de similitude est une somme pondérée de cinq éléments : les termes extraits de la description des services présente dans les registres, le profil OWL-S, le modèle OWL-S, la description WSDL et le « grounding ». Une matrice de similitude est construite sur la base des mesures de similitude obtenues pour chaque paire de services Web. A partir de cette matrice, la méthode de clustering permet de récupérer les groupes de services Web similaires. De la même façon que précédemment, l'aspect sémantique est apporté par l'utilisation de WordNet. Il s'agit d'une approche ascendante. Le regroupement permet l'amélioration des pratiques de découverte, et les services découverts sont ordonnés en indiquant leur degré de pertinence selon des termes propres aux clusters.

Dans [43], les auteurs utilisent les notions de services Web équivalents et de services Web remplaçants. Ces notions sont basées sur la comparaison des signatures des opérations. Deux services sont équivalents s'ils ont les mêmes opérations en termes de nombre et de paramètres en entrée et que ces paramètres sont de même type. Un service peut être remplaçant d'un autre service s'il le spécialise. En d'autres termes, un service remplaçant offre une fonctionnalité supplémentaire. La classification est obtenue par la méthode d'analyse formelle de concepts. Cette méthode de classification automatique s'insère dans l'outil nommé Web Service Personal Address Book. Ce dernier implémente une solution pour faciliter la tâche de découverte et de sélection des services Web les plus pertinents. La classification permet à un utilisateur de choisir un service et d'identifier clairement et facilement ses substituts potentiels. Dans cette approche ascendante les descriptions considérées sont syntaxiques.

1.2.4 Discussion et Conclusion

L'adoption à large échelle de la technologie des services Web pose le problème de la gestion dynamique de leur cycle de vie. Un nombre important de travaux s'intéressent à ce problème et tirent parti de la similitude entre les services pour pouvoir les classifier. L'objectif poursuivi est l'optimisation des différentes phases du cycle de vie des services. Nous avons vu

que différentes approches et méthodes étaient utilisées pour traiter le problème. Nous faisons le point sur ces méthodes.

En ce qui concerne la classification automatique, elle est abordée aussi bien sous l'angle de l'apprentissage supervisé que sous l'angle du clustering. Les travaux dans le cadre de l'apprentissage supervisé visent à regrouper les services qui appartiennent au même domaine. Pour ce faire, ils utilisent l'information à priori contenue dans la base d'apprentissage étiquetée pour construire un vecteur caractéristique. Les principales différences entre ces travaux tiennent à la nature de l'information utilisée à ce niveau et aux algorithmes de classification utilisés. Dans les travaux présentés, on utilise aussi bien l'information syntaxique que l'information sémantique lorsque celle-ci est disponible. De plus, on fait appel aussi bien à l'information élargie qu'à l'information ciblée. Notons que l'utilisation conjointe d'information syntaxique et sémantique permet d'accroître l'efficacité des techniques de classification. Dans le clustering, les communautés ne sont pas prédéterminées mais elles sont mises à jour par les algorithmes de classification. Les travaux présentés visent essentiellement à accroître l'efficacité de la découverte et de la publication dans les registres. Ils concernent les représentations syntaxiques et sémantiques et utilisent aussi bien l'information ciblée que l'information élargie.

Pour ce qui est des modèles d'organisation, les travaux considèrent deux aspects différents de la notion de communautés. [33] ainsi que [34] emploient le terme de communauté pour répondre à une organisation ontologique de services Web qui partagent le même domaine d'intérêt. [35] définissent une communauté comme un agrégateur de services Web offrant une interface unifiée. C'est-à-dire que les services Web partagent la même fonctionnalité sans pour autant avoir les mêmes propriétés non fonctionnelles. La définition de la communauté selon [36] est très proche. Une communauté rassemble un ensemble de services répondant à un même ensemble de besoins fonctionnels.

Pour conclure sur ces travaux, nous pouvons faire plusieurs remarques.
Tout d'abord, bien que les services Web sémantiques présentent de nombreux avantages, les services syntaxiques en production restent majoritaires, et il nous semble important de continuer à proposer des solutions pour ce type de description.

Ensuite, l'utilisation de sources d'informations variées est un aspect important qui permet d'enrichir la notion de similitude entre services. Ceci suppose néanmoins de définir des techniques efficaces pour combiner des informations de nature très différente. Cela passe nécessairement par une recherche plus approfondie sur la similitude des opérations, qui sont la réelle expression des fonctionnalités d'un service. Même réduite à ce simple niveau, la notion de similitude n'en reste pas moins problématique.
Les relations entre représentation syntaxique et sémantique méritent d'être explorées plus avant. En effet, les résultats sur l'exploitation conjointe de ces deux types d'information suggèrent que ces informations sont plus complémentaires que redondantes.
La frontière entre les notions de communauté pour la substitution et de domaine est relativement lâche. En effet, dans un même domaine sont regroupés des services aux fonctionnalités identiques qui peuvent être substituables et des services dont les fonctionnalités sont complémentaires. Il faut peut-être distinguer ces deux notions au travers

de la granularité. Une communauté représente ainsi un groupe ou les éléments possèdent une cohésion très forte, alors qu'un domaine peut représenter une juxtaposition de communautés fortement cohésives qui peuvent éventuellement se recouvrir partiellement.

C'est sur ces points que porte notre travail. Dans les chapitres suivants nous proposons un modèle de classification de services Web pour les descriptions syntaxiques et sémantiques, qui utilise différents degrés de similitude entre les opérations des services. Ce modèle d'organisation est élaboré en vue de la découverte et de la substitution. Chaque définition de la similitude donne lieu à une organisation spécifique d'un ensemble de services en communautés. Le choix d'un substitut ou la recherche d'un service peut ainsi se faire préférentiellement dans un ensemble de communautés ou l'autre en fonction des contraintes de l'utilisateur ou des services.

1.3 Interaction et réseaux de services Web

De nombreux systèmes, aussi bien naturels qu'artificiels, peuvent être représentés par des réseaux, c'est-à-dire des nœuds reliés par des liens. En fonction des domaines modélisés, les nœuds et les liens peuvent représenter des individus et des relations sociales, des ordinateurs et des câbles, des molécules et des réactions chimiques, des pages Web et des hyperliens etc. Lorsqu'il s'agit de services Web, les nœuds peuvent être les services eux-mêmes, leurs opérations ou leurs paramètres, et les liens représenter les relations d'interaction qu'ils entretiennent. Dans ce paragraphe, nous dressons un état de l'art des travaux utilisant une approche réseau pour modéliser un ensemble de services Web et leurs interactions. Nous allons néanmoins distinguer les travaux selon qu'ils abordent la problématique de l'interaction avec des modèles de réseaux au sens classique ou des modèles de réseaux complexes caractéristiques des grands graphes de terrain. Dans ce qui suit, nous analysons ces travaux et nous concluons le paragraphe en identifiant leurs limites et en proposant de nouvelles pistes.

1.3.1 Eléments d'analyse

Au côté des deux axes précédemment énoncés, à savoir l'utilisation de modèles de réseaux au sens classique ou de modèles de réseaux complexes, un ensemble de caractéristiques utilisées pour la modélisation des réseaux nous guide dans la classification des travaux. Nous avons identifié cinq caractéristiques qui sont la *granularité*, le *modèle*, le *mode*, la *description* et la *mise en correspondance*. Nous détaillons chacune de ces caractéristiques ci-après.

> ***La granularité*** est relative aux entités sur les nœuds. Elle détermine leur nature. Du plus grossier au plus fin, les nœuds des réseaux de services Web sont les *services Web* eux-mêmes, les *opérations* ou les *paramètres* des services.
>
> ***Le modèle*** définit la nature des liens entre les nœuds du réseau. Nous avons pu identifier deux types de modèles : le modèle de *dépendance*, terminologie empruntée à [44] et le modèle *d'interaction*.

- Dans un modèle de dépendance, les nœuds du réseau sont les paramètres des services Web. Les liens traduisent la relation de dépendance qui existe entre un paramètre en entrée d'une opération ou d'un service et un paramètre en sortie d'une même opération ou d'un même service. Ainsi, les liens représentent soit une opération, soit un service Web.

- Dans le modèle d'interaction, les nœuds du réseau sont les opérations ou les services. Les liens traduisent une relation d'interaction ou d'invocation entre deux opérations ou entre deux services. Ainsi, chaque lien représente un ensemble de paramètres qui constitue le flux d'information entre deux opérations ou entre deux services dans le cadre d'une invocation.

Le mode d'invocation est un paramètre propre aux réseaux d'interaction. Il est relatif à la quantité d'information considérée pour créer un lien entre deux opérations (ou deux services Web). Cette information est constituée par les paramètres. Deux cas sont considérés :

- Dans un premier cas, l'ensemble des paramètres doit être fourni par un premier nœud pour pouvoir en invoquer un second. On parlera d'*invocation totale*.

- Dans un deuxième cas, une partie seulement des paramètres peut être fournie par le premier nœud pour invoquer le second. On parlera dans ce cas d'*invocation partielle*. Cette terminologie est empruntée à [25].

La description fait référence au type de description des services Web. Ces deux types sont les descriptions *syntaxique* et *sémantique*.

- Dans une description syntaxique, à chaque paramètre est associé un nom et un type XML.

- Dans une description sémantique, à chaque paramètre est associé un concept ontologique.

La mise en correspondance détermine la façon dont la similitude est calculée entre les paramètres. Elle est directement liée au type de description considéré, comme nous l'avons vu dans le paragraphe consacré à la similitude.

Nous considérons également deux éléments supplémentaires, la *nature* des données utilisées (services Web réels ou artificiels) et leur *nombre*. Ces deux derniers éléments permettent d'évaluer le réalisme des conditions d'expérimentations.

1.3.2 Approche réseau

Dans les travaux relatifs à cette approche, les réseaux sont utilisés majoritairement en vue de la synthèse de compositions. Des techniques diverses et variées sont utilisées telles que la mise en correspondance de graphes, les algorithmes de chaînage et l'interrogation de base de

données. Ce qui relie ces solutions, c'est qu'elles n'utilisent pas d'information à priori sur les propriétés topologiques des réseaux d'interaction.

Réseaux d'interaction de services sémantiques et algorithme de chaînage

Dans [34], les nœuds du réseau sont les services Web et les liens d'interaction, appelés « liens de dépendance sémantique », sont calculés en utilisant les opérateurs de mise en correspondance sémantiques introduits par [24]. Les auteurs utilisent le mode d'invocation partielle. Le réseau est utilisé pour la recherche de compositions par le biais d'un algorithme de chaînage avant à deux passes. La première passe isole le sous graphe des plans de composition et la seconde passe sélectionne la composition optimale. La partie expérimentale porte sur les performances de l'algorithme sur un réseau construit à partir d'un ensemble de descriptions variant de 100 à 5000. Les expériences sont conduites sur un nombre conséquent de services. Les auteurs se concentrent sur des descriptions sémantiques qui sont par ailleurs synthétiques et générées automatiquement.
Dans [45], le réseau d'interaction est construit sur le même modèle que dans les travaux de [34]. Sachant que l'on cherche avant tout à satisfaire les buts de la requête, les auteurs proposent l'utilisation d'un algorithme de chaînage arrière à une passe pour la recherche automatique de compositions. Une solution est sélectionnée parmi l'ensemble des plans de compositions retournés à partir d'un critère de qualité. Pour la partie expérimentale, les performances de l'algorithme sont testées sur un réseau construit à partir d'une collection de 60 descriptions artificielles. Dans ce travail, les auteurs se concentrent sur des descriptions sémantiques. Les données considérées sont artificielles et la collection de descriptions est de petite taille.

Réseaux d'interaction et mise en correspondance de graphes

Dans [46], les auteurs proposent essentiellement une méthode de génération d'un réseau d'interaction de services à partir d'un registre UDDI dans lequel les fournisseurs déclarent les relations entre services. Ils proposent un mode d'invocation totale. Les auteurs supposent en effet seulement que le nombre de paramètres en sortie d'un service doit être supérieur ou égal au nombre de paramètres en entrée d'un autre service pour qu'un lien soit créé dans le réseau. La mise en correspondance des paramètres n'est pas une problématique abordée dans ce travail. La méthode de recherche de compositions proposée est réalisée par la mise en correspondance d'un graphe de requête avec le graphe d'interaction. Des expérimentations sont conduites sur cinq ensembles de descriptions auto générées qui contiennent entre 500 et 8000 descriptions. Les liens dans le réseau sont établis selon une probabilité prédéfinie.

Réseaux d'interaction et base de données

Dans [47], le modèle présenté est un modèle d'interaction de services. Les liens entre les services Web sont pondérés selon le degré de similitude sémantique entre les paramètres. Le réseau ainsi formé est stocké dans une base de données relationnelle. Tous les chemins possibles entre services sont préalablement calculés et stockés. La recherche de compositions peut alors se faire par l'intermédiaire de requêtes SQL. On recherche tous les chemins dont les entrées correspondent aux entrées de la requête et les sorties correspondent aux sorties de

la requête. Les expériences sont conduites sur un ensemble de 10 000 descriptions synthétiques. Les descriptions sont élémentaires. Elles contiennent un paramètre en entrée et un paramètre en sortie. Elles sont de type syntaxique mais les paramètres sont liés à des concepts ontologiques. Dans ce travail, le nombre des descriptions utilisées est important, cependant nous pouvons voir une limitation qui tient au nombre de paramètres.

Réseau de dépendance

Dans [44], des services Web décrits en OWL-S sont modélisés sous la forme d'automates d'interfaces qui illustrent la dépendance entre les entrées et les sorties des services. A partir de cette représentation, l'ensemble des entrées, l'ensemble des sorties et l'ensemble des dépendances entre les entrées et les sorties sont extraits et permettent la construction du réseau des dépendances. Les auteurs suggèrent ensuite l'utilisation d'un algorithme de recherche dans les graphes pour extraire des compositions. Dans ce travail, le problème de la mise en correspondance des paramètres n'est pas abordé et l'approche reste théorique ; aucun résultat expérimental n'est fourni.

Réseau d'interaction et classification de services Web

Les travaux présentés dans [48] concernent la classification de services Web. Ils se démarquent néanmoins des travaux de classification axés sur la similitude des services. En effet, les auteurs proposent de regrouper les services qui entrent fréquemment dans une même composition. C'est la raison pour laquelle nous avons choisi de présenter ce travail à ce niveau. La classification est opérée sur un réseau d'interaction de services Web. Les nœuds du réseau sont les services et un lien d'interaction entre deux services est pondéré par le nombre de fois où les deux services sont composés. L'opération de regroupement consiste à rassembler les services de telle façon que les liens entre deux nœuds d'un même cluster soient fortement pondérés et que les liens entre deux nœuds de différents clusters soient faiblement pondérés. Les services sont groupés en clusters par l'utilisation d'un algorithme de b-coloration et un second algorithme permet de maintenir et de mettre à jour la classification. Notons que dans ces travaux, le réseau d'interaction est un réseau non orienté et que la mise en correspondance des services n'est pas abordée.

1.3.3 Approche réseaux complexes

Les travaux qui s'inspirent de cette approche visent à inférer des connaissances sur les propriétés topologiques des réseaux afin d'accroître l'efficience des processus de composition.

Réseaux d'interaction de services et analyse de liens

Dans [49], les auteurs proposent un algorithme de composition guidé par l'analyse des liens d'un réseau d'interaction. L'idée sous-jacente est que la pertinence d'un service pour entrer dans une composition est liée à son importance en termes de connectivité avec son environnement. On peut ainsi associer à chaque nœud du réseau un rang traduisant sa notoriété. Différentes métriques globales (PageRank, Hits Rank) ou locales (hubs, autorités)

peuvent être utilisées à cet effet. Cette information est alors utilisée afin de parcourir l'espace des services par le biais de l'algorithme de recherche du plus court chemin A*. Partant d'une solution initiale, la composition est construite en ajoutant de nouveaux services. L'ordre dans lequel sont auscultés les services qui peuvent potentiellement entrer dans la composition, dépend de leur rang. Le service de plus grand rang est évalué en premier et les autres sont placés dans une file d'attente prioritaire par rang décroissant. Ce processus est itéré jusqu'à ce que les compositions satisfaisant la requête soient découvertes. La similitude entre paramètres est calculée comme une distance sémantique. Elle correspond au nombre minimum de liens qu'il faut traverser dans une ontologie pour relier deux concepts. Le mode d'invocation partielle est utilisé. Les expériences sont conduites sur un réseau extrait à partir d'un ensemble de 2450 descriptions sémantiques générées automatiquement. Ce travail constitue une des rares contributions qui s'inspire des grands graphes de terrain pour incorporer une information topologique afin d'accroître l'efficacité de la composition. Cependant, malgré l'utilisation d'un grand nombre de descriptions, les résultats ne sont pas validés sur des données réelles.

Réseaux d'interaction et grands graphes de terrain

Les travaux reportés dans [25] constituent à notre connaissance la première contribution dans le domaine qui soit résolument axée sur les réseaux complexes. Cinq types de réseaux (modèle de dépendance et modèle d'interaction d'opérations et de services décliné en mode d'invocation totale et partielle) sont proposés pour représenter des services caractérisés par une description syntaxique. La similitude est calculée sur les types et sur les noms des paramètres. Pour les noms, cette fonction considère l'égalité des chaînes de caractères ainsi que la similitude approximative. A partir d'une collection de 984 descriptions WSDL réelles provenant de sites spécialisés, l'auteur étudie les propriétés caractéristiques des grands graphes de terrain à savoir la propriété petit monde et la distribution en loi de puissance des degrés. Les résultats montrent que tous les réseaux étudiés possèdent bien la propriété petit monde et que la distribution des degrés suit une loi de puissance. Néanmoins, l'exposant de cette loi de puissance est de l'ordre de 1,4 alors que pour la plupart des grands graphes de terrain cette valeur évolue entre 2 et 3 [50]. L'auteur propose un générateur de descriptions synthétiques WDSL nommé Web Service Discovery and Composition Benchmark (WSBen) basé sur la topologie du réseau de paramètres. L'utilisateur peut spécifier entre autre différents modèles de réseaux. Ceci permet de générer des descriptions en adéquation avec les modèles de réseaux complexes (Newman, Barabasi) ou les modèles de réseaux aléatoires (Erdös-Rényi). Cet outil permet de mettre en place un environnement de tests pour les problèmes de découverte et de composition de services Web [51]. L'auteur propose également un algorithme de composition basé sur la planification qui est évalué à partir d'un échantillon de services et de requêtes générées par WSBen. Il met en évidence l'influence de la topologie des réseaux sur les performances des algorithmes de composition évalués [52]. Notre travail s'inscrit dans la continuité de cette étude dans laquelle un certain nombre de zones d'ombre sont à clarifier. Tout d'abord, l'auteur considère des graphes non orientés. Il faut évaluer l'impact de cette approximation, car les réseaux de composition sont nécessairement orientés. De plus, ce travail n'aborde pas l'aspect sémantique des descriptions et les éventuelles relations entre les représentations syntaxiques et sémantiques des services Web.

1.3.4 Discussion et Conclusion

Si l'on se réfère aux éléments d'analyse des travaux abordant l'interaction des services Web sous la forme de réseaux, nous pouvons dresser le bilan suivant.

Tout d'abord, dans la majorité des travaux, ce sont des modèles de réseaux à granularité service Web qui sont proposés. Seul [25] utilise les trois niveaux de granularité. [44] utilise quant à lui un modèle de réseau à granularité paramètre. Hormis la situation peu probable où tous les services sont mono-opération, la granularité service ne nous apparaît pas comme la plus adaptée. En effet, dans la réalité, un service peut contenir plusieurs opérations et ce sont plus précisément les opérations qui sont réellement composées.

En ce qui concerne le type de description et les fonctions de mise en correspondance mises en œuvre, trois cas de figure se présentent. Certains travaux n'abordent pas cette problématique. Ainsi dans les travaux de [46], [44] et [48], le type de description n'est pas considéré et la mise en correspondance n'est en conséquence pas traitée.
Les travaux de [45], [34] et [49] utilisent des descriptions sémantiques et la mise en correspondance prend en compte différents degrés de correspondance décrits dans [24]. Dans tous les cas, un seul réseau est représenté en prenant en compte ces différents degrés. Enfin dans [25] et [47], les descriptions sont syntaxiques et le schéma de mise en correspondance se décline en syntaxique. La liaison avec la sémantique est opérée à travers WordNet. Notons que dans aucun des travaux présentés, les deux types de descriptions ne sont utilisés conjointement.

L'aspect topologie apparaît seulement dans [25] et [49]. Cependant L'utilisation faite de la topologie des réseaux prend des orientations différentes. Partant des similitudes qui peuvent être observées entre la problématique de la composition de services et celle de l'analyse des grands graphes de terrain, dans [49], les auteurs s'intéressent à la connectivité du réseau dans le but de guider un algorithme de recherche de compositions. Le travail de [25] s'inscrit résolument dans cette optique qui consiste à considérer la composition de services Web comme un nouveau domaine d'application des réseaux complexes. A ce titre, il s'attache à dresser un paysage topologique des réseaux de services Web à description syntaxique. Il met au point un générateur de descriptions WSDL en adéquation avec les propriétés topologiques identifiées afin de pouvoir comparer et étudier les performances des algorithmes de composition.
Ces travaux mettent bien en valeur l'intérêt qu'il y a à tirer parti d'une approche de la problématique de la composition via les réseaux complexes. Ils soulèvent néanmoins un certain nombre de questions auxquelles nous nous efforcerons de répondre dans ce travail.

Tout d'abord, il est nécessaire d'affiner et de compléter l'étude topologique des réseaux syntaxiques. Une caractérisation plus fine des propriétés topologiques doit aussi être menée pour des réseaux à description sémantique. La comparaison des réseaux issus de descriptions syntaxiques et sémantiques peut permettre de mieux appréhender l'influence de la sémantique sur cette topologie.

Inspirés par les travaux sur la classification de services qui entrent fréquemment dans une composition proposés dans [48], nous proposons d'utiliser les outils de découverte de communautés développés dans le cadre des réseaux complexes pour permettre la classification des réseaux d'interaction. Dans ce cadre, une communauté est un groupe de services qui interagissent de façon préférentielle.

1.4 Conclusion

Un défi notable à relever dans le domaine des services Web est la gestion de l'aspect dynamique de l'environnement et du grand nombre de services. Ceci passe indéniablement par l'organisation et la structuration de l'espace des services Web.

L'objet de ce chapitre était de fournir une vue d'ensemble des travaux relatifs à l'organisation d'un ensemble de services Web. Pour cela, nous avons présenté un panel de propositions pour la classification des services sur le critère de leur similitude. Nous avons également présenté un ensemble de propositions pour l'organisation sous forme de réseaux d'un ensemble de services sur le critère de leurs possibles interactions. A partir des études de ces travaux, nous suggérons quatre pistes de recherche.

La première est la définition d'un modèle de classification des services selon leur similitude. Nous proposons d'utiliser une modélisation réseau pour cette classification.

La seconde est la modélisation de réseaux d'interaction syntaxiques et sémantiques avec différentes fonctions de mise en correspondance. Pour les descriptions sémantiques, nous associons un réseau à chacun des degrés de mise en correspondance.

La troisième est l'étude de la topologie des réseaux modélisés en s'intéressant plus particulièrement aux propriétés caractéristiques des grands graphes de terrain.

Enfin, la dernière piste est l'utilisation des réseaux d'interaction et de la méthode de détection de communautés dans les réseaux complexes pour la classification des services Web selon une relation d'interaction.

2. MODELES DE RESEAUX DE SERVICES WEB

2.1 Introduction

Dans ce chapitre nous présentons deux modèles de réseaux de services Web, un modèle d'interaction et un modèle de similitude. Le modèle de similitude est lié à la substitution et à la découverte. Il est le support de la classification des services selon le critère de la similitude. Le modèle d'interaction est lié à la composition des services Web. Il est le support de la classification des services selon le critère de l'interaction.

La définition de ces modèles répond au double objectif poursuivi dans le cadre de notre travail. Le premier objectif est d'analyser les relations d'interaction et de similitude pouvant exister au sein d'un ensemble de services Web et de vérifier l'adéquation du paysage topologique formé par un espace de services Web avec les modèles de grands graphes de terrain. Cette analyse s'appuie essentiellement sur les propriétés structurelles fondamentales des réseaux complexes. Le second objectif est de proposer des méthodes de classification de services Web basées sur les réseaux. Cette approche est ici liée aux fonctionnalités offertes par un service.

Lors de l'étude des travaux antérieurs utilisant les réseaux pour représenter un ensemble de services Web, nous avons vu que différentes déclinaisons de réseaux existent. Ces déclinaisons dépendent d'un ensemble de paramètres (granularité, modèle, mise en correspondance, etc.). Nous utilisons ces paramètres pour la modélisation des réseaux d'interaction et de similitude. Pour cette raison, nous apportons les précisions nécessaires relatives à ces variables dans le contexte de nos travaux.

Nous nous intéressons à l'aspect fonctionnel des services et nous nous concentrons plus particulièrement sur les paramètres en entrée et en sortie des opérations. Afin d'identifier les éléments pris en compte par les fonctions de mise en correspondance, par les modes d'invocation et les fonctions de similitude, nous définissons un ensemble de notations comme suit. Un service Web désigné par une lettre grecque a un nom et est constitué d'un ensemble d'opérations. Chaque opération désignée par un chiffre a un nom et contient un ensemble de paramètres en entrée noté I et un ensemble de paramètres en sortie noté O. Dans une description syntaxique, chaque paramètre d'entrée ou de sortie du service est décrit par son nom défini par le fournisseur de services sous la forme d'une chaîne de caractères que nous désignerons par *nom*. La description sémantique désigne quant à elle le paramètre au travers d'un *concept* qui fait référence au concept ontologique. Nous présentons l'ensemble de ces notations dans le tableau 2.

TAB. 2 - Notations pour la désignation des services Web, des opérations et des paramètres.

	Décrit par	Désigné par
Service Web	nom, ensemble d'opérations	*{α, β, γ, ...}*
Opération	nom ensemble de paramètres en entrée/sortie	*{1, 2, 3, ...}*
Ensemble de paramètres en entrée/sortie		*I / O*
Paramètre	nom, concept	*{a, b, c, ...}*

La figure 3 représente un service α avec deux opérations *1* et *2*, les paramètres en entrée $I_1 = \{a, b\}$, $I_2 = \{c\}$ et les paramètres en sortie $O_1 = \{d\}$, $O_2 = \{e, f\}$.

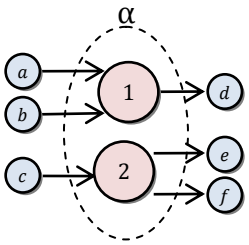


FIG. 3 - Représentation schématique d'un service Web α, avec deux opérations *1* et *2*. $I_1 = \{a, b\}$, $O_1 = \{d\}$, $I_2 = \{c\}$, $O_2 = \{e, f\}$.

2.2 Modèle de similitude

Le modèle de similitude est conçu sur la base de la similitude entre opérations. Nous ne considérons pas le niveau de granularité service car dans la réalité, ce sont les opérations qui sont en dernier ressort invoquées. Ce modèle peut se décliner en fonction du degré de similitude entre les opérations. Nous définissons quatre opérateurs ensemblistes qui traduisent différents degrés de similitude entre les ensembles de paramètres d'entrée et de sortie des opérations. Ces opérateurs utilisent les fonctions de mise en correspondance pour la comparaison des paramètres. Dans le cas des descriptions syntaxiques, la mise en correspondance porte sur le nom des paramètres et repose sur les métriques de distance entre chaînes de caractères. La mise en correspondance pour des descriptions sémantiques est opérée à l'aide des opérateurs sémantiques.

2.2.1 Définition

D'une manière générale, nous définissons un réseau de similitude comme un graphe dont les nœuds correspondent à des opérations et dont les liens indiquent un certain degré de similitude entre ces opérations. Bien évidemment, la nature de la relation de similitude est extrêmement importante. Elle peut être définie de plusieurs façons. Par la suite, nous décrivons les quatre *fonctions de similitude* que nous utilisons pour construire les réseaux et nous en donnons leur interprétation. Soient deux opérations *i* et *j*, cette fonction peut être

symétrique, *f(i, j) = f(j, i)* ou asymétrique *f(i, j) ≠ f(j, i)*. Dans le premier cas on aura affaire à un réseau non orienté alors que la seconde situation nécessite d'utiliser un réseau orienté.

2.2.2 Fonctions de similitude

Les quatre fonctions que nous définissons pour construire les réseaux de similitude sont basées sur les travaux de [28] et [29] dans le cadre de la découverte de services. Les auteurs présentent plusieurs opérateurs et les utilisent pour comparer des ensembles de concepts ontologiques dans le cadre des descriptions sémantiques. L'hypothèse essentielle sur laquelle repose la définition de ces opérateurs est qu'un utilisateur spécifie ses besoins en termes de ce qu'il veut réaliser en utilisant un service. Autrement dit, l'utilisateur connaît parfaitement les buts qu'il veut atteindre, mais la façon de les atteindre n'est pas une préoccupation essentielle. La réponse à sa requête peut être fournie par un service individuel ou un ensemble des services interagissant pour satisfaire ses besoins. L'élément prépondérant est donc les buts poursuivis par l'utilisateur qui sont représentés par les paramètres de sortie du service.

Nous avons sélectionné quatre de ces opérateurs qui traduisent différentes situations d'adéquation entre les buts d'un utilisateur et les sorties fournies par les services.
Match traduit le fait que tous les besoins de l'utilisateur sont satisfaits. *Partial match,* correspond à la situation où seule une partie des buts est satisfaite. Il faudra donc faire appel à des services supplémentaires pour satisfaire la requête. Les deux autres opérateurs que nous avons retenus ont été introduits par [29] pour prendre en compte deux situations jusque-là ignorées. *Excess match* traduit le cas où le service publié satisfait pleinement les buts de l'utilisateur et lui fournit en plus, des informations supplémentaires. Quant à *Relation match,* il a été introduit pour les situations où un service peut satisfaire les buts de l'utilisateur et que celui-ci ne peut fournir les entrées pour invoquer le service. Pour utiliser le service il devra donc faire appel à des services complémentaires.

Nous avons adapté les définitions de ces opérateurs à nos objectifs qui sont la détermination d'une valeur de similitude entre deux ensembles de paramètres. Ces fonctions sont définies en termes de relations ensemblistes. Elles opèrent selon une comparaison verticale. Supposons que nous voulons comparer deux opérations *i* et *j*. I_i et I_j sont respectivement les ensembles de paramètres en entrée de *i* et de *j*. O_i et O_j sont respectivement les ensembles de paramètres en sortie de *i* et de *j*. Cela revient à comparer I_i avec I_j et à comparer O_i avec O_j. Nous donnons ci-après la définition des fonctions de similitude définies par analogie avec les opérateurs présentés ci-dessus que nous appelons *FullSim, PartialSim, ExcessSim* et *RelationSim*.

Définition des degrés de similitude

FullSim signifie « full similarity » ou similitude totale. Deux opérations *i* et *j* sont totalement similaires si elles offrent exactement le même ensemble de paramètres en sortie ($O_1 = O_2$) et si les ensembles de paramètres en entrée se recoupent ($I_1 \cap I_2 \neq \emptyset$).

PartialSim signifie « partial similarity » ou similitude partielle. PartialSim est une fonction asymétrique. Une opération *2* est partiellement similaire à une opération *1* si certains paramètres en sortie de *1* sont manquants dans l'ensemble de paramètres en sortie de *2*

$(O_1 \supset O_2)$ et si les ensembles de paramètres en entrée des deux opérations se recoupent $(I_1 \cap I_2 \neq \emptyset)$.

ExcessSim signifie « excess similarity » ou similitude avec excès. ExcessSim est une fonction asymétrique. Une opération *2* est similaire avec excès à une opération *1* si *2* fournit tous les paramètres en sortie de *1* ainsi que des paramètres supplémentaires $(O_1 \subset O_2)$, et si *2* a au plus les entrées de *1* $(I_1 \supseteq I_2)$.

RelationSim signifie « relational similarity » ou similitude relationnelle. RelationSim est une fonction symétrique. Deux opérations *1* et *2* ont une similitude relationnelle si elles ont exactement les mêmes ensembles de sorties $(O_1 = O_2)$ et si les ensembles de paramètres en entrée sont disjoints $(I_1 \cap I_2 = \emptyset)$.
Ces définitions sont regroupées dans le tableau 4.

TAB. 4 - Définition des fonctions de similitude.

Fonction de similitude	Relations ensemblistes	Propriété de symétrie
FullSim	$(I_1 \cap I_2 \neq \emptyset) \wedge (O_1 = O_2)$	*symétrique*
PartialSim	$(I_1 \cap I_2 \neq \emptyset) \wedge (O_1 \supset O_2)$	*asymétrique*
ExcessSim	$(I_1 \supseteq I_2) \wedge (O_1 \subset O_2)$	*asymétrique*
RelationSim	$(I_1 \cap I_2 = \emptyset) \wedge (O_1 = O_2)$	*symétrique*

2.2.3 Fonctions de mise correspondance

Pour déterminer les relations entre deux ensembles de paramètres, il convient de comparer les paramètres individuellement. Cette opération est réalisée par les fonctions de mise en correspondance. Nous définissons une fonction de mise en correspondance pour les descriptions syntaxiques et sémantiques des services.

Dans la ***fonction de mise en correspondance syntaxique***, l'information que nous utilisons pour le calcul de la similitude est le *nom* des paramètres. Dans les services Web réels, il n'y a pas de consensus sur la manière de nommer les paramètres. Chaque fournisseur peut utiliser ses propres politiques de nommage. Par conséquent, des paramètres ayant des noms identiques peuvent véhiculer une information différente. De la même façon, des paramètres ayant des noms différents peuvent véhiculer la même information. Pour évaluer la similitude entre les paramètres, la correspondance syntaxique consiste à mesurer la similitude entre deux chaînes de caractères. Nous pouvons distinguer deux cas qui sont la similitude stricte et la similitude approximative. Pour la *similitude stricte*, deux paramètres sont dits similaires si leurs noms sont identiques. Pour la *similitude approximative*, deux paramètres sont considérés comme similaires si la valeur d'une fonction de distance entre chaînes de caractères est supérieure à une certaine valeur de seuil.

Dans la ***fonction de mise en correspondance sémantique***, nous évaluons la similarité entre paramètres en comparant les *concepts* ontologiques associés. La fonction repose sur la subsomption de concepts. Bien que l'on puisse utiliser l'ensemble des opérateurs classiques de mise en correspondance sémantique, nous préférons nous limiter au cas de l'équivalence

des concepts. Nous parlerons de mise en correspondance par l'opérateur *exact*. Notons que cette définition diffère de celle adoptée dans [24]. En effet l'opérateur classique *exact* englobe deux relations. La première est la relation d'équivalence des deux concepts comparés. La seconde est la relation de subsomption pour laquelle un paramètre en sortie de la requête considérée est une sous-classe d'un paramètre en sortie de la description considérée. Nous préférons nous en tenir à l'équivalence de concepts car la deuxième clause suppose qu'un service à même de fournir le général peut fournir le spécifique. Hors ceci n'est pas toujours le cas. Si l'on se réfère à l'ontologie présentée figure 2 (chapitre 1), on peut concevoir des situations où un fournisseur s'engage, par exemple, à fournir des romans historiques sans pouvoir fournir tous les sous-types de romans historiques (classique, médiéval, moderne). Dans ce cas le but de l'utilisateur ne peut être satisfait. Etant donné que notre préoccupation est de rester le plus proche possible des buts utilisateur, nous ne nous intéressons pas à la hiérarchie des concepts pour les réseaux de similitude. Ceci permet de réduire la taille des éventuelles compositions pour répondre aux besoins de l'utilisateur. Dans l'hypothèse où les buts lui sont fournis par un service, il ne lui reste plus qu'à mettre en adéquation ses entrées avec ceux du service en passant éventuellement par des services intermédiaires.

Chaque fonction de similitude permet de construire un réseau spécifique. Par la suite, nous utiliserons le nom de l'opérateur utilisé pour désigner les réseaux obtenus avec les diverses fonctions de similitude. Ainsi, un réseau FullSim est un réseau dans lequel les relations entre les opérations sont calculées selon la fonction de similitude FullSim. Les réseaux FullSim et RelationSim, du fait du caractère symétrique des fonctions de similitude, sont des réseaux non orientés. Les réseaux PartialSim et ExcessSim qui sont issus de fonctions de similitude asymétriques, sont quant à eux orientés.

2.2.4 Interprétation des fonctions

Chaque fonction de similitude conduit à une notion de similitude porteuse de sa propre sémantique. Afin d'illustrer cette variété, nous montrons par le biais d'un exemple, comment les fonctions de similitude peuvent être interprétées, et en quoi elles sont pertinentes pour comparer les opérations des services Web. Considérons l'ensemble des six opérations présentées dans le tableau 5.

TAB. 5 – Six opérations et leur signature.

Désignation	Nom opération	Entrées	Sorties
1	CITYNAME_ZIP	I_1={ZIP}	O_1={CITY-NAME}
2	CITYNAME_ZIPGEOGRAPHICALREGION	I_2 ={ZIP, GEOGRAPHICALREGION}	O_2 ={CITY-NAME}
3	GEOGRAPHICALLOCATION_ZIP	I_3={ZIP}	O_3={CITY-NAME, LONGITUDE, LATITUDE}
4	WEATHER_ZIP	I_4={ZIP}	O_4={WEATHER}
5	WEATHER_CITYNAME	I_5 ={CITY-NAME}	O_5={WEATHER}
6	WEATHERWEATHERREPORTSUBSCR_CITYNAME	I_6 = {CITY-NAME}	O_6 = {WEATHER, WEATHERREPORTSUBSCR}

Les opérations *1* et *2* sont totalement similaires ; elles produisent les mêmes sorties et l'intersection de leurs paramètres en entrée est non vide. L'opération *1* est partiellement similaire à l'opération *3* ; elle produit une partie seulement des paramètres de sortie de l'opération *3* et l'intersection de leurs paramètres en entrée est non vide. L'opération *6* est similaire avec excès à l'opération *5* ; en effet elle produit l'ensemble des sorties de l'opération *5* et un paramètre supplémentaire et l'intersection de leurs paramètres en entrée est non vide. Enfin, les opérations *4* et *5* sont similaires de façon relationnelle car elles produisent les mêmes sorties et l'intersection de leurs paramètres en entrée est vide. Nous avons pu remarquer que cet exemple est mené sur des données syntaxiques. Ce sont les noms des paramètres qui sont traités par une fonction de mise en correspondance. Ce même exemple aurait tout aussi bien pu être conduit sur des données sémantiques si nous avions considéré les concepts des paramètres au lieu de leur nom. Les relations de similitude entre ces opérations sont illustrées par la figure 4.

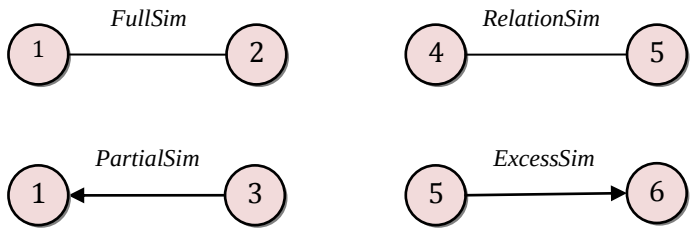


FIG. 4 – Relations de similitude entre opérations.

Dans un contexte d'utilisation des services Web, le but de l'utilisateur est souvent la chose à laquelle il est nécessaire d'attacher une importance primordiale [28]. Ainsi, supposons qu'un utilisateur cherche à obtenir le bulletin météo de sa propre ville en fournissant le nom de cette ville et son code postal. Il pourrait naturellement rechercher les opérations qui satisfassent à la fois les entrées et les sorties de sa requête. Mais de telles attentes sont généralement trop restrictives pour pouvoir être satisfaites. C'est pour cette raison que nous n'avons pas considéré ce cas dans la conception des fonctions de similitude. Et c'est justement le cas dans notre exemple, puisqu'aucune opération, dans un contexte de découverte, ne correspond à la requête.

La similitude FullSim peut être considérée comme la seconde meilleure solution, puisque celle-ci inclut les sorties attendues et une partie des entrées disponibles dans la requête. Les opérations *4* et *5* sont deux candidates potentielles pour la requête en question.

Il peut aussi arriver qu'aucune opération ne réponde à ces critères. Dans ce cas, l'utilisateur peut être amené à assouplir les contraintes relatives à son but. Dans notre exemple, supposons que les opérations *4* et *5* soient indisponibles et que l'utilisateur accepte une autre solution. La réponse se trouve alors dans l'opération *6* qui est similaire avec excès, ExcessSim, à la requête. Elle retourne un abonnement en sus du rapport météorologique par rapport à la requête initiale. Il est probable que l'utilisateur ne soit pas intéressé par ce résultat, s'il est à la recherche d'un service gratuit et que le surplus de l'abonnement soit un service payant. Dans un autre cas, il aurait pu être intéressé par des sorties supplémentaires comme une liste de bulletins météorologiques pour les villes voisines par exemple.

Supposons que l'utilisateur soit toujours en quête d'un rapport météorologique, mais qu'il ne puisse fournir qu'un code postal. Si l'opération *4* est indisponible alors aucune opération ne

peut être trouvée en utilisant FullSim, PartialSim ou ExcessSim. L'opération *5* pourra satisfaire le besoin. Elle présente une similitude relationnelle, RelationSim, avec la requête car la sortie est identique au but, mais les entrées n'ont rien en commun. Cette opération ne peut certes pas être utilisée directement, mais en collaboration avec d'autres opérations dans le cadre d'une composition. Dans ce cas, l'opération *1* peut d'abord être invoquée et son paramètre en sortie, un nom de ville, utilisé pour invoquer l'opération *5*.

Nous avons défini quatre fonctions de similitude capables de comparer des opérations sur la base de leurs ensembles de paramètres. Ces fonctions font appel aux fonctions de mise en correspondance définies précédemment et utilisant des opérateurs pour comparer les noms ou les concepts de paramètres. Chaque fonction de similitude correspond à un concept de similitude spécifique. Chaque fonction appliquée à une collection de descriptions de services Web donnera lieu à un réseau particulier.
Il est important d'insister sur le fait que les fonctions de similitude proposées sont conçues pour être complémentaires. La fonction FullSim correspond à la meilleure solution. Ensuite les fonctions PartialSim, ExcessSim et RelationalSim peuvent répondre à des situations particulières et spécifiques, directement en relation avec le contexte, comme nous l'avons vu dans l'exemple ci-dessus.

2.3 Modèle d'interaction

Pour représenter les interactions entre un ensemble de services sous forme de réseaux, les nœuds peuvent être définis à partir des trois niveaux de granularité (paramètre, opération, service). Dans un réseau dont les nœuds sont les paramètres, les liens représentent les services ou les opérations. Pour les deux autres types de granularité, les liens représentent les paramètres communs qui permettent aux services d'interagir. Bien que ces réseaux soient de nature très différente nous les désignons tous sous le vocable de réseaux d'interaction. En effet, ils traduisent de diverses façons les relations d'interaction entre un ensemble de services. Pour les distinguer on utilisera le niveau de granularité.

En outre, les réseaux se distinguent aussi selon la fonction de mise en correspondance de paramètres mise en œuvre. Celle-ci est directement liée au type de description. Comme nous l'avons vu dans l'exposé des travaux connexes, elles se déclinent différemment selon que les services utilisent une description syntaxique ou sémantique.

En ce qui concerne les granularités opération et service, les réseaux se déclinent aussi à partir d'une dernière variable, à savoir le mode d'invocation. Le mode d'invocation permet d'exprimer la quantité d'information utilisée pour relier deux nœuds. On distingue l'invocation partielle et l'invocation totale.

Si l'on considère un réseau d'opérations, on parle d'invocation totale quand une opération produit l'ensemble des paramètres nécessaires pour invoquer une autre opération. Il suffit que l'opération invoquant fournisse un seul paramètre nécessaire à l'opération invoquée pour parler d'invocation partielle. Ceci sous-entend qu'une opération peut fournir une partie seulement de ses paramètres à une autre pour que l'invocation puisse avoir lieu.

En résumé, on peut ainsi décliner des réseaux de paramètres syntaxiques ou sémantiques avec les différentes fonctions de mise en correspondance. Les réseaux d'opérations et de services possèdent quant à eux un degré de liberté en plus caractérisant le mode d'interaction.

2.3.1 Définitions

Réseau d'interaction de paramètres

Un ***réseau d'interaction de paramètres*** est défini comme un graphe orienté dans lequel les nœuds représentent l'ensemble des paramètres et les liens matérialisent les opérations. Autrement dit, un lien est créé entre chacun des paramètres en entrée d'une opération et chacun de ses paramètres en sortie. Dans ce contexte, chaque opération *i* peut être définie comme un triplet (I_i, O_i, K_i), où I_i désigne l'ensemble des paramètres d'entrée, O_i désigne l'ensemble des paramètres de sortie et K_i désigne l'ensemble des liens de dépendance. Pour construire l'ensemble des liens de dépendance, nous considérons que chaque paramètre en sortie d'une opération dépend de chaque paramètre en entrée de cette même opération. La partie gauche de la figure 5 représente trois services Web α, β et γ. Leurs quatre opérations sont numérotées *1, 2, 3* et *4*. Les neuf paramètres en entrée et sortie sont labélisés de *a* à *i*. Les relations de dépendance entre les paramètres sont représentées dans le tableau 6. A titre d'exemple, considérons l'opération *2*. Elle est définie par (I_2, O_2, K_2) où $I_2 = \{c\}$, $O_2 = \{e, f\}$, $K_2 = \{(c, e), (c, f)\}$; *e* et *f* dépendent tous les deux de *c*.

TAB. 6 – Paramètres des opérations et relations de dépendance associées.

Opération	Paramètres d'entrée	Paramètres de sortie	Liens de dépendance
1	$I_1 = \{a, b\}$	$O_1 = \{d\}$	$K_1 = \{(a, d), (b, d)\}$
2	$I_2 = \{c\}$	$O_2 = \{e, f\}$	$K_2 = \{(c, e), (c, f)\}$
3	$I_3 = \{f\}$	$O_3 = \{g, h\}$	$K_3 = \{(f, g), (f, h)\}$
4	$I_4 = \{d, g\}$	$O_4 = \{i\}$	$K_4 = \{(d, i), (g, i)\}$

Lorsque l'on considère non plus une unique opération, mais une collection entière, on peut dire qu'un paramètre *b* dépend d'un paramètre *a* si et seulement si il existe une opération *i* telle que $a \in I_i$ et $b \in O_i$. Un réseau de paramètres permet d'exprimer cette triple information : les nœuds représentent les paramètres (I_i et O_i) et les liens représentent les dépendances K_i. La figure 5 (Droite) représente le réseau correspondant aux trois opérations de la partie gauche. Par exemple, à partir de chacun des paramètres en entrée de l'opération *1*, c'est-à-dire *a* et *b*, il existe un lien en direction du paramètre en sortie, c'est à dire *d*. A partir du paramètre en entrée *c* de l'opération *2*, il existe un lien vers chacun de ses paramètres en sortie, c'est-à-dire *e* et *f*. Dans ce réseau, la présence d'un lien à partir d'un nœud *x* vers un nœud *y*, indique qu'au moins une opération utilise le paramètre *x* comme entrée et le paramètre *y* comme sortie. Ceci peut aussi être interprété en termes de *production*. Dans ce cas, un tel lien signifie qu'il existe

une ou plusieurs opérations à même de produire le paramètre *y* si le paramètre *x* leur est fourni.

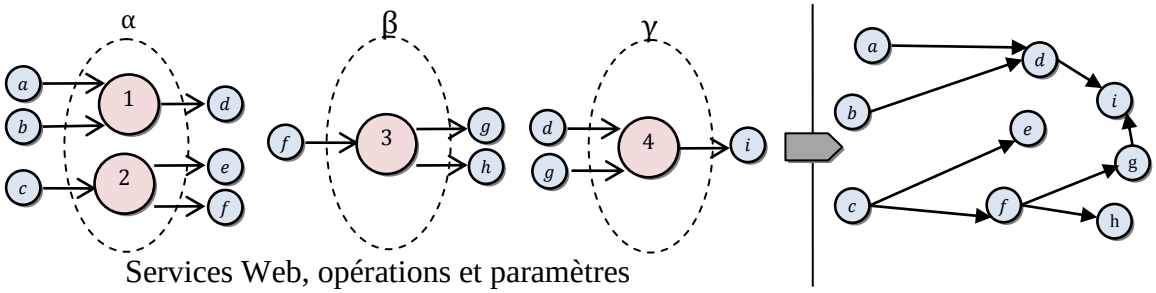


FIG. 5 - Réseau d'interaction de paramètres à 9 nœuds labélisés de *a* à *i* (Partie droite) obtenu à partir de 4 opérations numérotées de *1* à *4* (Partie gauche).

La connectivité dans un réseau d'interaction de paramètres résulte en partie du fait que certains paramètres peuvent être utilisés par plusieurs opérations. Ils peuvent d'autre part être utilisés en tant que paramètres d'entrée par certaines opérations et en tant que paramètres de sortie par d'autres opérations. Par exemple, sur la figure 5 (Partie gauche), les paramètres {*d, f, g*} apparaissent plus d'une fois, soit comme entrée, soit comme sortie de plusieurs opérations. *d* est en sortie de *1* et en entrée de *4*, *f* est en sortie de *2* et en entrée de *3*, *g* est en sortie de *3* et en entrée de *4*. Ces paramètres sont néanmoins représentés par un seul nœud dans le réseau, comme on peut le voir sur la figure 5 (Partie droite). Pour distinguer ces deux situations, nous introduisons les termes d'*instance* de paramètre et d'*archétype* de paramètre. Une **instance** est une occurrence d'un paramètre dans une description de service. Un **archétype** est le représentant de plusieurs instances similaires. Un archétype correspond à un nœud dans le réseau. Dans un réseau d'interaction de paramètres, les instances de paramètres sont représentées par le même archétype si elles sont similaires.
Dans l'exemple de la figure 5, on compte deux instances des paramètres *d*, *f* et *g*. Un archétype de paramètre est ainsi censé représenter un ensemble d'instances similaires, c'est à dire véhiculant la même information. En conséquence, il convient de décider si deux instances sont similaires et si elles doivent être groupées dans le même archétype. Cette tâche revient à la *fonction de mise en correspondance*.

Réseau d'interaction d'opérations

Un ***réseau d'interaction d'opérations*** est défini comme un graphe orienté dans lequel les nœuds représentent l'ensemble des opérations et les liens matérialisent un flux d'information entre deux opérations. Soit une opération *i* décrite par le couple d'ensemble de paramètres d'entrée et de sortie (I_i, O_i), pour traduire une relation d'interaction entre cette opération source vers une opération cible *j* décrite par le couple (I_j, O_j), deux modes d'interaction peuvent être considérés :

- Dans le cas d'une *interaction totale*, un lien est créé à partir de l'opération source *i* vers l'opération cible *j* si et seulement si pour chaque paramètre d'entrée de l'opération cible *j* il existe un paramètre de sortie similaire dans *i*. En d'autres termes,

le lien existe si l'opération *i* est à même de fournir toute l'information requise pour invoquer l'opération *j*.

- Dans le cas d'une *interaction partielle*, un lien est créé d'une opération *i* vers une opération *j* s'il existe au moins un paramètre en sortie de l'opération source similaire à un paramètre en entrée de l'opération cible. Autrement dit, pour que le lien existe, il suffit que l'opération *i* soit à même de fournir seulement une partie de l'information nécessaire pour pouvoir invoquer l'opération *j*. Notons que l'interaction partielle ne recouvre pas l'interaction totale.

A titre d'exemple, considérons l'ensemble des services représentés dans la figure 6. La partie gauche représente un ensemble de services Web nommés *α*, *β* et *γ*, leurs opérations numérotées de *1* à *4*, les paramètres en entrée et en sortie labélisés de *a* à *i*. La partie droite correspond aux deux types de réseaux d'interaction associés. Le réseau construit selon le mode d'invocation totale est représenté dans la partie supérieure (a). Le réseau construit selon le mode d'invocation partielle est représenté dans la partie inférieure (b). Considérons le réseau en mode d'interaction totale, figure 6 (a). Toutes les entrées de l'opération *3*, c'est à dire $I_3 = \{f\}$, sont incluses dans les sorties de l'opération *2*, $O_2 = \{e, f\}$. Ce que nous traduisons par $I_3 \subset O_2$. Pour cette raison, il existe un lien de l'opération *2* vers l'opération *3*. Dans cet exemple, aucune autre opération n'est à même de fournir tous ses paramètres à une opération pour pouvoir l'invoquer selon le mode d'interaction totale. Considérons maintenant le réseau en mode d'invocation partielle figure 6 (b). L'opération *4* dont les entrées sont $I_4 = \{d, g\}$, peut recevoir une partie de ses paramètres, le paramètre *d*, de l'opération *1*. Les sorties de l'opération *1* sont en effet $O_1 = \{d\}$. Ce que nous traduisons par $O_1 \subset I_4$. L'opération *4* peut recevoir une autre partie de ses paramètres, le paramètre *g*, de l'opération *3*. En effet, les paramètres en sortie de l'opération *3* sont $O_3 = \{g, h\}$. Ce que nous traduisons par $O_3 \subset I_4$. Notons dans ce cas précis que, par le biais de deux opérations, l'opération *4* est à même de recevoir tous ses paramètres en entrée. Dans d'autres cas, il se peut que malgré l'intervention de plusieurs opérations sources, une opération cible ne puisse pas recevoir tous ses paramètres d'entrée.

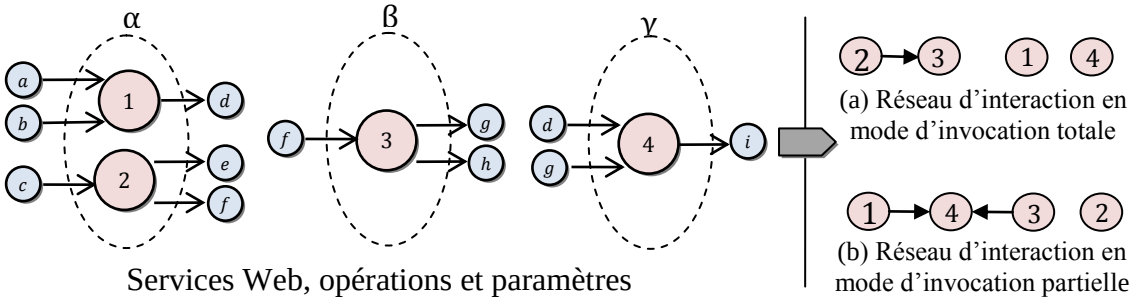


FIG. 6 - Réseau d'interaction d'opérations à invocation totale (a) et réseau d'interaction d'opérations à invocation partielle (b) issus de 4 opérations numérotées de *1* à *4* (Partie gauche).

Dans un réseau d'interaction d'opérations, un lien entre deux opérations représente la possibilité de les composer. En conséquence, il convient de décider si deux paramètres, l'un

en sortie d'une opération source et l'autre en entrée d'une opération cible, sont similaires. C'est précisément le rôle de la *fonction de mise en correspondance* définie dans le paragraphe suivant.

Réseau d'interaction de services

Un ***réseau d'interaction de services*** est défini comme un graphe orienté dans lequel les nœuds représentent l'ensemble des services et les liens matérialisent un flux d'information entre deux opérations quelconques de chacun des services. Soit un service λ décrit par un ensemble d'opérations $\{i, j, \ldots., k\}$ et un service φ décrit par un ensemble d'opérations $\{m, n, \ldots., s\}$, pour traduire une relation d'interaction entre le service source λ vers un service cible φ, il suffit qu'une opération du service source soit à même d'invoquer une opération du service cible.
On peut considérer deux cas liés aux différents modes d'invocation des opérations :

- Dans le cas d'une interaction totale, un lien est créé à partir du service source λ vers le service cible φ, s'il existe au moins une opération en interaction totale entre les deux services.

- Dans le cas d'une interaction partielle, un lien est créé du service source λ vers un service cible φ, s'il existe au moins une opération en interaction partielle entre les deux services.

A titre d'exemple, considérons l'ensemble des services représentés dans la figure 7. La partie gauche représente un ensemble de services Web nommés α, β et γ, leurs opérations numérotées de *1* à *4*, les paramètres en entrée et en sortie labélisés de *a* à *i*. La partie droite correspond aux deux types de réseaux d'interaction associés. Le réseau construit selon le mode d'interaction totale est représenté dans la partie supérieure (a). Le réseau construit selon le mode d'interaction partielle est représenté dans la partie inférieure (b).

Considérons le réseau en mode d'invocation totale, figure 7 (a). Le service α peut invoquer le service β en interaction totale au travers de l'opération *2*. Ceci se traduit donc par un lien dirigé du service α vers le service β. On ne crée pas de lien de β vers α car β ne peut invoquer aucune des opérations de α. Le seul mode d'interaction possible entre des opérations du service α et celles du service γ est une interaction partielle. En effet, α peut fournir une partie des paramètres pour invoquer l'opération *4* de γ à partir de l'opération *1*. Il n'y a donc pas de lien entre ces deux services. Le seul mode d'interaction possible entre β et γ est aussi une interaction partielle, l'opération *3* de β pouvant fournir une partie des paramètres pour invoquer l'opération *4* de γ. Il n'y a donc aucun lien entre ces deux services.

Considérons maintenant le réseau en mode d'invocation partielle, figure 7 (b). Le service α, peut invoquer le service β en interaction totale au travers de l'opération *2*. Néanmoins il n'existe aucune opération en interaction partielle entre ces deux services. Il n'y a donc pas de lien entre ces services. En ce qui concerne les interactions possibles entre α et γ, on remarque que α peut fournir une partie des paramètres pour invoquer l'opération *4* de γ à partir de

l'opération *1*. Il y a donc un lien dirigé de α vers γ entre ces deux services. Pour finir, entre β et γ il y a aussi une interaction partielle. En effet, l'opération *3* de β pout fournir une partie des paramètres pour invoquer l'opération *4* de γ. Il y a donc un lien entre ces deux services dirigé de β vers γ. Remarquons que ce type de réseau peut être construit en considérant un service Web comme une boîte noire car il suffit qu'un paramètre en sortie d'un service soit similaire à un paramètre d'entrée d'un autre service pour créer un lien entre ces services.

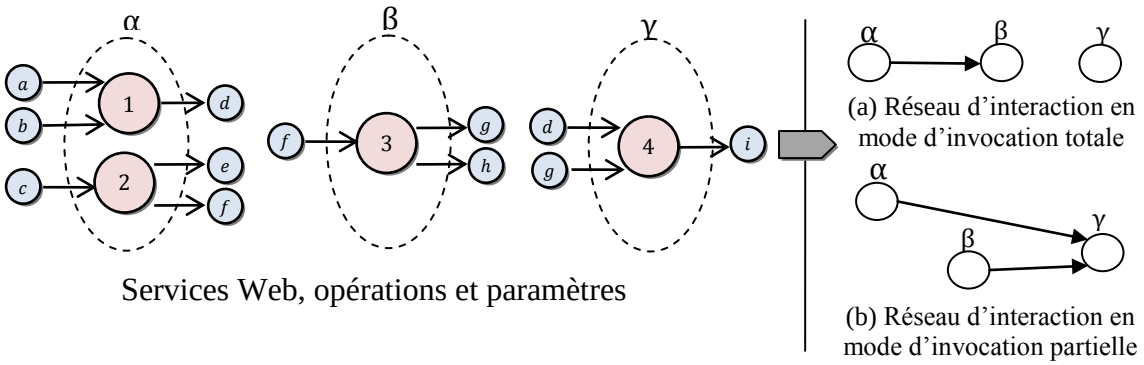


FIG. 7 - Réseau d'interaction de services à invocation totale (a) et réseau d'interaction de services à invocation partielle (b) issus des 3 services libellés α, β, et γ (Partie gauche).

Remarquons également que quelle que soit la granularité du réseau, les opérations sont au cœur des définitions. En effet le réseau des paramètres traduit les interactions entre paramètres d'entrée et de sortie d'une même opération. Les réseaux de services sont aussi définis à partir des opérations. Ils traduisent quant à eux, le fait que deux services peuvent interagir à travers deux opérations compatibles. Dans tous les cas, l'existence d'un lien est néanmoins subordonnée à la notion de similitude entre paramètres, similitude qui est traitée via les fonctions de mise en correspondance.

2.3.2 Fonctions de mise en correspondance

Dans le contexte de la modélisation des réseaux d'interaction, nous sommes dans le cas d'une correspondance horizontale. Autrement dit on doit évaluer la similitude entre les paramètres de sortie d'un service et les paramètres d'entrée du service qu'il désire invoquer. Nous définissons une fonction de mise en correspondance pour chacun des deux types de description (syntaxique, sémantique).

La mise en correspondance syntaxique concerne la description syntaxique des services. Elle repose sur la similitude des noms de paramètres. Les fonctions utilisées sont identiques à celles présentées dans le cadre des réseaux de similitude. Nous distinguons deux cas qui sont la similitude stricte et la similitude approximative. Pour la *similitude stricte*, deux paramètres sont dits similaires si leurs noms sont exactement les mêmes chaînes de caractères. Pour la *similitude approximative*, deux paramètres sont considérés comme similaires si les paramètres sont similaires au sens d'une certaine distance.

La mise en correspondance sémantique concerne la description sémantique des services. Elle repose quant à elle sur les notions de similitude entre concepts ontologiques. Bien que plus riche et plus précise que la description syntaxique, la description sémantique soulève de nouveaux problèmes. En effet, les descriptions sémantiques des services Web n'ont aucune raison d'être exprimées par référence aux mêmes ontologies. Il est donc nécessaire d'établir la correspondance entre des concepts appartenant à différentes ontologies. D'une façon générale cette problématique constitue un vaste champ d'investigation pour lequel on recense un grand nombre de travaux [53]. C'est un domaine prometteur pour la gestion de la diversité des sources d'information distribuées et de leur hétérogénéité. Dans le cadre de ce travail, nous nous limitons volontairement à une vision restrictive de la mise en correspondance sémantique qui suppose que pour décrire le même concept, tous les services utilisent la même ontologie. Autrement dit, deux concepts qui ne proviennent pas de la même ontologie ne peuvent pas être similaires. On peut néanmoins envisager d'utiliser toute autre fonction de mise en correspondance sémantique pour décider de la similitude entre concepts et ainsi décliner de nouveaux réseaux d'interaction à partir de ces fonctions alternatives.

Lorsque deux concepts appartiennent à la même ontologie, leur comparaison est facilement réalisée en exploitant la hiérarchie ontologique. Pour mesurer la similitude entre deux concepts nous utilisons l'opérateur *exact* tel que défini pour les réseaux de similitude, les opérateurs *plugin*, *subsume* et *fail* introduits dans [24] pour la découverte de services ainsi qu'un nouvel opérateur nommé *fitin*. Rappelons que l'on s'intéresse ici à la comparaison de services (ou d'opérations) à travers la mise en correspondance des concepts associés à leurs paramètres d'entrée et de sortie.

- Dans une relation *plugin*, deux paramètres sont dits similaires si le concept du paramètre en sortie est plus spécifique que le concept du paramètre en entrée. Autrement dit, le concept du paramètre en entrée subsume le concept du paramètre en sortie. Considérons le fragment d'ontologie figure 8 et deux services respectivement nommés `NiveauScolaire_ManuelBiologie` et `ManuelSecondaire_Prix`. Le premier service prend en entrée un niveau `secondaire` et fournit une liste de manuels de `biologie`. Le deuxième service prend en entrée tous les types de `manuels` scolaires et en fournit le `prix`. Le premier service peut entrer en interaction avec le second dans une relation plugin car le concept `manuel` est plus général que le concept `biologie`. Cependant, seule une partie des capacités du service `ManuelSecondaire_Prix` est utilisée. En effet, ce service est capable de fournir le tarif de tous les manuels secondaires et universitaires. Pour pouvoir utiliser la totalité de ces capacités, il faut faire appel à d'autres services en complément du service invoquant. Notons néanmoins que les buts du service invoquant sont pleinement satisfaits. Pour créer une interaction *plugin* entre deux opérations il faut que tous les paramètres concernés soient en relation *plugin*.

- Dans une relation *subsume*, deux paramètres sont dits similaires si le concept du paramètre en sortie est plus général que le concept du paramètre en entrée. Considérons le même fragment d'ontologie figure 8 et deux services définis comme suit. Le premier service nommé `NiveauSecondaire_ManuelTechnologie`

prend en entrée le niveau scolaire `secondaire` et fournit une liste de manuels de `technologie` (biologie, informatique, etc.). Le deuxième service nommé `ManuelInformatique_Prix` prend en entrée des manuels d'`informatique` et en fournit le `prix`. Le premier service peut entrer en interaction avec le second dans une relation subsume car le concept `informatique` est plus spécifique que le concept `technologie`. Cependant, la réponse obtenue, prix de manuels d'informatique, n'est qu'une réponse partielle. Dans ce cas, l'intégralité des buts du service invoquant ne peut être satisfaite. Pour obtenir les tarifs de tous les manuels secondaires de technologie, il faudrait envisager l'utilisation d'autres services en aval de l'interaction. Pour créer une interaction *subsume* entre deux opérations il faut que tous les paramètres concernés soient en relation *subsume*.

- Dans une relation *fitin* deux paramètres sont dits similaires s'il existe une relation d'équivalence entres eux ou si le concept du paramètre en sortie est plus spécifique que le concept du paramètre en entrée. Cette définition englobe ainsi la relation *exact* et la relation *plugin*. Elle étend la relation *exact* présentée dans les travaux de Paolucci. Elle suppose que tout super concept s'engage à fournir tous les sous concepts. La définition de Paolucci quant à elle ne concernait que le niveau immédiatement supérieur de la hiérarchie. Cette nouvelle définition suppose ainsi une utilisation plus rigoureuse de l'ontologie. En ce qui concerne la satisfaction des buts du service invoquant, on se retrouve dans la même situation que dans le cas *plugin* à savoir que ses buts sont pleinement satisfaits. En ce qui concerne le service invoqué, il est sous utilisé dans cette interaction. Pour créer une interaction *fitin* entre deux opérations il faut que tous les paramètres concernés soient en relation *fitin*. Autrement dit, les paramètres peuvent être soit en relation *exact*, soit en relation *plugin*.

- Enfin, la relation *fail* signifie qu'il n'y a aucune relation de subsomption entre les concepts. Soient les deux services suivants décrits sur la base de l'ontologie figure 8. Le premier service `NiveauSecondaire_ManuelSecondaire` fournit tout type de `manuels` secondaires à partir du niveau fourni. Le deuxième service `ManuelUniversitaire_Tarif` fournit les `tarifs` de manuels `universitaires`. Dans cette situation, aucune interaction entre ces deux services n'est possible. Lorsque tous les paramètres concernés sont en relation *fail*, il n'y a pas d'interaction entre deux opérations.

Les relations exact et fail sont des relations symétriques. Les relations plugin et subsume sont des relations asymétriques. fitin comprend une relation symétrique et une relation asymétrique. Nous récapitulons dans le tableau 7, la signification des opérateurs sémantiques.

Remarquons là aussi, que l'on peut substituer aux fonctions de mise en correspondance que nous venons de préciser, toute fonction de similitude entre concept basée sur la structure de l'ontologie ou sur le contenu informatif entre concepts.

TAB. 7 – Définition des opérateurs de la fonction de mise en correspondance sémantique.

Opérateur	Relation
exact	*conceptEntrée* ***équivalent*** *conceptSortie*
plugin	*conceptEntrée* ***subsume*** *conceptSortie*
subsume	*conceptSortie* ***subsume*** *conceptEntrée*
fitin	*conceptSortie* ***équivalent*** *conceptEntrée* OU *conceptEntrée* ***subsume*** *conceptSortie*
fail	***pas de relation de subsomption entre*** *conceptEntré* **et** *conceptSortie*

Ces opérateurs peuvent être ordonnés selon un degré de pertinence comme suit : *exact* > *fitin* > *plugin* > *subsume* > *fail*.
Une correspondance exact sera en effet la meilleure adéquation possible entre un paramètre en sortie d'une opération et un paramètre en entrée d'une autre opération. La relation fitin est préférable à la relation plugin car elle permet d'avoir à la fois des relations plugin et exact au sein d'une même interaction. Enfin, la relation subsume permet de récupérer des interactions moins pertinentes qu'avec les opérateurs précédents.

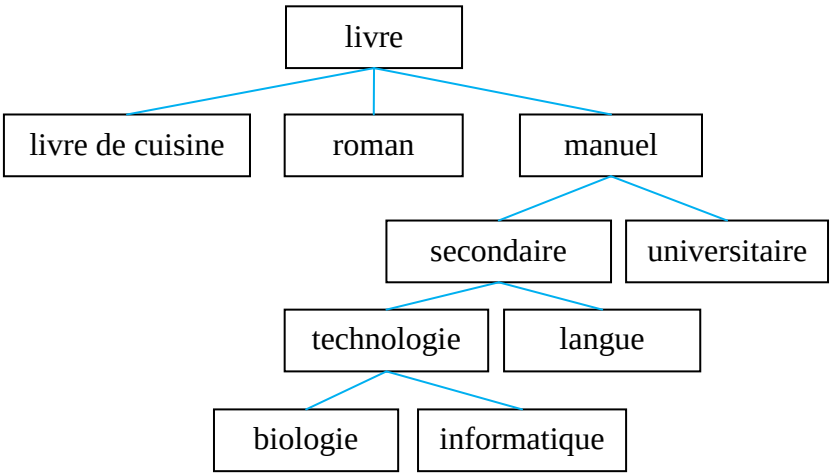


FIG. 8 – Fragment d'ontologie relative aux livres.

2.4 Conclusion

Dans ce chapitre, nous avons présenté les modèles de similitude et d'interaction qui peuvent être utilisés pour représenter les ensembles de services Web. Dans les réseaux de similitude, les nœuds représentent des opérations alors que dans les réseaux d'interaction ceux-ci peuvent représenter des paramètres, des opérations ou des services. La définition de tous ces modèles fait appel à une fonction de mise en correspondance dont le but est de déterminer la similitude entre deux paramètres. Nous avons présenté les fonctions de similitude que nous nous proposons d'utiliser dans la suite de ces travaux aussi bien pour les services dont la description est syntaxique que sémantique. Nous tenons néanmoins à préciser que d'autres fonctions de similitude plus élaborées peuvent être utilisées sans nuire à la généralité des

modèles proposés. En ce qui concerne les réseaux d'interaction d'opérations et de services, nous avons considéré en outre un autre paramètre permettant de différencier les réseaux, à savoir le mode d'invocation. On distingue le mode d'invocation totale du mode d'invocation partielle. Dans le premier cas, toute l'information nécessaire est fournie pour invoquer une opération (ou une quelconque opération d'un service). Dans le second cas, l'opération ou le service invoquant ne peut fournir qu'une partie de l'information.

Ces modèles permettent de résumer l'essentiel des relations que peuvent entretenir des services Web entre eux par l'intermédiaire de leurs opérations et de leurs paramètres. Les réseaux issus de l'instanciation de ces modèles peuvent être vus comme des objets utilisables à différentes fins. Ils peuvent, entre autres, permettre d'obtenir une vision topologique d'un ensemble de services Web à un moment donné.

3. RESEAUX COMPLEXES

3.1 Introduction

Alors que la théorie des graphes englobe les résultats fondamentaux sur les graphes, la théorie des réseaux s'intéresse aux graphes présents dans le monde réel, c'est-à-dire aux systèmes réels qui peuvent être représentés sous forme de grands graphes. En raison de sa simplicité et de sa traduction graphique intuitive, la représentation sous forme de graphe est devenue extrêmement utilisée dans des domaines d'application très variés. Ainsi, l'étude des réseaux complexes est aujourd'hui un sujet de tout premier ordre dans de nombreuses disciplines, y compris la physique, la biologie, l'informatique et les sciences sociales. De nombreuses et récentes études témoignent de cette activité [54] [55] [56] [57]. Cet attrait croissant est en partie dû à la découverte que les grands graphes de terrain (réseaux du monde réel), bien que de natures totalement différentes, peuvent partager de grandes similitudes dans leurs propriétés topologiques.

La « science des réseaux » est une discipline scientifique émergente qui se développe parallèlement à partir de disciplines scientifiques très diverses. C'est la raison pour laquelle on peut en trouver plusieurs définitions. Ainsi Watts [58] définit la science des réseaux comme la science du monde réel, le monde des gens, de l'amitié, des rumeurs, des maladies, des entreprises et des crises financières. Le National Research Council définit la science des réseaux comme l'étude des représentations en réseaux des phénomènes physiques, biologiques et sociaux, en vue d'élaborer des modèles prédictifs de ces phénomènes. Se situant dans une perspective informatique, Brandes parle quant à lui de théorie des graphes appliquée [59]. Zaidi [60] propose d'étendre cette définition à l'étude de la théorie, des méthodes et algorithmes applicables aux modèles de graphes représentant des systèmes connectés du monde réel.
Le ciment des travaux en la matière est d'établir des outils pour mieux appréhender le comportement de ces réseaux qui, au-delà des particularités de chaque domaine, présentent des motifs et des régularités statistiques communes.

3.1.1 Domaines d'application

Dans le contexte de la théorie des réseaux, un réseau complexe est un grand graphe ayant une structure irrégulière, complexe et évoluant de façon dynamique dans le temps. Il exhibe des caractéristiques topologiques non triviales, c'est-à-dire que l'on ne voit pas apparaître dans des réseaux simples comme les grilles ou les réseaux aléatoires. Cette définition ouvre un vaste champ d'applications. Les réseaux sociaux, les réseaux d'information, les réseaux technologiques ou encore les réseaux biologiques constituent néanmoins les domaines d'investigation prépondérants de l'analyse des réseaux complexes.

> Les **réseaux sociaux** constituent un champ d'application privilégié grâce à l'essor des applications communautaires. Ils font depuis longtemps l'objet de nombreuses recherches en sociologie. Un réseau social est généralement défini comme un

ensemble de relations d'un type spécifique (par ex. de collaboration, de soutien, de conseil, de contrôle ou d'influence) entre un ensemble d'acteurs. Les acteurs peuvent être des individus ou des entités sociales comme des associations, des entreprises, des gouvernements etc. Les acteurs représentent ainsi les nœuds du réseau et les relations sont représentées par des liens. On peut citer à titre d'exemple, les réseaux de connaissance où deux individus sont reliés s'ils se connaissent, les réseaux de contact physique permettent de relier deux individus s'ils ont été physiquement en contact, les réseaux de collaboration qui permettent de relier des individus qui ont travaillé ensemble. La théorie des réseaux est utilisée aujourd'hui dans le domaine social pour mieux comprendre les interactions entre des groupes sociaux et leur environnement. De nombreux travaux ont par exemple étudié les collaborations scientifiques [61], afin de mieux appréhender l'évolution et le développement des domaines de recherche.

Les **réseaux d'information**, aussi appelés réseaux de connaissances, forment une seconde catégorie de réseaux complexes. Cette catégorie de réseaux permet de représenter des liens abstraits de référencement entre des supports d'information. Parmi eux, un exemple classique est celui de citation d'articles scientifiques. C'est un réseau orienté dans lequel les nœuds du réseau sont les articles. On crée un lien de l'article qui cite vers l'article cité. Les réseaux de citation constituent un bon sujet d'étude en raison de la précision et de l'abondance des données. Un exemple tout aussi important de réseau d'information est le World Wide Web. Les sommets sont les pages Web et les liens représentent des liens hypertextes. On établit un lien entre la page d'origine et la page référencée.

Les **réseaux technologiques** ou réseaux d'infrastructure, qui sont des réseaux créés par l'homme, représentent des connections matérielles entre objets distribués dans un espace géographique. Ils désignent typiquement la distribution de ressources. Dans cette catégorie, citons les réseaux de distribution électrique où les liens représentent les câbles entre les lieux de production et de consommation, les réseaux de transport avec les liaisons aériennes ou les voies terrestres. Internet est dans cette catégorie l'un des réseaux les plus étudiés principalement pour sa dynamique. Dans ce réseau, les nœuds représentent des routeurs et les liens des liaisons physiques entre routeurs. Une analyse de ce type de réseau peut par exemple aider à identifier les points de rupture. Ceci permet de mettre en place une stratégie de protection en proposant des chemins alternatifs pour véhiculer les ressources en cas de défaillance d'un nœud.

Les **réseaux biologiques** permettent de représenter les nombreuses relations intervenant dans le monde du vivant. Ainsi le métabolisme d'une cellule est un système complexe qui met en jeu un ensemble de réactions métaboliques. Ces réactions, qui sont généralement catalysées par des enzymes et transforment des métabolites substrats en métabolites produits, sont représentées par un réseau. Les nœuds du réseau sont les substrats et les produits. Un lien orienté du substrat vers le produit est créé s'il existe une réaction métabolique qui agit sur un substrat pour donner un produit. Cette modélisation permet de réaliser des requêtes complexes comme, par exemple, le calcul (et la prédiction) de tous les métabolites pouvant être générés à partir d'un ensemble de composés sources. Ces approches permettent

également d'identifier des réactions pouvant être considérées comme des cibles thérapeutiques potentielles. Par ailleurs elles sont susceptibles d'apporter des éléments de réponse quant à la perturbation des systèmes biologiques (cellule, tissu, organisme) par des faibles doses et/ou des mélanges de contaminants alimentaires. On peut aussi citer les réseaux de protéines, les réseaux de gènes ainsi que les réseaux de la chaîne alimentaire avec les relations proie-prédateurs entre espèces d'un écosystème, qui sont également très étudiés.

3.1.2 Axes d'études

Pour analyser et comprendre l'organisation de ces grands graphes de terrain, les différents domaines d'application partagent les mêmes fondements méthodologiques. Ainsi, la science des réseaux a établi des méthodes spécifiques pour connaître les propriétés et comprendre le comportement des réseaux complexes. On peut distinguer quatre axes majeurs de développement en ce qui concernent les travaux sur les réseaux complexes :

La **définition d'outils d'analyse** qui consiste à mettre au point des mesures qui permettent d'étudier les propriétés statistiques des réseaux complexes. On peut distinguer trois niveaux de granularité dans ces travaux : les réseaux dans leur globalité, les communautés ou sous-ensembles structurés et les nœuds.

La **définition d'outils de génération de réseaux** qui englobe des travaux visant à mettre au point des algorithmes de génération de réseaux artificiels qui possèdent des propriétés topologiques particulières afin de mieux comprendre la structure et l'évolution de tels réseaux.

La **définition d'outils de traitement** pour la recherche de chemins, de motifs, de groupes en vue d'organiser et de comprendre le comportement statique et dynamique des réseaux. A cet égard, la découverte de communautés et l'étude des processus de diffusion sont des problématiques majeures.

La **définition d'outils de visualisation** qui vise à l'exploration visuelle des réseaux complexes afin d'inférer de nouvelles connaissances.

Bien entendu, bien que ces quatre axes répondent à des objectifs bien différents, dans la réalité ils se recouvrent et s'interpénètrent de sorte qu'il est difficile de les séparer. Dans ce qui suit, nous introduisons les définitions des mesures de base pour caractériser les propriétés structurelles d'un réseau. Ensuite nous présentons les propriétés communément observées dans les réseaux réels. En dernier lieu nous donnons les caractéristiques topologiques de quelques exemples de grands graphes de terrain.

3.2 Propriétés topologiques fondamentales

3.2.1 Définitions de base

La théorie des graphes est le cadre naturel pour le traitement mathématique exact des réseaux complexes. Formellement, un réseau complexe peut être représenté comme un graphe. Un graphe est une représentation abstraite pour modéliser des relations de pair à pair entre des objets d'une collection. La structure mathématique est simple. Les objets sont habituellement représentés par des cercles appelés *nœuds* ou *sommets*. Les relations entre les objets sont représentées par des lignes appelées *arêtes* dans un réseau non-orienté et *arcs* dans un réseau orienté. Nous donnons ci-après les définitions formelles correspondantes.

> Un ***graphe non-orienté*** *G = (V, E)* est composé d'un ensemble *V* de sommets (ou nœuds) et d'un ensemble *E* de paires de sommets nommées arêtes.
>
> Un ***graphe orienté*** *G = (V, E)* est composé d'un ensemble *V* de sommets (ou nœuds) et d'un ensemble *E* de paires de sommets nommées arcs.
>
> Nous noterons *n* le nombre de sommets ($n = |V|$) et m le nombre d'arêtes ou d'arcs ($m = |E|$) dans un graphe. *n* est la taille du réseau.

Formellement, un graphe est étiqueté c'est-à-dire que chaque sommet ou chaque arête/arc appartient à un ensemble, donc porte une étiquette. Typiquement, les graphes sont étiquetés par des nombres entiers, mais une étiquette peut appartenir à n'importe quel ensemble comme l'ensemble de couleurs, l'ensemble de mots, l'ensemble des réels etc. L'étiquetage d'un graphe peut être conçu de façon à donner des informations utiles pour des problèmes comme le routage. Dans cet exemple, partant d'un sommet *u*, on veut arriver à un sommet *v*, c'est-à-dire que l'on souhaite acheminer une information de *u* à *v*.
La figure 9 représente deux réseaux de taille *25* étiquetés par des entiers, un réseau non orienté à gauche et un réseau orienté à droite.

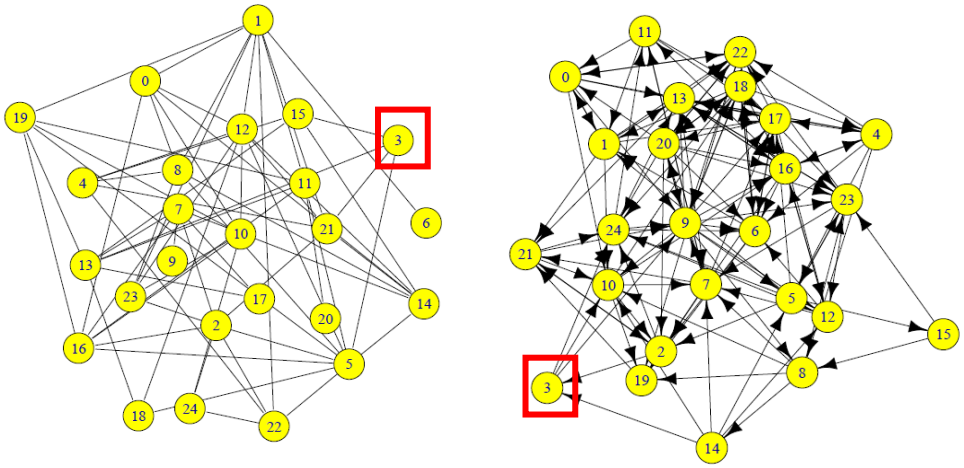


FIG. 9 - Un réseau non orienté (Gauche) et un réseau orienté (Droite).

Un graphe peut être représenté par une matrice carrée dite d'adjacence A de la taille du réseau dont les éléments sont donnés par:

$$A(i,j) = \begin{cases} 1 \text{ si il y a un arc ou une arête entre } i \text{ et } j \\ 0 \text{ sinon} \end{cases}$$

où i et j représentent les labels des nœuds. Dans les réseaux non orientés, la matrice d'adjacence est symétrique.

La matrice Laplacienne L est une autre représentation d'un graphe. Elle est définie par :

$$L = K - A$$

Ou K est une matrice diagonale dont les éléments correspondent au *degré* d'un nœud. Le degré d'un nœud fait référence au nombre de connexions que ce nœud possède dans le réseau.

3.2.2 Propriétés Structurelles

Propriétés basées sur le degré

On notera k_i le degré d'un nœud i. Pour un réseau orienté, on peut distinguer le degré entrant, le degré sortant et le degré total d'un nœud. Le degré entrant est le nombre de liens incidents (notons que nous utilisons le terme « lien » pour désigner indifféremment une arête ou un arc, le contexte permettant de faire la distinction si nécessaire). Le degré sortant correspond au nombre de liens émanant du nœud considéré. Le degré total est la somme des degrés entrants et sortants. Ainsi sur la figure 9, le nœud 3 sur le réseau non orienté de gauche est de degré *3*. Le nœud de même label sur le réseau orienté de droite à un degré entrant de 2, un degré sortant de 3 et un degré total de 5. Les nœuds qui ne possèdent pas de liens sont dits isolés.

Le *degré moyen* représente la valeur moyenne des degrés de l'ensemble du réseau dans un réseau non orienté. Pour un réseau de taille n, il peut être estimé à partir de la relation suivante :

$$\langle k \rangle = \frac{1}{n} \sum_{i=1}^{n} k_i$$

De la même façon, pour un réseau orienté on peut définir le *degré moyen entrant* $\langle k_{in} \rangle$, le *degré moyen sortant* $\langle k_{out} \rangle$ *et* le *degré moyen global* $\langle k_{all} \rangle$.

La *distribution des degrés* $P(k)$ mesure la probabilité qu'un nœud quelconque d'un réseau soit de degré *k*. Elle peut être estimée à partir de la fraction de nœuds de degré *k* dans un réseau. Elle détermine complètement les propriétés statistiques des réseaux non corrélés. Autrement dit les réseaux pour lesquels le degré d'un nœud est indépendant du degré de son environnement. Dans la pratique, elle est représentée par l'histogramme des degrés qui comptabilise pour chacune des valeurs de degré existant la proportion de nœuds qui partagent

cette valeur. Pour les réseaux orientés on peut distinguer la distribution des degrés entrants et la distribution de degrés sortants [62].

La *corrélation des degrés* révèle la façon dont le degré d'un nœud est lié à son environnement. Elle traduit le fait que des « attachements » plus ou moins préférentiels peuvent exister entre nœuds du réseau. Elle est formellement définie à partir de la probabilité conditionnelle $P(k'|k)$ pour qu'un nœud de degré k soit connecté à un nœud de degré k'. Si les nœuds de fort degré sont préférentiellement reliés à d'autres nœuds de fort degré, la corrélation des degrés est dite assortative. Si les nœuds de fort degré sont préférentiellement liés à des nœuds de faible degré, la corrélation des degrés est dite disassortative. La corrélation des degrés est généralement mesurée par le coefficient de corrélation de Pearson [63]. Lorsque la valeur du coefficient est de 1, le réseau est parfaitement assortatif. Avec un coefficient de valeur -1, le réseau est complètement disassortatif.
Les réseaux réels présentent habituellement un degré de corrélation significativement différent de zéro [56]. Selon Newman, les réseaux sociaux ont tendance à être assortatifs, tandis que d'autres types de réseaux sont généralement disassortatifs.
Les *hubs* et les *autorités* représentent des nœuds importants dans les réseaux orientés. Les *hubs* sont les nœuds qui ont un degré sortant supérieur à la moyenne. Quant aux *autorités* elles possèdent un degré entrant supérieur à la moyenne. Dans les réseaux non orientés, ces deux définitions sont confondues.

Propriétés basées sur les distances

Dans un graphe orienté, un *chemin* d'origine x et d'extrémité y est défini par une suite finie d'arcs consécutifs, reliant x à y. La notion correspondante dans les graphes non orientés est celle de *chaîne*. Un *chemin élémentaire* est un chemin ne passant pas deux fois par un même sommet, c'est-à-dire dont tous les sommets sont distincts. Un *chemin simple* est un chemin ne passant pas deux fois par un même arc, c'est-à-dire dont tous les arcs sont distincts. Par la suite on appellera « *chemin* » un chemin simple et élémentaire. La *longueur* d'un chemin est le nombre de lien du chemin. Les notions de chemins dans un graphe non orienté et dans un graphe orienté sont illustrées par la figure 10.

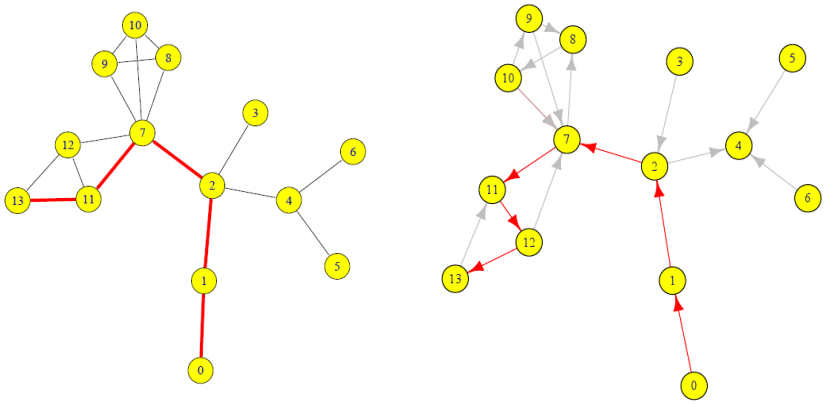


FIG. 10 - Un chemin dans un réseau non orienté de longueur de 5 (Gauche). Un chemin dans un réseau orienté de longueur de 6 (Droite).

La *distance* d_{ij} entre deux nœuds *i* et *j* d'un réseau est définie comme le nombre de liens sur le plus court chemin les reliant. Quand il n'y a pas de chemin entre les deux nœuds considérés, la distance est théoriquement infinie. Cependant, dans de nombreuses situations, ceci cause des problèmes, et cette distance est alors considérée comme nulle.

Le *diamètre D* d'un réseau est le maximum de l'ensemble des *distances* entre tous les sommets du réseau pris deux à deux. C'est la distance maximale pour joindre deux nœuds quelconques du réseau.

$$D = max\left\{d_{ij}\right\}$$

La *distance moyenne* est la valeur moyenne arithmétique de toutes les distances entre deux nœuds quelconques du réseau. Elle est exprimée par la formule suivante :

$$L = \frac{1}{n\,(n-1)} \sum_{i \neq j} d_{ij}$$

où d_{ij} est la distance entre les nœuds *i* et *j*. S'il y a des nœuds non connectés, cette équation diverge. Afin de surmonter ce problème, seuls les nœuds connectés peuvent être inclus dans la somme. Une autre façon de procéder consiste à considérer la moyenne harmonique des distance et de calculer l'efficacité du graphe donnée par :

$$E = \frac{1}{n\,(n-1)} \sum_{i \neq j} 1/d_{ij}$$

Transitivité

Cette propriété est aussi appelée fraction de triangles dans le réseau ou coefficient de clustering. Nous préférerons utiliser le terme de transitivité [56] afin d'éviter toute confusion avec le concept de structure de communauté ou cluster [64] [59]. La *transitivité* correspond donc à la densité de triangles d'un réseau, un triangle étant une structure de trois nœuds complètement connectés. Elle est donnée par la formule suivante :

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

Le facteur 3 au numérateur contrebalance le fait que chaque triangle contribue à trois triplets et contraint la valeur du coefficient à l'intervalle [0,1]. En d'autres termes, le coefficient de transitivité globale est une estimée de la probabilité moyenne que deux sommets qui sont voisins d'un troisième soient eux-mêmes voisins. Cette valeur donne une vue d'ensemble de la présence de triades dans un réseau. Notons qu'il existe également une définition locale du coefficient de transitivité. Dans ce cas le coefficient s'applique à un nœud relativement à son environnement immédiat.

Composante

Un graphe est *connexe* s'il existe un chemin entre tout couple de sommets. Quand on parle de connexité pour un graphe orienté, on considère non pas ce graphe mais le graphe non-orienté correspondant. Une composante d'un graphe est un sous-graphe connexe maximal. Un graphe orienté est dit fortement connexe si, pour tout couple de sommets *(i, j)* du graphe il existe un chemin de *i* à *j* et de *j* à *i*. La figure 11 représente un graphe à 4 composantes.

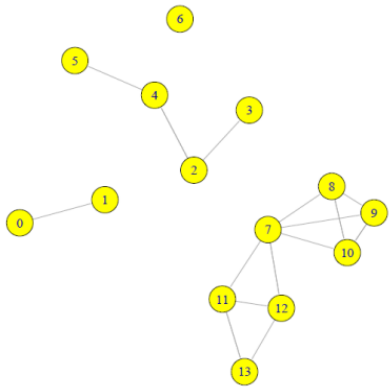


FIG. 11 - Un réseau à 4 composantes. Le nœud 6 est isolé.

Densité

La *densité* d'un réseau de taille *n* est définie comme la proportion de liens existant par rapport au nombre de liens possibles. Soit *m* le nombre de liens, existant la densité est :

$$\boldsymbol{d = \frac{m}{n\,(n-1)}}$$

Clique

Une clique est sous-graphe induit complet, c'est-à-dire un sous-ensemble des sommets tels que chacun est connecté à tous les autres. Une *p-clique* désigne une clique contenant *p* nœuds. La figure 12 représente un réseau qui contient huit 1-clique (ses nœuds), quinze 2-cliques (ses liens), huit 3-cliques, cinq 4-cliques et une 5-clique figurée en rouge.

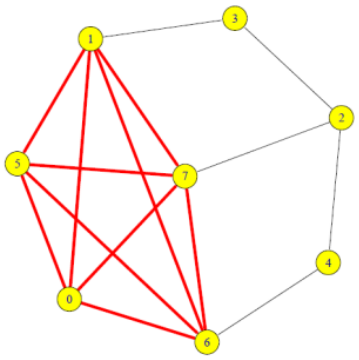


FIG. 12 - Un graphe à 8 nœuds avec en rouge une 5-clique.

3.3 Propriétés communes aux grands graphes de terrain

Deux classes importantes de réseaux émergent, les réseaux «petit monde» et les réseaux «sans échelle». Un réseau petit monde est caractérisé par une petite distance moyenne. Un réseau sans échelle est un réseau où la distribution des degrés suit une loi de puissance.

3.3.1 Propriété Petit monde

L'étude de nombreux réseaux a mis en évidence la présence de raccourcis qui connectent différentes parties du réseau réduisant ainsi la distance entre des nœuds qui seraient sans cela relativement éloignés. Ainsi, un réseau petit monde est un réseau dans lequel la plupart des nœuds peuvent être atteints de tous les autres par un petit nombre de sauts. Ceci se traduit par une petite distance moyenne. Cette valeur croît de façon logarithmique avec la taille du réseau. Lorsque la distance moyenne est petite dans les réseaux, on dit qu'ils possèdent la propriété *petit monde*.
La première manifestation de cette propriété a été observée dans le cadre des réseaux sociaux par Milgram [65]. Cette expérience consistait à demander aux participants de passer une lettre à une de leur connaissance dans le but de la faire parvenir à un individu cible. Le courrier devait être transmis à leur connaissance qui résidait le plus près de la destination finale jusqu'à atteindre sa destination. Les résultats de cette expérience ont montré que le courrier atteint sa cible en transitant par en moyenne six personnes. Cette expérience renvoie à l'idée que tout le monde est en moyenne à six étapes de toute autre personne sur terre, de sorte qu'une chaîne relativement courte de "l'ami d'un ami" peut être faite pour relier deux personnes. Cette propriété peut être interprétée comme l'efficacité de propagation dans un réseau. Elle a été observée dans de nombreux grands graphes de terrain. Le Web constitue par exemple un réseau petit monde. Sa distance moyenne est 18,59 pour 8.10^8 nœuds [66]. C'est aussi une propriété de modèles de réseaux comme par exemple les réseaux aléatoires. La figure 13 permet d'illustrer cette propriété. Elle représente deux échantillons de graphes aléatoires d'Erdös-Rényi contenant 20 nœuds. Pour générer ces graphes on utilise la procédure suivante. Pour toutes les paires de nœuds, on crée un lien avec une probabilité *p*. Autrement dit on tire une variable aléatoire uniforme dans [0,1] et on crée un lien si la réalisation de la variable est inférieure ou égale à *p*. On voit bien que des connexions apparaissent autant pour des nœuds voisins que pour des nœuds éloignés et ceci indépendamment de la valeur de *p*.

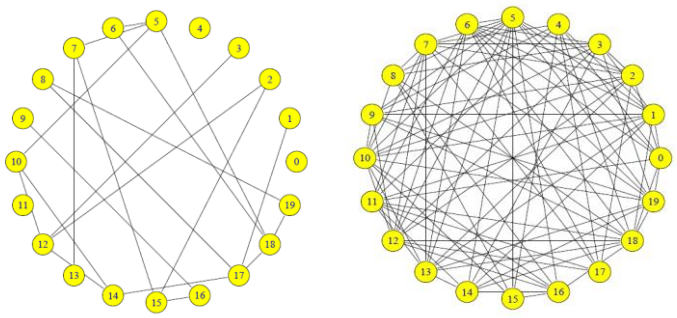


FIG. 13 - Réseaux aléatoires d'Erdös-Rényi comprenant 20 nœuds. Le réseau de gauche est créé avec une probabilité p = 0.1. Le réseau de droite est créé avec une probabilité p = 0.5.

Dans les grands graphes de terrain, une petite distance moyenne est souvent associée à un coefficient de transitivité élevé. Un coefficient de transitivité élevé a d'abord été observé dans les réseaux sociaux. Il traduit le fait que deux personnes qui ont toutes deux des liens avec une troisième personne ont de grandes chances d'avoir des relations directes entres elles. En d'autres termes, la probabilité que l'on devienne ami avec une personne avec laquelle on a des connaissances en commun est assez élevée. De nombreux autres réseaux suivent cette tendance comme le World Wide Web [67] ou encore les réseaux de transport [68] [69]. Plus la transitivité est élevée, plus il est probable d'observer un lien entre deux nœuds qui sont tous deux connectés à un troisième. La présence de triades est ainsi une caractéristique importante dans la plupart des systèmes réels. Un exemple typique est celui du réseau de la collaboration scientifique. Supposons qu'un chercheur *a* collabore séparément avec les chercheurs *b* et *c*. Il est probable que les domaines de recherche de *b* et de *c* soient les mêmes et il y a de fortes chances que *b* et *c* collaborent également. En ce sens ces réseaux se distinguent des réseaux aléatoires caractérisés par une très faible transitivité toutes choses étant égales (taille et nombre d'arêtes du même ordre de grandeur). En effet, dans les réseaux aléatoires les arêtes sont distribuées aléatoirement et la présence de triplets transitifs est rare. Ainsi sur la figure 13, on peut remarquer que le graphe généré avec $p = 0.1$ ne possède aucune triade. Une forte valeur de transitivité est plus caractéristique des réseaux réguliers. Les graphes aléatoires sont ainsi caractérisés par une petite distance moyenne et une faible valeur de transitivité alors que de nombreux grands graphes de terrain sont caractérisés par une petite distance moyenne et une forte transitivité. C'est la raison pour laquelle Watts et Strogatz [69] ont proposé de définir un réseau « petit monde » comme un réseau caractérisé par une petite distance moyenne et un coefficient de transitivité élevé.

Remarquons que ces deux propriétés sont complémentaires. La distance moyenne est une mesure de l'efficacité globale de transmission de l'information entre deux nœuds quelconques du réseau alors que la transitivité mesure plutôt l'efficacité locale. Dans ce qui suit, nous associons la propriété « petit monde » à la notion de petite distance moyenne uniquement.

L'information sur la propriété petit monde des réseaux peut répondre à des préoccupations bien différentes. Ainsi, dans les réseaux sociaux, elle peut permettre d'agir sur la propagation des épidémies [70] [71]. Elle peut aussi permettre d'élaborer des stratégies de marketing pour cibler des consommateurs [72]. Sur Internet, elle permet d'estimer le nombre de sauts nécessaires à un paquet pour aller d'un réseau à un autre et d'utiliser cette information pour améliorer l'efficacité des applications distribuées [73].

3.3.2 Réseau Sans échelle

La distribution des degrés est une donnée importante pour l'analyse des réseaux. Elle est particulièrement révélatrice d'une structure spécifique et a des conséquences importantes pour la compréhension des phénomènes naturels et artificiels. On pourrait penser que la distribution des degrés d'un graphe soit relativement homogène, autrement dit que tous les nœuds possèdent un degré qui varie très peu autour de la valeur moyenne des degrés. Cette hypothèse est contredite par bon nombre d'études expérimentales sur les grands graphes de

terrain. La plupart des réseaux dans le monde réel, notamment l'Internet, le World Wide Web et certains réseaux sociaux, ont une distribution des degrés qui est très différente de la distribution binomiale d'un graphe aléatoire d'Erdös-Reyni. La distribution des degrés observée est fortement inhomogène et caractérisée par une forme asymétrique décalée vers la droite. Autrement dit, une grande majorité des nœuds présente un degré faible et un petit nombre de nœuds présente un degré élevé [62]. Cette distribution peut être approximée par une loi de puissance de la forme :

$$p_k \approx ck^{-\gamma}$$

La figure 14 représente une distribution en loi de puissance. Elle est souvent représentée sur un graphique aux échelles logarithmiques. Dans ce cas, le graphe d'une loi de puissance est une droite. En effet, la relation ci-dessus peut s'écrire :

$$\log(p_k) \approx \log(c) - \gamma \log(k \quad)$$

Dans une représentation logarithmique, la pente de la droite permet ainsi d'estimer la valeur de l'exposant γ.

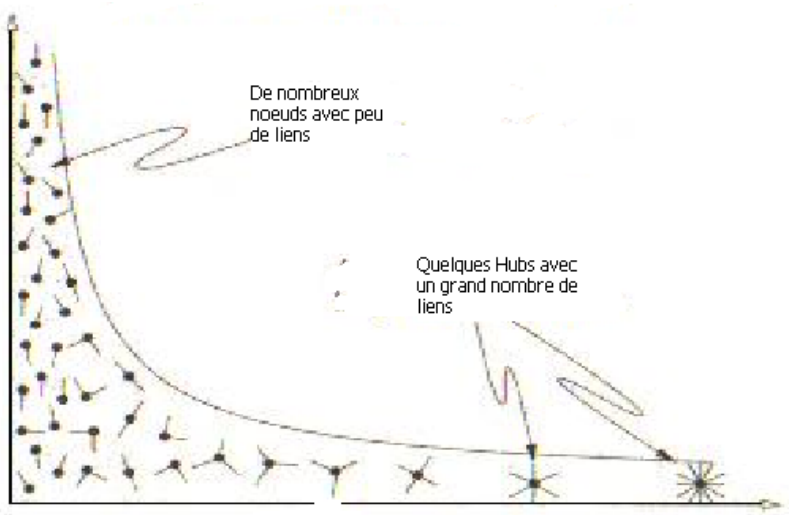


FIG. 14 – Illustration de la distribution des degrés en loi de puissance.
Extrait de http://www.macs.hw.ac.uk/~pdw/topology/ScaleFree.html.

Beaucoup de phénomènes naturels ou artificiels tels que la taille des villes, la répartition des richesses, l'ampleur des catastrophes naturelles sont répartis selon une distribution en loi de puissance. Ceci implique que les occurrences de faible valeur sont extrêmement communes alors que les occurrences de grande valeur sont extrêmement rares. Ce comportement général est parfois désigné sous le vocable « loi de Pareto » ou loi des 80/20 lorsque qu'on se réfère à la distribution des revenus, et parfois aussi à la loi de Zipf qui elle fait référence à l'utilisation du vocabulaire dans la langue anglaise. Ce qui diffère essentiellement entre ces lois, c'est la valeur de l'exposant γ. Les études expérimentales montrent que la valeur du coefficient γ varie généralement entre 2 et 3 pour les grands graphes de terrain qui présentent cette propriété.

Notons que pour les réseaux orientés, on définit la distribution des degrés entrants, la distribution des degrés sortants et la distribution globale qui correspond au réseau non orienté. Les exposants des lois de puissance correspondant sont respectivement γ_{in}, γ_{out}, γ_{all}.
On utilise la terminologie « réseau sans échelle » car la distribution en loi de puissance est caractérisée par une invariance d'échelle. Autrement dit, la distribution observée se conserve quelle que soit la taille du réseau. Les réseaux sans échelle sont ainsi caractérisés par la présence de hubs reliés à une majorité de nœuds faiblement connectés. Pour ces réseaux, la notion de nœud typique n'a donc pas de sens car les fluctuations de la connectivité d'un nœud à un autre sont importantes.

Une des toutes premières études où la distribution en loi de puissance a été mise en évidence concerne les réseaux de citations d'articles scientifiques [74]. On remarque dans ce cas un petit nombre « d'autorités scientifiques » très souvent citées parmi une multitude de travaux faiblement cités. L'auteur fut le premier à proposer un mécanisme pour expliquer l'apparition de lois de puissance dans les réseaux de citations, qu'il a appelé un avantage cumulatif [75]. Cette propriété a été aussi identifiée pour Internet dans [76]. Les auteurs ont également proposé un mécanisme pour expliquer l'apparition de cette propriété qu'ils ont appelée l'attachement préférentiel et qui est essentiellement le même que celui proposé par Price. Si un nœud a un degré élevé, il a une probabilité plus élevée pour attirer davantage de connexions et donc sa connectivité croît à un rythme plus rapide que les nœuds avec une connectivité faible. Un autre terme couramment utilisé pour désigner ce principe est « les riches s'enrichissent plus » (The rich get richer).
La figure 15 présente des échantillons de réseaux obtenus à l'aide de générateurs qui permettent d'approximer la distribution en loi de puissance des degrés. On voit très clairement apparaître des hubs. Ainsi, dans le réseau de gauche, le nœud *36* est fortement connecté. Dans le graphe de droite la présence de hub est encore plus caractéristique.

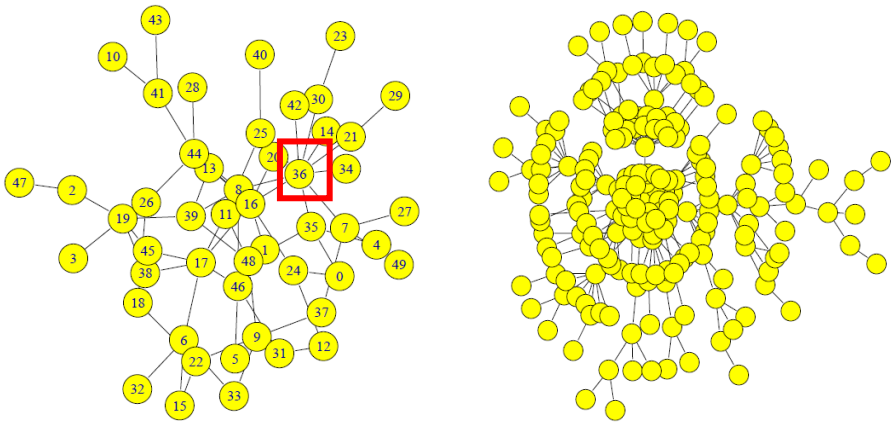


FIG. 15 - Réseau sans échelle de taille n = 50 généré avec le modèle de Bender et Canfield [77] (Gauche). Réseau sans échelle de taille n = 200 généré avec le modèle BA de Barabàsi-Albert [76] (Droite).

Cette information sur le degré des nœuds a des implications importantes dans l'étude des réseaux. Par exemple, dans un réseau social, les hubs jouent un rôle important pour la diffusion de l'information étant donné qu'ils représentent des personnes ayant de nombreux

liens sociaux. De nombreuses stratégies peuvent ainsi être développées en tirant parti de l'existence des hubs.

3.3.3 Organisation en composantes

L'étude de l'organisation en composantes se concentre généralement sur la taille des composantes. Cela permet d'avoir une idée globale de la topologie du réseau. Généralement, on observe deux configurations. Une première configuration dans laquelle la taille des composantes est uniformément distribuée. Une seconde dans laquelle de petites composantes jouxtent un composant de taille beaucoup plus conséquente. Ces deux cas de figure sont illustrés par la figure 16. Dans certains réseaux, la taille de l'élément le plus grand est une quantité importante [78]. Par exemple, dans un réseau de communication comme Internet, la taille de la plus grande composante représente la plus grande fraction du réseau au sein de laquelle la communication est possible. C'est par conséquent une mesure de l'efficacité du réseau dans l'accomplissement de sa tâche [79] [80] [81]. La taille de la composante la plus importante est souvent assimilée à la notion de « composante géante » issue de la théorie des graphes aléatoires.

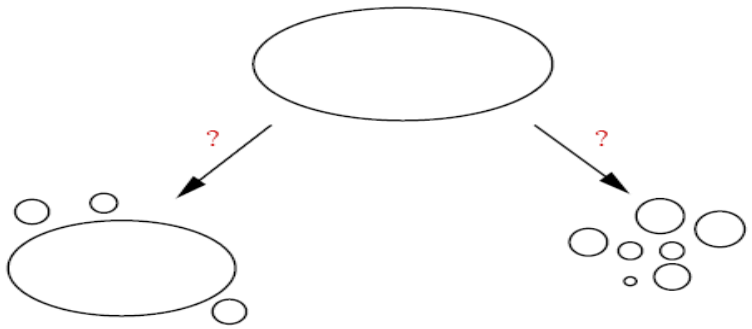


FIG. 16 - Deux types d'organisation en composantes dans les réseaux complexes. Réseau avec composante géante (Gauche) et réseau sans composante géante (Droite).

3.4 Quelques exemples de grands graphes de terrain

Le tableau 8 récapitule les principales caractéristiques topologiques de deux réseaux technologiques (AS2001, Routeurs), de deux réseaux d'informations (Web, Gnutella), de deux réseaux biologiques (Protéines, Métabolique) et de deux réseaux sociaux (Math 99, Acteurs). Nous reportons aussi les valeurs de ces paramètres pour un réseau aléatoire d'Erdös-Rényi (E-R). Ces exemples permettent d'illustrer les grandes similitudes observées à l'échelle macroscopiques de phénomènes de nature très différentes.

TAB. 8 - Valeurs caractéristiques des paramètres pour des grands graphes de terrain dans les domaines technologique, biologique et des réseaux sociaux [63]. Pour les réseaux orientés, l'exposant de la loi de puissance est donné pour les degrés entrants/sortants.

Réseau	Taille n	Degré moyen $<k>$	Distance moyenne L	Transitivité C	Exposant Loi de puissance γ	Corrélation des degrés
E-R	n	np	$\ln(n)/\ln(\langle k\rangle)$	p	Binomiale	0
AS2001	11174	4.19	3.62	0.24	2.38	<0
Routeurs	228263	2.80	9.5	0.03	2.18	>0
Gnutella	709	3.6	4.3	0.014	2.19	<0
Web	$2x10^8$	7.5	16	0.11	2.1/2.7	--
Protéines	2115	6.80	2.12	0 .07	2.4	<0
Métabolique	778	3.2	7.40	0.7	2.2/2.1	<0
Math 99	57516	5.00	8.46	0.15	2.47	>0
Acteurs	225226	61	3.65	0.79	2.3	>0

Le réseau AS2001 cartographie Internet au niveau des systèmes autonomes le 16 avril 2001 [82], tandis que le réseau Routeurs se situe lui, au niveau des routeurs. On remarque que globalement, au facteur d'échelle près, ces réseaux sont similaires hormis pour la corrélation des degrés et la transitivité. Le réseau des Systèmes Autonomes est disassortatif alors que le réseau de routeurs est plutôt assortatif. Il est par ailleurs pratiquement dix fois plus transitif. L'explication tient peut-être à l'organisation plus politique des systèmes autonomes, les routeurs étant déployés par rapport à des considérations plus techniques.

Gnutella est un réseau d'échange pair à pair non orienté qui s'appuie sur l'infrastructure technologique [83]. En ce qui concerne la transitivité, il s'apparente au réseau de routeurs avec une transitivité encore plus faible, par contre il est disassortatif tout comme les réseaux autonomes. Ce comportement est lié à la hiérarchisation introduite par la présence de « super pairs » dans cette architecture. Le réseau Web est un réseau d'information orienté qui représente les liens entre les pages Web. Pour ce réseau de taille impressionnante, la distance reste relativement faible et la valeur de la transitivité est pratiquement la moitié de celle observée pour le réseau des Systèmes autonomes.

Le premier réseau biologique concerne les relations protéiniques dans les levures [84]. Les nœuds représentent les protéines. Un lien est créé si elles interagissent. Le second est un réseau de réactions métaboliques [85]. Ils sont tous deux sont sans échelle et disassortatifs. Ces deux réseaux présentent par ailleurs des caractéristiques assez différentes. Ainsi, contrairement au réseau de protéines, le réseau métabolique est fortement transitif.

Les deux réseaux sociaux considérés concernent un réseau de relation de collaborations scientifiques entre mathématiciens construit à partir des publications communes [69] [86] et un réseau d'acteurs qui ont eu un rôle dans un même film. Ces deux réseaux sont sans échelle et disassortatifs. La taille du réseau d'acteurs est seulement 4 fois supérieure à celle du réseau de mathématiciens et le degré moyen dans le réseau d'acteurs est dix fois supérieur à celui des mathématiciens. La distance moyenne est dans un rapport deux. Le réseau d'acteurs est fortement transitif tandis que cette caractéristique est du même ordre que celle du Web pour

les mathématiciens. Bien que ces deux réseaux soient du même type, ils traduisent néanmoins des schémas relationnels sociaux relativement différents.

3.5 Conclusion

En dépit des origines très diverses des systèmes modélisés, les grands graphes de terrain présentent des caractéristiques topologiques similaires. Contrairement à des systèmes qui peuvent être très « compliqués », mais en suivant un plan prédéfini, ces systèmes du monde réel ne suivent pas une évolution selon un plan d'organisation prédéfini. Au contraire, ils résultent d'une évolution dynamique décentralisée et non planifiée. A partir de mécanismes microscopiques observés au niveau des nœuds et des liens de ces réseaux fortement hétérogènes, des motifs et des régularités statistiques communes apparaissent. L'étude de ces propriétés partagées par la plupart des grands graphes de terrain a permis d'apporter des réponses à des questions aussi diverses que la résistance aux pannes ou à des attaques dans les réseaux technologiques, la propagation d'épidémie et de virus dans les réseaux sociaux et d'informations, et le fonctionnement de la cellule en biologie.
Tout comme le Web, les services Web constituent un « nouveau » réseau d'information dont le développement s'apparente à celui des grands graphes de terrain. A ce titre ils doivent donc être modélisés et étudiés dans le cadre de la théorie des réseaux. C'est le propos du chapitre suivant qui concerne l'étude topologique des réseaux d'interaction de services Web.

4. TOPOLOGIE DES RESEAUX D'INTERACTION

Dans ce chapitre, nous présentons l'analyse topologique des réseaux d'interaction de services Web issus d'une collection de descriptions. A ce niveau nous avons trois objectifs. Tout d'abord nous voulons prolonger les travaux de Oh [25] en ce qui concerne les représentations syntaxiques des services. Nous levons ainsi son approximation qui consiste à considérer des réseaux non orientés. Notre second objectif est de caractériser et de comparer les réseaux sémantiques que nous avons définis au chapitre précédent. Enfin nous tenons à comparer ces deux types de descriptions à partir des propriétés topologiques des réseaux qu'elles permettent d'engendrer.

Dans une première partie, nous exposons la méthodologie adoptée pour conduire cette analyse. Nous présentons les critères qui ont guidé le choix d'une collection ainsi que l'ensemble des réseaux extraits de la collection.
La deuxième partie est consacrée à l'analyse des réseaux de paramètres issus de la collection retenue. Nous nous intéressons aux caractéristiques structurelles de base. Nous étudions la taille et le nombre de liens des réseaux ainsi que leur structure en composantes. Nous examinons par ailleurs la propriété petit monde des réseaux. L'étude de la distribution des degrés nous permet de caractériser les réseaux de paramètres par rapport à la propriété sans échelle. Nous analysons également la transitivité et la corrélation des degrés dans les réseaux.
L'analyse des réseaux d'opérations fait l'objet de la troisième partie. Nous adoptons la même démarche que celle suivie pour l'analyse des réseaux de paramètres.
Enfin, nous fournissons en quatrième partie une comparaison des réseaux de paramètres et d'opérations sur la base de leur topologie.

4.1 Méthodologie

La méthodologie adoptée se subdivise en trois parties. Nous présentons tout d'abord notre démarche pour le choix d'une collection de descriptions de services Web. Ensuite nous introduisons WS-NEXT, un extracteur de réseaux de services Web que nous avons spécialement conçu et développé dans le cadre de ce travail. Nous terminons par la démarche suivie pour l'analyse des propriétés des réseaux d'interaction.

4.1.1 Recherche d'une collection

Critères de comparaison

Afin de pouvoir poursuivre les objectifs fixés, nous devons disposer d'une collection de services Web qui puissent répondre aux critères suivants :

- Les descriptions doivent constituer un échantillon représentatif de services qui permettent de synthétiser des compositions.

- Elles doivent autant que possible être issues du monde réel.

- La collection doit contenir un nombre suffisant de descriptions pour que les propriétés estimées à partir des réseaux la représentant soient statistiquement significatives.

- Les services doivent être décrits syntaxiquement et sémantiquement afin de permettre une étude comparée basée sur les types de description.

Pour trouver une telle collection, nous sommes passés par une phase prospective au cours de laquelle nous avons dû recenser et analyser les collections de descriptions de services Web existantes. Dans ce qui suit nous résumons les éléments qui nous ont permis de choisir une collection.

Collections de descriptions de services Web

Un certain nombre de collections de descriptions de services Web sont à disposition de la communauté scientifique. Elles sont fournies notamment par l'organisation ICEBE [87], le projet ASSAM WSDL Annotator [88], SemWebCentral [39], OPOSSum [89], les auteurs de [90]. Leurs principales caractéristiques sont rassemblées dans le tableau 9.

TAB. 9 - Collections de descriptions de services Web.

Nom	Source	Nature	Langage	Taille	Caractéristiques particulières
ICEBE05 test set	Organisation ICEBE05	Artificielles	WSDL	-	Composition
Public Web Services	Fan et al.	Réelles	WSDL	1544	-
Full Dataset	Projet ASSAM	Réelles	WSDL	800	Annotation
Dataset2	Projet ASSAM	Réelles	OWL-S	164	Services issus de FullDataset Annotée avec Assam annotator
OWLS-TC3	SemWebCentral	Réelles	OWL-S	1007	Interface simple, 1 opération par description, Découverte
SAWSDL-TC1	SemWebCentral	Réelles, Partiellement ré échantillonnées	SAWSDL	894	Interface simple, 1 opération par description, Découverte
SWS-TC	SemWebCentral	N/A	OWL-S	241	N/A

La collection ICEBE05 a été conçue dans le cadre du défi ICEBE'05 (IEEE International Conference on e-Business Engineering). Ce défi se rapporte à la découverte et à la composition de services Web. Les ensembles de tests ICEBE05 sont générés automatiquement. La collection contient 18 ensembles de tests. La taille des ensembles est de

3356, 5356 ou 8356. Les descriptions sont spécifiées en WSDL simplifié et comprennent seulement les sections «message» et «port».

Le projet de l'annotateur WSDL ASSAM (Automated Semantic Service Annotation with Machine learning) porte sur une application conçue pour aider l'utilisateur à annoter sémantiquement des services WSDL existants. Au côté de l'annotateur, deux collections de descriptions de services Web nommées Full Dataset et Dataset2 sont présentes. Full Dataset est une collection de fichiers WSDL organisés dans une structure arborescente correspondant à 26 domaines. Elle contient 816 fichiers WSDL issus du monde réel. Les descriptions ont été recueillies depuis salcentral [91] et XMethods [92]. Dataset2 est une collection de 164 fichiers OWL-S dont les descriptions ont été obtenues en annotant un sous-ensemble des fichiers de la collection Full Dataset.

SemWebCentral est une communauté qui s'est donné pour but de mutualiser les efforts émanant de la communauté du Web sémantique. Des projets de développement d'outils sont trouvés sur le site ainsi que des collections de descriptions de services Web. Trois collections sémantiques sont disponibles: OWLS-TC (OWLS Test Collection), SAWSDL-TC (SAWSDL Test Collection) et SWS-TC (Semantic Web Services Test Collection). Elles sont utilisées pour le concours international annuel sur la Sélection Sémantique de Services (S3).
Les collections OWLS-TCx et SAWSDL-TCx (x désigne la version) sont des collections de descriptions sémantiques. OWLS-TC3 fournit 1007 descriptions sémantiques écrites en OWL-S. Elles sont classées en sept domaines différents: éducation, soins médicaux, nourriture, voyage, communication, économie, armes. Les descriptions des collections OWLS-TCx sont obtenues à partir d'une annotation semi-automatique de descriptions syntaxiques WSDL dont une partie a été récupérée à partir des registres publics UDDI IBM. En ce qui concerne les paramètres, ces descriptions comprennent uniquement l'information sémantique sous la forme de concepts ontologiques. SAWSDL-TC1 est une collection de descriptions au format SAWSDL. Elle offre 894 descriptions de services Web sémantiques organisés dans les mêmes domaines thématiques qu'OWLS-TC3. Une partie des descriptions provient de la collection OWLS-TC2. Elles ont ensuite été ré-échantillonnées pour augmenter la taille de la collection. Les paramètres des opérations sont décrits syntaxiquement par un nom et sémantiquement par un concept ontologique. Actuellement, la version 4 d'OWLS-TC et la version 3 de SAWSDL-TC sont disponibles depuis septembre 2010.

SWS-TC est une collection de 241 descriptions OWL-S. Aucune information n'est disponible sur la nature des descriptions et leurs éventuelles caractéristiques particulières.

OPOSSum (Online Portal for Semantic Services) est une initiative communautaire pour le développement d'une grande collection de services Web réels avec des descriptions sémantiques. Son objectif est de créer un jeu de tests approprié pour éprouver les technologies sémantiques. OPOSSum rassemble les trois collections sémantiques de SemWebCentral. Il fournit également la collection Jena Geography Dataset, dont les descriptions ont été explicitement recueillies dans le cadre d'OPOSSum. La collection contient 201 descriptions issues du monde réel et extraites de sources publiques comme seekda [93], XMethods, webservicelist [94], ProgrammableWeb [95] et GeoNames [96]. Tous les services Web décrits appartiennent au domaine de la géographie et du géocodage.

Public Web Services est une collection qui a été créée dans le but d'étudier l'état des services Web qui étaient publiquement disponibles. Elle contient 1544 descriptions écrites en WSDL collectées depuis les sites XMethods, seekda, webservicelist, WebserviceX.NET [97] et BindingPoint [98].

Bilan comparatif et choix d'une collection

Il apparaît donc que la plupart des collections sont limitées par leur taille, leur degré de réalisme ou ne possèdent tout simplement pas les deux types de description (syntaxique et sémantique). Pour résumer, la collection ICEBE05 est une collection de grande taille, mais les descriptions sont uniquement syntaxiques, elles sont artificielles et générées de façon automatique. Les collections Public Web Services et Full Dataset contiennent un nombre substantiel de descriptions réelles, mais elles sont également uniquement syntaxiques. Dataset2 et SWS-TC sont des collections sémantiques mais de trop petite taille pour être exploitables. OWLS-TC est suffisamment grande mais les descriptions contiennent uniquement l'information sémantique. Notre choix s'est donc porté sur la collection SAWSDL-TC1. SAWSDL-TC1 est la seule à notre connaissance, qui soit à même de répondre à tous nos besoins. Elle est de taille suffisante pour satisfaire nos exigences. Les descriptions contiennent à la fois l'information syntaxique et l'information sémantique. Finalement, bien qu'elle ait été ré-échantillonnée, les descriptions originales sont réelles. SAWSDL-TC1 possède cependant une particularité. Les descriptions présentent une seule interface contenant une unique opération. Ceci implique que les réseaux d'interaction d'opérations et de services sont confondus pour cette collection. Dans ce qui suit, nous parlons donc uniquement de réseaux d'interaction d'opérations et de paramètres.

4.1.2 Extraction de réseaux avec WS-NEXT

Les réseaux sur lesquels porte notre analyse sont extraits à l'aide de l'outil Web Services Network EXTractor (WS-NEXT). WS-NEXT est un extracteur de réseaux qui permet, à partir d'une collection de descriptions de services Web, d'extraire un ensemble de réseaux de similitude et d'interaction conforme aux modèles que nous avons définis préalablement. Cet outil que nous avons développé pour nos besoins est conçu pour être utilisé facilement au travers d'une interface simple et intuitive. Celle-ci permet de spécifier la collection à analyser, le répertoire cible pour accueillir les fichiers de sortie, le nom et le langage de description de la collection à traiter et le profil à utiliser pour ce traitement. Le profil permet de spécifier l'ensemble des caractéristiques définissant un réseau. Dans tous les cas, un profil définit la *granularité (paramètre, opération, service)*, la *description* (*syntaxique*, *sémantique*) et la fonction de *mise en correspondance* utilisée (*égale*, *approximative*, *exact*, *plugin*, *subsume*, *fitin*). De plus, il précise le *mode d'invocation* (*total*, *partiel*) pour les réseaux d'opérations et de services. La figure 17 représente les profils de réseaux d'interaction pouvant être générés par WS-NEXT, sous forme arborescente. Un profil correspond à un chemin partant de la racine de l'arbre jusqu'à une feuille terminale.

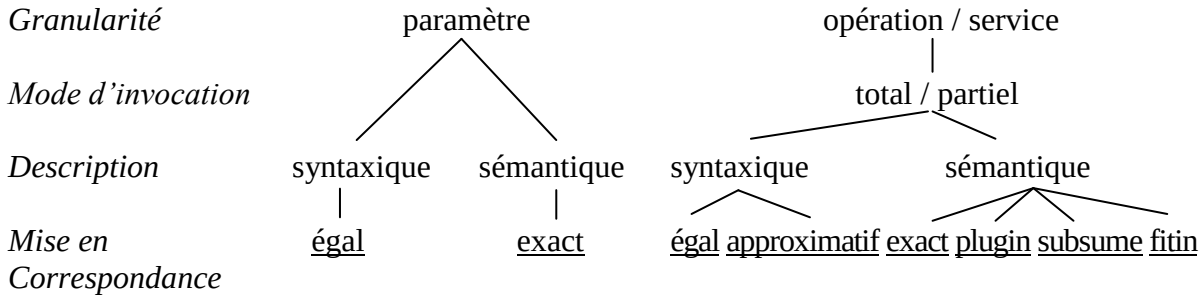


FIG. 17 - Profils de réseaux d'interaction pouvant être générés par WS-NEXT.

L'architecture de WS-NEXT et les étapes d'extraction des réseaux sont illustrées par la figure 18. WS-NEXT est organisé en trois modules qui sont l'analyseur, l'extracteur et le rédacteur.

- L'analyseur traite les fichiers de descriptions un par un. Il détecte les doublons et les élimine. Pour des descriptions sémantiques, les ontologies correspondantes doivent être disponibles lors de l'analyse des fichiers. Les résultats de l'étape d'analyse sont des objets internes qui représentent la collection de descriptions à traiter.

- L'extracteur transforme les descriptions de services en réseaux. Pour ce faire, il prend en entrée les objets « service » ainsi que les profils sélectionnés par l'utilisateur, pour extraire les réseaux appropriés. Les résultats de l'étape d'extraction sont des objets internes qui représentent les réseaux associés à la collection de descriptions.

- Le rédacteur génère plusieurs fichiers localisés dans un répertoire de sortie spécifié :
    - Les fichiers transcrivant les objets réseaux et la représentation interne de la collection ainsi que des métadonnées à propos de la collection source et le profil de l'extraction
    - Des fichiers log contenant les erreurs et les avertissements éventuels correspondant à toutes les étapes du processus
    - Des fichiers statistiques

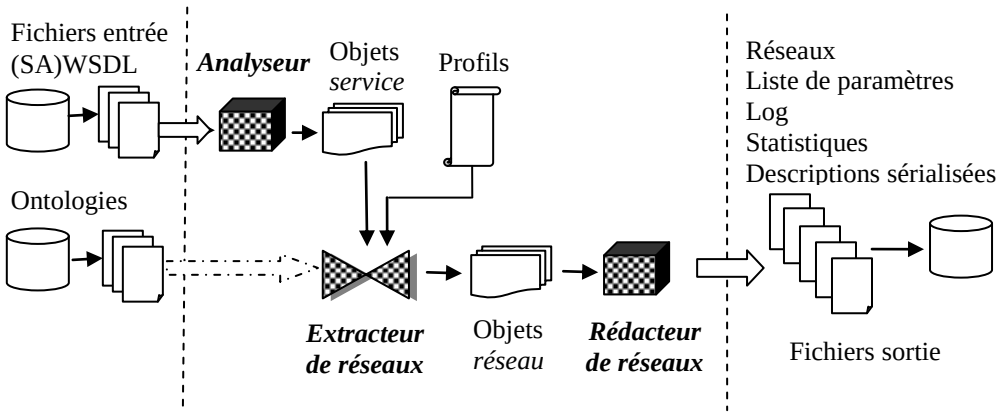


FIG. 18 – Architecture de WS-NEXT. Fichiers d'entrée (Gauche). Architecture du système (Milieu). Fichiers de sortie (Droite).

WS-NEXT est écrit en Java. Il est multi plateforme et repose sur Java Runtime Environment 6. La version actuelle de WS-NEXT supporte les fichiers d'entrée au format WSDL et SAWSDL et produit des fichiers réseau au format Pajek.net [99]. Cependant, l'outil est facilement extensible pour accepter d'autres formats puisque des interfaces Java sont disponibles pour les entrées et les sorties. Ce module de base est couplé à un outil d'analyse basé sur R [100] qui permet d'extraire un ensemble de propriétés topologiques du réseau.

4.1.3 Démarche d'analyse

Nos expérimentations sont tout particulièrement focalisées sur l'étude de la collection SAWSDL-TC1, car c'est la seule qui permette de générer des réseaux syntaxiques et sémantiques pour deux niveaux de granularité. Nous conduisons l'analyse topologique de chacun de ces réseaux. Nous évaluons également l'impact de la sémantique sur leur topologie par des comparaisons inter descriptions comme le montre la figure 19 (Flèches horizontales bleues) et nous interprétons les résultats au regard de différentes activités du cycle de vie des services Web. Nous voulons également savoir si les réseaux d'interaction de paramètres et les réseaux d'interaction d'opérations, qui représentent tous deux la composition de services, véhiculent la même information en termes de propriétés topologiques. En comparant les propriétés des réseaux d'interaction en fonction de la granularité, comme illustré sur la figure 19 (Flèches verticales vertes), nous allons pouvoir donner des éléments de réponse.

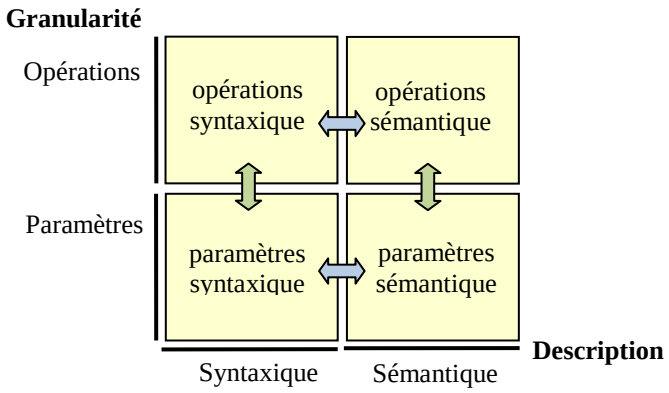


FIG. 19 - Comparaison de la topologie des réseaux d'interaction selon la description (Flèches horizontales bleues) et selon la granularité (Flèches verticales vertes).

La présentation des résultats est organisée autour de la comparaison syntaxique-sémantique. Nous présentons tout d'abord les réseaux de paramètres puis les réseaux d'opérations. En dernier lieu nous établissons une comparaison entre ces deux niveaux de granularité.

Par ailleurs, nous reportons les principaux résultats de l'analyse de la collection Public Web Services afin de pouvoir juger de la consistance des propriétés topologiques des réseaux à description syntaxique.

4.2 Réseaux de paramètres

Dans ce type de réseaux, la fonction de mise en correspondance regroupe les paramètres similaires dans un archétype. C'est la raison pour laquelle, dans le cas du sémantique, nous ne considérons que la mise en correspondance exact. En effet, supposons par exemple que la mise en correspondance subsume soit utilisée, cela revient à représenter par un même nœud du réseau un concept et toutes ses spécialisations. Si tant est que le concept soit à la racine de l'ontologie, celle-ci est toute entièrement projetée dans un nœud du réseau. En ce qui concerne la description syntaxique, seule la mise en correspondance égal est considérée pour des raisons similaires. En effet, plus le seuil de décision est faible, plus des chaînes de caractères différentes sont projetées dans le même nœud. Les deux réseaux étudiés sont donc le réseau syntaxique avec une mise en correspondance égal et le réseau sémantique avec une mise en correspondance exact. L'arborescence des profils utilisés par l'extracteur est représentée par la figure 20.

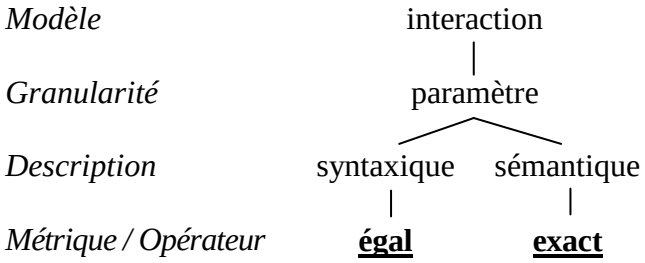


FIG. 20 - Réseaux d'interaction de paramètres.

4.2.1 Caractéristiques de base

Taille du réseau et nombre de liens

Nous observons une différence significative entre le nombre d'instances de paramètres issus des descriptions et le nombre de nœuds dans les réseaux. Les 2136 instances de paramètres de la collection sont représentées par 385 nœuds dans le réseau syntaxique et par 357 nœuds dans le réseau sémantique. L'occurrence moyenne d'un paramètre est ainsi de l'ordre de 5.

La différence de taille observée entre les deux réseaux est de l'ordre de 9%. La taille du réseau sémantique est plus petite que celle du réseau syntaxique. Cela signifie que la mise en correspondance sémantique a contribué à regrouper un plus grand nombre d'instances. Ce résultat met en relief la présence de faux négatifs dans le réseau syntaxique. Les faux négatifs sont les instances généralement différentes d'un point de vue syntaxique parce que les paramètres ont des noms différents. Ils sont cependant identiques d'un point de vue sémantique parce que les paramètres sont associés au même concept. Par exemple, les instances de paramètres nommés `_AUTHOR`, `_AUTHOR1` et `_AUTHOR2` sont représentées par trois nœuds distincts dans le réseau syntaxique. Dans le réseau sémantique, elles sont regroupées en un unique nœud car elles sont associées au même concept `author`.
La mise en correspondance sémantique peut permettre également d'éliminer les faux positifs. Les faux positifs correspondent à des instances représentées par le même nœud alors qu'elles

ne portent pas la même sémantique. La mise en correspondance syntaxique les regroupe de façon inappropriée alors qu'une correspondance sémantique évite cette erreur d'interprétation. Ainsi de nombreux fichiers de descriptions WSDL utilisent le nom générique `parameter` pour nommer des paramètres de natures très différentes. Ce cas n'est pas observé dans SAWSDL-TC1 car la collection à été préalablement « nettoyée ».
La différence entre le nombre de liens est de l'ordre de 5%. Le réseau syntaxique comprend plus de liens que le réseau sémantique parce qu'il y a moins de paramètres regroupés dans les archétypes ce qui augmente le nombre de liens représentant les opérations.

Structuration en composantes

Globalement, les deux réseaux (syntaxique et sémantique) sont très similaires. Dans les deux cas, une composante géante jouxte des composantes plus petites et des nœuds isolés. La figure 21 représente ces réseaux sans les nœuds isolés. La composante géante est juxtaposée à un ensemble des petites composantes.

FIG. 21 – Réseaux d'interaction de paramètres extraits à partir de la collection SAWSDL-TC1. Réseau syntaxique égal (Gauche). Réseau sémantique exact (Droite). Les nœuds isolés ne sont pas représentés.

Le tableau 10 regroupe les principales caractéristiques permettant de résumer la structuration en composantes des réseaux.
La proportion de nœuds isolés est relativement faible (de l'ordre de 5%). Les nœuds isolés correspondent à des paramètres appartenant à des services qui ont seulement des paramètres en entrée ou bien à des services qui ont seulement des paramètres en sortie. De plus ils ne sont pas regroupés dans un archétype. La différence observée entre le nombre de nœuds isolés dans les deux réseaux tient au fait que cette dernière condition n'est pas respectée. Ainsi, le paramètre `Geopolitical-entity1` est un paramètre d'une opération qui n'a pas de paramètre de sortie. Ce paramètre est isolé dans le réseau syntaxique. Dans le réseau sémantique il est représenté par le concept `Geopolitical-entity`. Il existe par ailleurs d'autres opérations qui utilisent ce concept.

TAB. 10 – Caractéristiques de base des réseaux d'interaction de paramètres : réseau global, nœuds isolés, petites composantes, composante géante (pour la composante géante, les proportions sont calculées sur les réseaux sans nœuds isolés).

	Réseau syntaxique	Réseau sémantique
Réseau Global		
Taille *n*	385	357
Nb liens *m*	738	703
Nœuds isolés		
Nombre	18	15
Proportion	4,67%	4,2%
Petites composantes		
Nombre	16	15
Proportion de nœuds	25,8%	20,4%
Etendue de la taille	2-29	2-14
Taille moyenne	6,9	5,2
Ecart type	8,0	5,9
Composante géante		
Taille	269	268
Nb liens	633	621
Proportion de nœuds	73%	78%
Proportion de liens	86%	88%
Densité *d*	0,0087	0,0086

Les petites composantes regroupent plus de 20% des nœuds du réseau. Elles sont plus nombreuses et la proportion est plus élevée pour le réseau syntaxique car les regroupements s'opèrent mieux dans le réseau sémantique. La distribution de la taille des petites composantes mesurée à travers la moyenne semble plutôt mettre en valeur la représentation syntaxique avec une taille moyenne plus élevée. Il faut néanmoins noter que l'écart type est lui aussi sensiblement plus élevé. La distribution étant très fortement asymétrique, il convient plutôt de comparer les histogrammes donnés dans la figure 22. On voit alors que ces distributions sont très similaires.

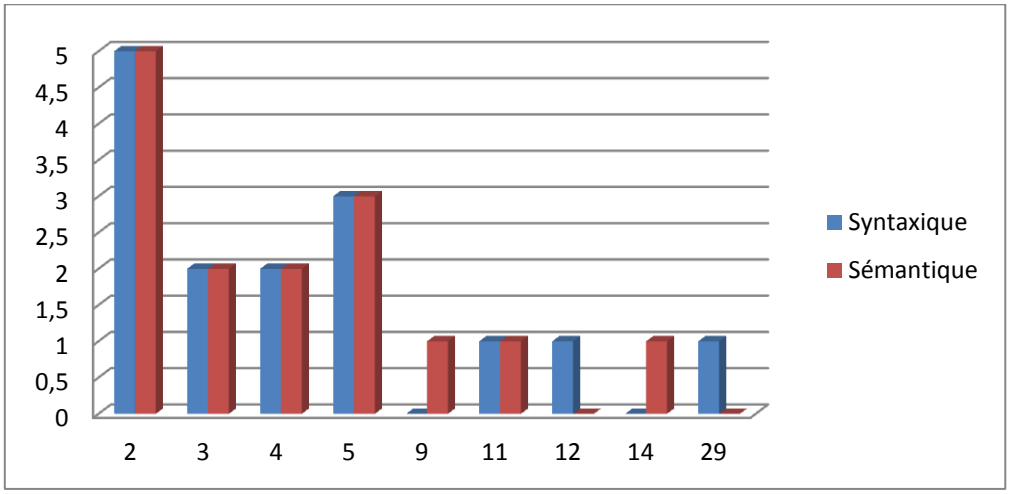


FIG. 22 - Histogramme de la taille des petites composantes.

Globalement, le réseau sémantique contient moins de nœuds isolés et de petites composantes que le réseau syntaxique. Ces propriétés sont plus propices à la composition. En effet, la présence de nombreuses composantes dans un réseau reflète la décomposition de la collection en plusieurs groupes non interactifs de paramètres.

La présence d'une composante géante dans les deux types de réseaux est aussi une propriété intéressante en termes de composition. Cela signifie que le nombre d'interactions dans lesquelles plusieurs opérations sont impliquées est élevé, permettant ainsi à un grand nombre d'opérations d'interagir. La composante géante du réseau sémantique est de plus grande taille. Elle contient également une plus grande proportion de liens. Ceci témoigne de la présence d'un nombre plus élevé d'interactions. Pour conclure, cette analyse de la structure des réseaux d'interaction de paramètres montre qu'un plus grand nombre de services peuvent interagir dans le cas de l'utilisation d'un réseau sémantique.

La taille de la composante géante est une mesure de l'efficacité du réseau dans l'accomplissement de sa tâche. Dans les réseaux de paramètres, la grande majorité des nœuds et des liens se retrouvent dans les composantes géantes. En conséquence, comme il a déjà été pratiqué dans des travaux précédents [25] [69], nous restreignons notre attention aux composantes géantes pour la mesure des distances, des degrés et de la transitivité.

4.2.2 Distances et propriété petit monde

Les valeurs mesurées du diamètre et de la distance moyenne relevées pour les réseaux sont reportées dans le tableau 11. A titre comparatif, nous avons reporté les valeurs de la distance moyenne estimée sur des réseaux aléatoires d'Erdős-Réyni générés avec des paramètres comparables (taille du réseau, nombre de liens). Ces valeurs correspondent à la moyenne obtenue à partir de deux réalisations. L'écart type est reporté.

TAB. 11 – Distance moyenne et diamètre dans les composantes géantes des réseaux d'interaction de paramètres.

	Réseau syntaxique	Réseau sémantique
Distance moyenne *L*	2,75	1,97
Distance moy./écart-type Erdős-Réyni	6,29/0,13	6,24/0,16
Diamètre *D*	7	5

Les réseaux d'interaction de paramètres syntaxique et sémantique possèdent tous deux une petite distance moyenne. Les valeurs correspondantes pour les réseaux aléatoires d'Erdős-Réyni sont trois fois plus élevées. Ces résultats laissent à penser que les réseaux de paramètres possèdent la propriété petit monde. En d'autres termes, de nombreux raccourcis existent dans les réseaux. En termes de composition, cela signifie que l'on peut satisfaire les buts utilisateurs en utilisant très peu d'opérations en moyenne. Si l'on compare les deux réseaux, il s'avère que le réseau sémantique exhibe une distance moyenne plus petite que celle du réseau syntaxique. Ainsi, alors que la taille moyenne d'une composition est de l'ordre de trois

opérations pour le réseau syntaxique, elle se réduit à deux pour le réseau sémantique. Ceci constitue un gain significatif en termes d'efficacité de la composition.

Le diamètre de la composante géante est un indicateur du plus long chemin de dépendance existant dans le réseau considéré. Il est plus grand dans la composante syntaxique que dans la composante sémantique. La différence entre les deux réseaux là encore est significative. Dès lors qu'une solution existe, toute requête peut ainsi être satisfaite en utilisant sept opérations dans le réseau syntaxique contre cinq pour le réseau sémantique. Moins les opérations sont nombreuses pour la production d'un paramètre donné, plus la production est efficace. Ceci impacte toutes les activités du cycle de vie des services. Par exemple, pendant l'exécution d'une composition, cette situation diminue le risque de rencontrer des opérations indisponibles.

En résumé, les réseaux syntaxique et sémantique possèdent tous deux la propriété petit monde. Cette caractéristique plus marquée pour les réseaux sémantiques met en évidence leur plus grande efficacité pour la composition de services.

4.2.3 Distribution des degrés et propriété sans échelle

La visualisation des réseaux permet de constater que quelques paramètres ont un grand nombre de liens alors que la majorité en possède peu. Cette situation est caractéristique d'une distribution des degrés en loi de puissance. Afin de vérifier cette hypothèse, nous avons utilisé la méthodologie décrite dans [101]. Dans un premier temps on estime la valeur de l'exposant de la loi de puissance. Puis nous effectuons un test d'adéquation de loi afin de décider si l'hypothèse de la loi de puissance est plausible. Le test utilisé est le test de Kolmogorov-Smirnov. Il est basé sur la mesure de la distance maximale entre les données empiriques et la distribution estimée. Le test fournit par ailleurs une valeur de probabilité *p* qui quantifie la plausibilité de cette hypothèse. Si la valeur de la probabilité p est grande, alors la différence entre les données empiriques et le modèle peut être attribuée à des fluctuations statistiques seules. Dans le cas contraire, on peut penser que les fluctuations statistiques ne suffisent pas pour expliquer les différences entre les deux distributions et que l'on peut écarter l'hypothèse de loi de puissance.

Le tableau 12 regroupe les valeurs estimées de l'exposant de la distribution des degrés entrants, sortants et globaux. Ce dernier cas correspond à un réseau non orienté. Les valeurs estimées pour la distribution des degrés entrants et globaux sont quasi identiques. Dans les deux cas, la valeur de l'exposant du réseau sémantique est plus faible que celle du réseau syntaxique. Ceci signifie que l'on observe plus de nœuds à fort degré dans le réseau sémantique que dans le réseau syntaxique. En effet, lorsque la valeur de l'exposant diminue, la proportion d'évènements rares augmente. L'explication tient au fait que plus de nœuds rallient les hubs dans la description sémantique.

Les valeurs estimées de l'exposant de la distribution des degrés sortants sont assez singulières. En effet, elles diffèrent notablement de celles observées pour les deux autres

estimées. De plus, la plus faible valeur de l'exposant du réseau syntaxique laisse à penser qu'il contient plus de nœuds à fort degré que son homologue sémantique.

TAB. 12 - Exposant de la loi de puissance estimée dans les réseaux d'interaction de paramètres.

	Réseau syntaxique	Réseau sémantique
Exposant de la loi de puissance γ		
Degrés entrants γ_{in}	3,15	2,99
Degrés sortants γ_{out}	2,01	3,45
Degrés globaux γ_{all}	3,15	3,04

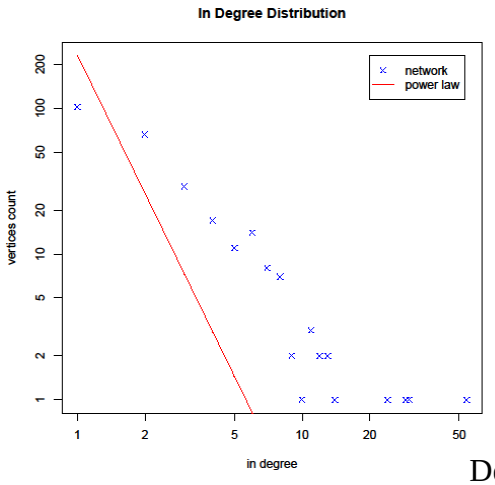
Dans la figure 23 sont représentées, sur un même graphique en échelle log-log, les distributions empiriques observées et les distributions estimées en utilisant les valeurs des exposants reportées dans le tableau précédent pour les réseaux syntaxique et sémantique. Ceci nous permet d'apprécier l'adéquation de la loi de puissance pour ces données. En effet, dans une échelle log-log, toutes les données empiriques devraient se trouver sur une droite. La pente de cette droite permet d'estimer la valeur de l'exposant. On remarque que contrairement aux autres cas, les distributions des degrés sortants des réseaux sont plus dispersées.

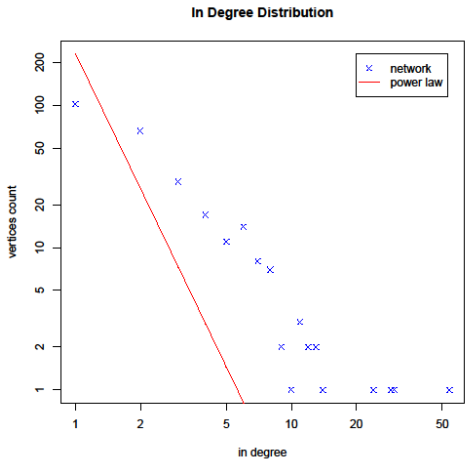
Le tableau 13 reporte les valeurs de la distance du test d'adéquation de loi et les valeurs des probabilités associées. Une faible valeur de la distance traduit une meilleure adéquation, ce qui conduit à une plus grande valeur de la probabilité. Au vu de ces résultats, on peut sans problème considérer que la loi de puissance est une hypothèse plausible pour caractériser la distribution des degrés globaux. On peut en faire de même pour la distribution des degrés entrants. L'hypothèse est plus questionnable pour la distribution des degrés sortants, et ceci tout particulièrement pour le réseau syntaxique. La valeur de *p* nous incline plutôt à rejeter cette hypothèse. Nous pensons néanmoins que cette singularité tient plus aux fluctuations statistiques et ne remet pas en cause le comportement général en ce qui concerne la distribution en loi de puissance. Les propriétés des estimateurs et du test d'adéquation sont en effet fortement influencées par la taille de l'échantillon.

TAB. 13 - Valeurs de la distance du test d'adéquation de loi de puissance et valeurs des probabilités associées pour la distribution des degrés dans les réseaux d'interaction de paramètres.

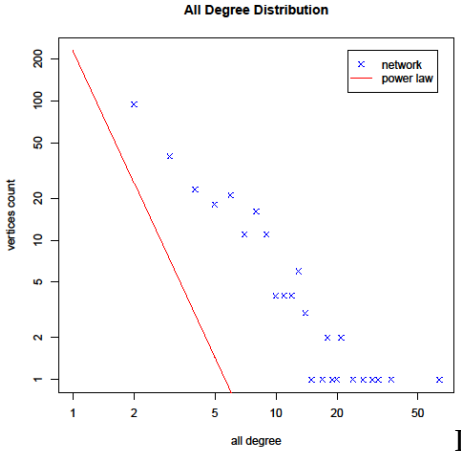
	Réseau syntaxique	Réseau sémantique
Valeur du test d'adéquation		
Degrés entrants D_{in}	0,053	0,047
Degrés sortants D_{out}	0,09	0,063
Degrés globaux $D\gamma_{all}$	0,038	0,037
Probabilité du test d'adéquation		
Degrés entrants p_{in}	0,42	0,57
Degrés sortants p_{out}	0,02	0,21
Degrés globaux p_{all}	0,81	0,84

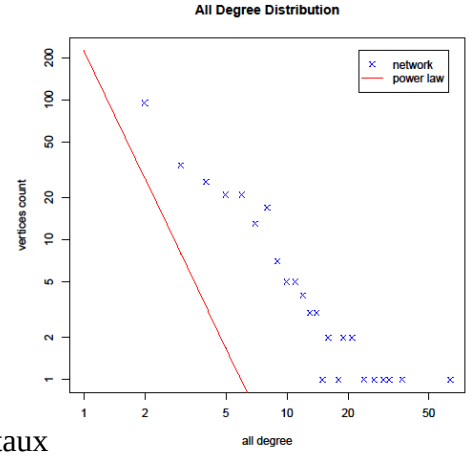
Réseau Syntaxique | Réseau Sémantique



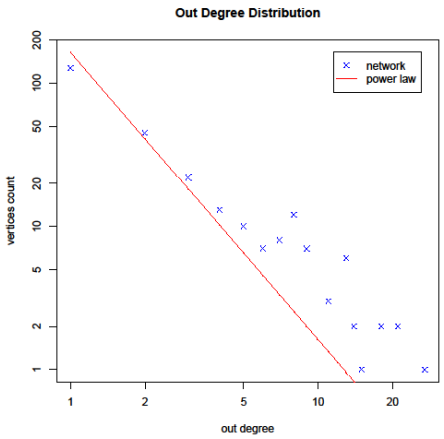


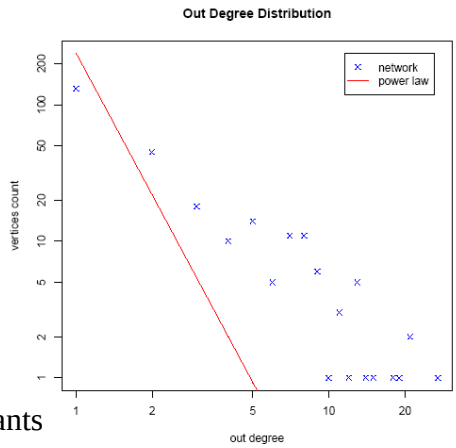
Degrés entrants





Degrés totaux





Degrés sortants

FIG. 23 - Distribution des degrés empirique et estimée de la loi de puissance pour les réseaux d'interaction des paramètres sur une échelle log-log.

4.2.4 Transitivité et corrélation des degrés

Comme nous pouvons le voir dans le tableau 14, la transitivité mesurée dans les réseaux d'interaction de paramètres est relativement faible. Les valeurs du coefficient de transitivité sont d'un ordre de grandeur comparable aux valeurs des coefficients de transitivité obtenus sur des réseaux d'Erdős-Réyni comparables (même nombre de nœuds et de liens).

Sachant que ces réseaux aléatoires sont connus pour avoir un très faible coefficient de transitivité, nous pouvons dire que les réseaux de paramètres ne sont pas transitifs, comme d'ailleurs un certain nombre de grands graphes de terrain. En effet, sur la figure 21 on peut distinguer une organisation hiérarchique des nœuds. Il en résulte une structure de réseau dominée par des arbres plutôt que par des triangles.
La structure en arbre favorise l'apparition de hubs et d'autorités. Les hubs correspondent aux paramètres utilisés comme entrée par de nombreuses opérations tandis que les autorités correspondent aux paramètres en sortie de nombreuses opérations. Hubs et autorités jouent un rôle central dans le processus de composition. Ainsi dans les réseaux de paramètres, `_COUNTRY` et `_PRICE` sont de tels paramètres remarquables.

Si un paramètre est un hub, la production de nombreux autres paramètres dépend de la disponibilité des opérations qui le produisent. Si celles-ci deviennent indisponibles, tous ces paramètres ne peuvent plus être produits. La défaillance des opérations produisant les hubs peut être ainsi très préjudiciable. Si un paramètre est une autorité, il est produit par de nombreuses opérations. Les défaillances des opérations à même de produire les autorités ont donc des conséquences moins graves. Ces résultats sont aussi en adéquation avec la propriété sans échelle de la distribution des degrés qui est caractérisée par un petit nombre de nœuds fortement connectés.

Dans le tableau 14, on remarque que les valeurs de la corrélation des degrés des réseaux syntaxique et sémantique sont sensiblement égales. Ces valeurs négatives indiquent que les nœuds sont significativement organisés de façon disassortative. Autrement dit, les nœuds fortement connectés comme les hubs et les autorités sont préférentiellement liés à des nœuds faiblement connectés.

TAB. 14 - Coefficient de transitivité et corrélation des degrés des composantes géantes des réseaux d'interaction de paramètres.

	Réseau syntaxique	Réseau sémantique
Coefficient de transitivité		
Réseau de paramètres	0,039	0,031
Réseau Erdős-Réyni	0,018	0,020
Corrélation des degrés		
Réseau de paramètres	-0,21	-0,22

4.2.5 Conclusion

Nous avons montré que les réseaux d'information que représentent les réseaux d'interaction de paramètres présentent une structure en composantes avec la présence d'une composante géante. L'analyse des propriétés de la composante géante a montré qu'elle possède les deux propriétés caractéristiques des grands graphes de terrain, à savoir la propriété petit monde et la propriété sans échelle.

D'autre part, ces réseaux exhibent une corrélation des degrés disassortative et une faible transitivité. Les valeurs mesurées de la corrélation des degrés sont comparables à celles des réseaux technologiques (réseau des systèmes autonomes AS2001, réseau Gnutella) et à certains réseaux biologiques (réseau de protéines, réseau métabolique). Leur transitivité est proche de celle des réseaux de routeurs.

Ces réseaux peu denses se présentent sous une forme plutôt arborescente. On note aussi la présence d'un petit nombre de nœuds à forte connectivité. Cette structuration est directement liée à toutes ces propriétés.

La comparaison entre le réseau syntaxique et le réseau sémantique montre qu'une plus grande proportion de nœuds et de liens est intégrée dans la composante géante sémantique. En conséquence, on observe moins de nœuds isolés et de petites composantes dans le réseau sémantique. L'interconnexion du réseau sémantique conduit aussi à une plus petite distance moyenne et à un plus petit diamètre. L'introduction de la sémantique permet ainsi d'accroître l'efficacité des activités du cycle de vie des services web.

4.3 Réseaux d'opérations

L'arborescence des profils utilisés par l'extracteur pour générer les réseaux d'opérations est représentée sur la figure 24. Nous avons ainsi généré un ensemble de douze réseaux. Six réseaux utilisent le mode d'invocation totale, autrement dit l'ensemble des paramètres doit être fourni pour invoquer une opération. Les six autres utilisent le mode d'invocation partielle. Dans ce cas, il suffit qu'un paramètre soit fourni pour pouvoir invoquer une opération. Ces réseaux sont déclinés selon les deux types de descriptions. Pour la description sémantique, nous considérons les mises en correspondance exact, plugin, subsume et fitin. Pour la description syntaxique, nous considérons la mise en correspondance égal et la mise correspondance approximative. Pour la mise en correspondance approximative, nous avons testé trois métriques, Levenshtein, Jaro et Jaro-Winkler. Nous avons mené une étude comparative approfondie sur les propriétés topologiques des réseaux [102] générés à partir de ces métriques. Les résultats de cette étude ont montré que Jaro-Winkler est la meilleure métrique si l'on considère l'introduction de nouvelles similitudes entre paramètres sans l'apparition de faux positifs. Par souci de clarté, nous présentons par la suite seulement les résultats obtenus dans ces conditions optimales pour cette métrique. La mise en correspondance syntaxique approximative est donc représentée uniquement par le meilleur réseau obtenu avec Jaro-Winkler pour notre analyse comparative.

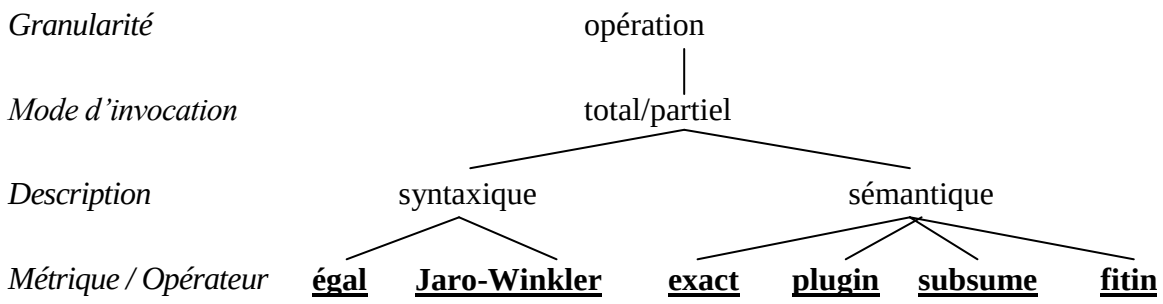


FIG. 24 - Réseaux d'interaction d'opérations.

Nous avons ainsi de nombreux niveaux de comparaison. Le premier d'entre eux est le mode d'invocation. L'invocation partielle offre une solution plus flexible que l'invocation totale. La comparaison inter-mode permet donc d'évaluer l'impact de cette définition plus lâche. Rappelons néanmoins qu'une composition qui trouve sa solution dans un réseau d'interaction à invocation totale offre des solutions plus directes et sera donc plus simple à gérer.

Comme pour les réseaux de paramètres, nous comparons les réseaux d'opérations selon la description. Nous établissons également une comparaison intra description en comparant la topologie des réseaux syntaxiques égal et Jaro-Winkler d'une part et la topologie des réseaux sémantiques exact, plugin, subsume et fitin d'autre part.

4.3.1 Caractéristiques de base

Taille du réseau et nombre de liens

Le tableau 15 répertorie les valeurs relevées en ce qui concerne la taille et le nombre de liens pour l'ensemble des réseaux dans le cas de l'invocation totale et partielle. Nous avons par ailleurs reporté la variation du nombre total de liens lorsque l'on passe d'un réseau en mode d'invocation totale à un réseau en mode d'invocation partielle toutes choses étant égales par ailleurs.

TAB. 15 – Caractéristiques de base des réseaux d'interaction d'opérations. Taille et nombre de liens des réseaux.

	Réseau syntaxique		Réseau sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Taille *n*	785	785	785	785	785	785
Invocation totale Nb liens *mt*	3740	3742	3488	2469	4033	5901
Invocation partielle Nb liens *mp*	4335	4383	3923	3749	5264	7610
Variation du nombre de liens $\frac{\Delta m}{mt}$	16%	17,1%	12,5%	51 %	30,5%	29%

Remarquons tout d'abord que le nombre de nœuds est identique pour tous les réseaux car les nœuds représentent les opérations. Pour comparer la densité des réseaux, il suffit donc d'observer le nombre de liens.

Concentrons-nous tout d'abord sur les réseaux obtenus dans le cadre de l'invocation totale. Comme nous le voyons dans le tableau 15, en ce qui concerne les réseaux sémantiques, le nombre de liens varie dans de grandes proportions. Le réseau qui comprend le plus de liens est le réseau fitin. Le nombre de liens de ce réseau est légèrement supérieur à la somme du nombre de liens des réseaux exact et plugin. Cela tient à la définition de l'opérateur qui englobe les définitions des opérateurs exact et plugin. Le réseau subsume vient en second avec un tiers de liens en moins que le réseau fitin. Dans ce cas, les paramètres de l'opération invoquée sont des spécialisations des paramètres de sortie de l'opération invoquante. La notion de similitude s'étend donc à toute la descendance dans la hiérarchie ontologique des paramètres de sortie de l'opération invoquante. Dans le cas extrême où le concept considéré est au sommet de la hiérarchie ontologique, la similitude concerne alors toute l'arborescence ontologique. Ceci explique le grand nombre de relations observées. Dans le réseau plugin, les paramètres de l'opération invoquée sont des généralisations des paramètres de sortie de l'opération invoquante. La notion de similitude s'étend dans ce cas seulement aux ascendants dans la hiérarchie ontologique des paramètres d'entrée de l'opération invoquée. Dans le cas extrême où le concept est au sommet de la hiérarchie ontologique, il ne peut y avoir de concepts similaires. Ceci explique le fait que ce réseau est celui qui contient le moins de liens. En effet, un certain nombre de paramètres correspondent à cette situation extrême. En ce qui concerne le réseau exact, il occupe la troisième place pour son nombre de liens. Ce nombre est comparable à ceux observés dans les réseaux syntaxiques. Les réseaux syntaxiques contiennent environ 7% de liens en plus. Ceci témoigne de la présence de liens inappropriés dans les réseaux syntaxiques qui n'apparaissent pas lorsque la mise en correspondance est sémantique. Ces faux positifs sont le résultat d'un appariement de paramètres de même nom et de concept différent. Notons que les différences observées entre les réseaux syntaxiques sont non significatives. Ces réseaux sont statistiquement indiscernables.

Nous nous intéressons maintenant aux réseaux en mode d'invocation partielle. Tout d'abord nous remarquons que ces réseaux comprennent toujours plus de liens que les réseaux équivalents en mode d'invocation totale. La variation s'échelonne de 12 à 50%. L'explication tient au fait qu'il suffit dans ces réseaux qu'un seul couple de paramètres soit en correspondance pour créer un lien alors que dans le cas précédent, on ne peut créer un lien que si l'ensemble des paramètres en entrée est en correspondance avec tout ou partie de l'ensemble des paramètres en sortie. En ce qui concerne les réseaux sémantiques, on observe le même comportement que pour leurs homologues en mode d'invocation totale. On peut ainsi les ordonner en fonction de leur nombre de liens de la même façon que précédemment. Ceci est d'ailleurs dû aux mêmes raisons. En termes de liens créés, le réseau plugin est celui qui bénéficie le plus de cette définition moins restrictive. La variation est comparable pour les réseaux subsume et fitin, le gain étant trois fois moins important pour le réseau exact. Les variations observées pour les réseaux syntaxiques sont comparables. Le plus grand pourcentage de nouveaux liens par rapport au réseau sémantique exact tient à la plus grande influence des faux positifs dans cette situation.

Structuration en composantes

Les réseaux d'interaction d'opérations partagent tous la même structure. Dans tous les cas, nous observons la présence de nœuds isolés et d'une composante géante. Hormis le réseau subsume à invocation partielle, tous les réseaux possèdent par ailleurs des petites composantes beaucoup plus petites que la composante géante. Le tableau 16 regroupe les informations concernant les nœuds isolés et les petites composantes pour l'ensemble des réseaux.

TAB. 16 – Structure en composantes des réseaux d'interaction d'opérations. Nœuds isolés et petites composantes.

	Réseau syntaxique		Réseau sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Nœuds isolés Invocation totale						
Nombre	351	349	383	397	361	330
Proportion	44,71%	44,74%	49,00%	50,57%	45,98%	42,30%
Nœuds isolés Invocation partielle						
Nombre	287	278	318	365	261	247
Proportion	36,56%	35,4%	40,51%	46,49%	32,25%	31,46%
Petites composantes Invocation Totale						
Nombre	5	5	7	4	4	2
Etendue de la taille	2-22	3-22	2-28	3-10	2-85	17-34
Petites composantes Invocation Partielle						
Nombre	2	2	4	3	0	1
Etendue de la taille	2-3	2-3	2-6	3-6	0	2

Le nombre de nœuds isolés est particulièrement élevé dans tous les réseaux. Dans un réseau d'interaction d'opérations, les nœuds isolés représentent des services Web qui n'interagissent pas avec d'autres. Aucun de leurs paramètres en sortie n'est utilisé comme entrée et aucun de leurs paramètres en entrée n'est fourni par d'autres services. Ces services peuvent uniquement être invoqués en tant que service atomique. Ils ne peuvent pas être intégrés dans une composition.

Considérons tout d'abord le mode d'invocation totale. Pour tous les réseaux considérés, la proportion de nœuds isolés est à rapprocher du nombre de liens de chacun des réseaux. Plus un réseau comporte de liens, plus sa proportion de nœuds isolés est faible. Ainsi, si l'on ordonne les réseaux selon la proportion de nœuds isolés dans un ordre croissant, cet ordre est conservé si l'on observe le nombre de liens. Notons néanmoins que les différences observées sont bien moins grandes. En effet, la variation globale de la proportion de nœuds isolés est de moins de 10% alors qu'elle est de l'ordre de 2 pour le nombre de liens.
En ce qui concerne le mode d'invocation partielle, globalement, la proportion de nœuds isolés se réduit de l'ordre de 10%. Les remarques précédentes restent valables, à savoir qu'il y a une parfaite corrélation entre le nombre de liens et la proportion de nœuds isolés.

La figure 25 représente les réseaux sans les nœuds isolés en mode d'invocation totale pour illustrer la structuration en composante. Le nombre et la taille des composantes géantes sont reportés dans le tableau 17.

En mode d'invocation totale, les petites composantes sont relativement peu nombreuses. Les réseaux comptent entre deux et sept petites composantes. Le réseau sémantique exact comprend le plus grand nombre de petites composantes. Il est néanmoins de ce point de vue comparable aux réseaux syntaxiques qui sont par ailleurs très similaires. Dans les réseaux plugin et subsume, la mise en correspondance exploitant toute la hiérarchie de l'ontologie, les contraintes sont relaxées par rapport à celle du réseau exact. C'est la raison pour laquelle les nœuds sont groupés au sein d'un nombre plus petit de composantes. Pour la même raison, le réseau fitin comprend seulement deux petites composantes. La taille des petites composantes évolue entre 2 et 34. Notons que dans le réseau subsume apparaît néanmoins une composante de taille non négligeable qui comprend 85 nœuds.
En mode d'invocation partielle, le nombre et la taille des petites composantes diminue fortement. Le nombre de composantes évolue ainsi entre 0 et 4 et leur taille de 2 à 6 nœuds. Dans cette situation, de plus en plus d'opérations sont ainsi rattachées à la composante géante. Ainsi, le réseau subsume ne comprend plus aucune petite composante.

TAB. 17 – Caractéristiques des composantes géantes des réseaux d'interaction d'opérations. Taille, nombre de liens et densité. Proportion de nœuds et de liens des composantes géantes dans les réseaux élagués (sans nœuds isolés).

	Réseau syntaxique		Réseau sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Invocation totale						
Taille	395	397	341	369	329	404
Nb liens	3666	3668	3426	2446	3864	5832
% nœuds	91%	91%	85%	95%	68%	91%
% liens	98%	98%	98%	99%	95%	99%
Densité *d*	0,0235	0,0233	0,0295	0,0180	0,0358	0,0358
Invocation partielle						
Taille	493	497	454	407	538	522
Nb liens	4334	4338	3911	3739	5264	7609
% nœuds	99,99%	99,99%	85%	96.90%	100%	99,61%
% liens	99,93%	99,95%	99.69%	99.73%	100%	99,98%
Densité *d*	0,0178	0,0176	0,019	0,022	0,018	0,028

En ce qui concerne les réseaux à invocation totale, la majorité des nœuds et des liens des réseaux élagués (sans nœuds isolés) se retrouvent dans la composante géante, tableau 17, à l'exception du réseau subsume dans lequel une petite composante de taille significative existe. La grande majorité des opérations qui peuvent entrer dans une composition sont ainsi regroupées. Les composantes fitin, plugin et exact offrent des possibilités pour satisfaire pleinement une requête si tant est qu'une solution existe et que les ontologies sont utilisées efficacement. Autrement dit, l'utilisation d'un concept engage un service à fournir toutes les spécialisations liées à ce concept. A contrario, la composante subsume ne garantit pas que les

requêtes puissent être pleinement satisfaites. La composante fitin comprend à la fois le plus grand nombre d'opérations et de liens. Elle offre ainsi un plus grand nombre de possibilités de compositions. Cette situation est reflétée par sa densité plus élevée. Bien qu'elle possède une taille inférieure à la composante plugin, la composante exact est nettement plus dense. Elle offre ainsi pour les opérations qui la composent de plus grandes possibilités d'interaction. La composante subsume possède une densité comparable à la composante fitin. Quand bien même les compositions possibles n'offrent pas la garantie de pouvoir satisfaire pleinement une requête, cette composante offre des possibilités de composition à ne pas négliger. La taille des composantes géantes des réseaux sémantiques exact, plugin et subsume est plus petite que celle des réseaux syntaxiques. Comme pour les réseaux de paramètres, ces résultats témoignent de la présence de faux positifs. Les faux positifs dans les réseaux d'opérations se traduisent par des liens supplémentaires dans les réseaux syntaxiques.

Si l'on s'intéresse à l'invocation partielle, la taille des composantes géantes est plus importante. Elles représentent quasiment le réseau élagué. La composante fitin conserve ses caractéristiques. Par contre, la composante exact contient plus de nœuds que la composante plugin avec une densité comparable, ce qui la rend d'autant plus efficace. La composante subsume intègre nettement plus de services. Ce gain est obtenu au détriment de la densité.

Pour résumer, nous dirons que le réseau sémantique exact présente plus de nœuds isolés, plus de petites composantes et une composante géante plus petite que les réseaux syntaxiques. Ces propriétés peuvent sembler moins efficaces en termes de composition. Cependant, la structure de connexion est plus précise dans un réseau sémantique. Ceci devrait en conséquence résulter en des compositions plus correctes. Nous pouvons considérer les réseaux plugin et subsume comme des solutions additionnelles pour une tâche de découverte de compositions. Dans ce cas, l'espace de recherche sémantique devient plus grand que l'espace syntaxique. Le réseau fitin constitue également une solution alternative pour la recherche de compositions, avec un espace de recherche qui sera d'emblée plus vaste, grâce à un appariement mixte des paramètres.

De la même façon que pour les réseaux de paramètres, nous nous concentrons sur les composantes géantes des réseaux pour l'étude de la distance moyenne, des propriétés liées au degré et de la transitivité.

Réseau syntaxique égal

Réseau sémantique exact

Réseau sémantique plugin

Réseau sémantique subsume

Réseau sémantique fitin

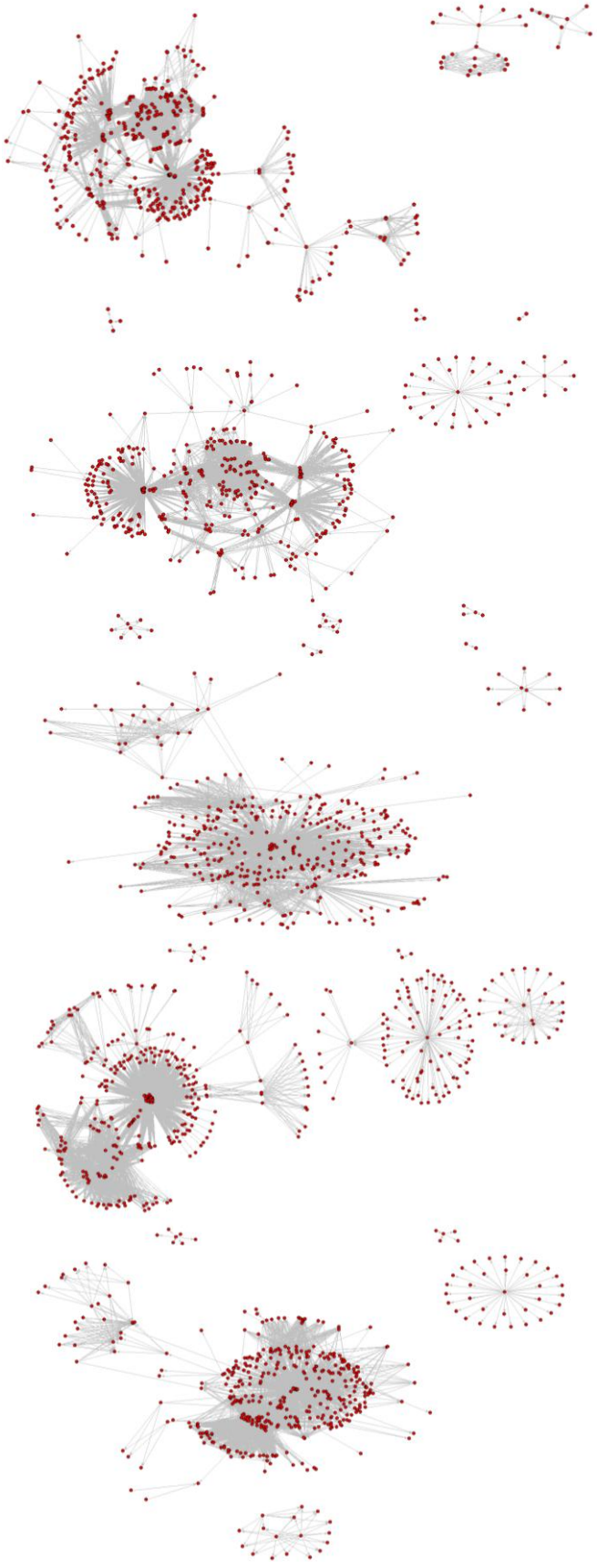


FIG. 25 - Réseaux d'interaction d'opérations à invocation totale élagués.

4.3.2 Distance et propriété petit monde

Comme le montre le tableau 18, tous les réseaux d'interaction d'opérations possèdent une petite distance moyenne. Pour chaque réseau, sa distance moyenne est en effet nettement plus petite que celle du réseau d'Erdős-Réyni correspondant. Ces résultats signifient que les réseaux d'opérations possèdent la propriété petit monde. En d'autres termes, de nombreux raccourcis existent dans les réseaux, indiquant que l'on peut trouver des compositions qui implémentent une fonctionnalité donnée en utilisant un nombre relativement faible d'opérations.

TAB. 18 - Distance moyenne et diamètre dans les composantes géantes des réseaux d'interaction d'opérations.

	Réseau syntaxique		Réseau sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Invocation totale						
Distance moyenne						
Opération *Lt*	2,19	2,18	1,87	1,31	1,38	2,30
Erdős Réyni *Lte*	$2{,}91/10^{-5}$	$2{,}91/10^{-3}$	$2{,}76/10^{-4}$	$2{,}93/10^{-2}$	$2{,}46/10^{-2}$	$2{,}53/10^{-3}$
Diamètre *D*	8	8	4	3	3	6
Invocation partielle						
Distance moyenne						
Opération *Lp*	2,41	2,40	1,90	1,28	1,40	2,26
Erdős Réyni *Lpe*	$3{,}08/10^{-3}$	$3{,}08/10^{-3}$	$3{,}06/10^{-3}$	$3{,}33/10^{-3}$	$2{,}99/10^{-3}$	$2{,}65/10^{-3}$
Diamètre *D*	7	7	4	3	4	6
Variations						
Opération $\Delta L/Lt$	10%	10%	1,6%	2,2%	1,4%	1,8%
Erdős-Réyni $\Delta L/Lte$	6%	6%	10,9%	12%	21,5%	4,8%

Considérons dans un premier temps le cas de l'invocation totale. On remarque que les composantes géantes des réseaux plugin et subsume possèdent les distances moyennes les plus faibles. Cette propriété semble être liée aux nombreux liens créés entre opérations par l'utilisation de la hiérarchie des ontologies. La composante géante du réseau exact se classe en troisième position pour une taille de réseau comparable. Remarquons enfin que le composante géante du réseau fitin possède la valeur la plus élevée pour ces deux propriétés. La combinaison des opérateurs exact et plugin, en offrant davantage de possibilité d'interactions, permet de générer des compositions impliquant un plus grand nombre d'opérations. Si l'on fait abstraction de la composante géante du réseau fitin, les distances moyennes des composantes géantes des réseaux syntaxiques sont plus élevées pour des réseaux de taille comparable. Ceci met en évidence l'apport de la représentation sémantique qui permet une organisation plus efficace de l'espace des services.

Les valeurs observées pour le diamètre sont en harmonie avec les valeurs des distances moyennes. En effet elles évoluent dans le même sens. Le diamètre des réseaux d'interaction d'opérations affiche des valeurs étonnamment faibles, étant donné la taille du réseau. Ces résultats peuvent être dus au fait que la collection considérée a été initialement conçue

essentiellement pour évaluer des fonctions de découverte de services. On peut également observer une différence de diamètre significative entre les réseaux syntaxiques et les réseaux sémantiques. Ceci confirme notre remarque précédente concernant la façon dont leur connectivité diffère. La connectivité est en effet le résultat de la fonction de mise en correspondance. Outre l'opposition syntaxique - sémantique, les résultats semblent aussi être liés à la flexibilité des fonctions de correspondance sémantiques plugin et subsume, puisque les distances moyennes et les diamètres sont plus petits pour les réseaux plugin et subsume que pour le réseau exact.

Dans le cas de l'invocation partielle, on peut constater que la taille des réseaux augmentant, la distance moyenne a tendance à croître aussi. Néanmoins, on peut remarquer que les variations observées pour les composantes géantes des réseaux d'interaction d'opérations sémantiques sont sans commune mesure avec celles observées pour les réseaux d'Erdős-Réyni correspondants. Les évolutions dans le même ordre de grandeur des valeurs observées pour les réseaux syntaxiques laissent penser à une plus grande sensibilité à l'influence du hasard. Remarquons que dans tous les cas, le diamètre du réseau est quasi inchangé.

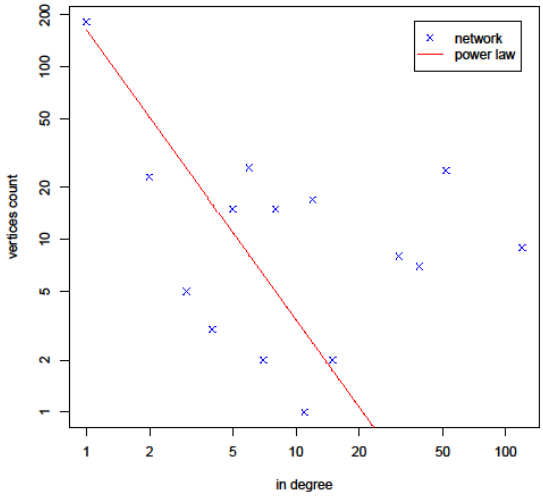
4.3.3 Distribution des degrés et propriété sans échelle

En utilisant la même procédure que pour les réseaux de paramètres, nous avons évalué l'hypothèse que la distribution des degrés des composantes géantes des réseaux d'opérations suive une loi de puissance. Tous les résultats du test de Kolmogorov Smirnov inclinent fortement à rejeter cette hypothèse. Les valeurs de probabilité relevées dans tous les cas sont très proches de zéro. Afin d'illustrer la déviation entre la distribution empirique des données et celle estimée en considérant que les données suivent une loi de puissance, nous avons reporté sur la figure 26 les résultats obtenus pour la composante géante du réseau d'interaction d'opérations sémantique exact. On voit très clairement que trouver une droite qui passe par les données n'est pas chose aisée. Ce qui rend très plausible l'infirmation de l'hypothèse de la loi de puissance. D'ailleurs, ces courbes mettent aussi en évidence le fait que l'effectif des nœuds à fort degré est très élevé en comparaison aux nœuds à faible degré. A titre d'exemple on peut noter que l'effectif des nœuds de degré 2 est de l'ordre de 20 dans la courbe représentant les degrés entrants et du même ordre pour les nœuds de degré 50.
Nous ne pouvons pas exclure que le ré-échantillonnage subi par la collection ait modifié de façon significative les caractéristiques statistiques des données. Cependant, cela reste difficile à évaluer puisque nous n'avons pas accès à cette information. Quoi qu'il en soit, tout laisse à penser que nos réseaux ne possèdent pas la propriété sans échelle.

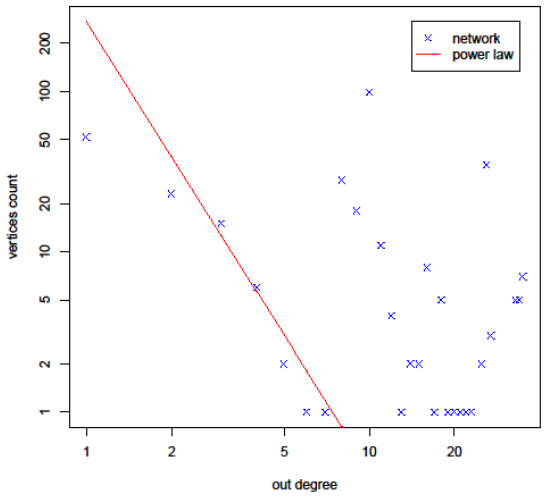
Le tableau 19 reporte les valeurs moyennes et la dispersion du degré global pour les composantes géantes des réseaux d'opérations. On remarque que pour tous les réseaux, le degré moyen est relativement élevé. Il évolue dans un rapport de deux. L'écart type est lui aussi assez conséquent car il est du même ordre de grandeur que le degré moyen. On peut remarquer que le passage de l'invocation totale à l'invocation partielle exerce globalement plus d'influence sur les valeurs de la moyenne que sur celles de l'écart type du degré moyen.

TAB. 19 - Degré moyen et écart type global dans les composantes géantes des réseaux d'interaction d'opérations.

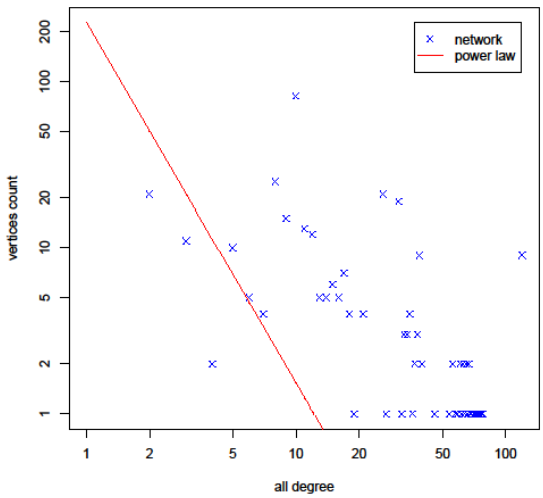
	Réseau syntaxique		Réseau sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Invocation totale Global $<k_{all}>$	18,6/22,7	18,5/22,7	20,1/23,3	13,2/26,4	26,0/30	28,8/42,3
Invocation partielle Global $<k_{all}>$	17,6/21,6	17,5/21,6	17,2/21,9	18,4/28,1	19,6/27,7	29,2/41,3



(a) Degrés entrants



(b) Degrés sortants



(c) Degrés totaux

FIG. 26 - Distribution des degrés et estimée de la loi de puissance pour le réseau d'interaction d'opérations sémantique exact sur une échelle log-log.

4.3.4 Transitivité et corrélation des degrés

Comme en témoignent les résultats présentés dans le tableau 20, la transitivité des composantes géantes des réseaux d'opérations, à l'instar de celle des réseaux de paramètres, est très faible. Elle est dans tous les cas du même ordre et souvent inférieure à celle des réseaux aléatoires comportant le même nombre de nœuds et de liens. On peut donc conclure que ces réseaux sont très peu transitifs. On peut toutefois remarquer que la transitivité a tendance à croître légèrement lorsque l'on passe de l'invocation totale à l'invocation partielle. On observe le phénomène inverse pour les réseaux d'Erdős-Réyni. Ceci laisse supposer une certaine structuration des réseaux indépendante du hasard.

TAB. 20 - Coefficient de transitivité des composantes géantes des réseaux d'interaction d'opérations.

	Syntaxique		Sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Invocation Totale	0,032	0,032	0,022	0,018	0,027	0,056
Réseau Erdős Réyni	0,047	0,048	0,060	0,037	0,092	0,070
Invocation Partielle	0,036	0,036	0,026	0,0355	0,045	0,11
Réseau Erdős Réyni	0,036	0,036	0,037	0,046	0,036	0,056

Les valeurs de la corrélation des degrés relevés sont présentées dans le tableau 21. La valeur négative de la corrélation des degrés observée pour toutes les composantes géantes des réseaux indique que les opérations sont connectées selon un schéma disassortatif. Ceci révèle donc la présence de nœuds importants (de fort degré), les hubs et les autorités connectées à des nœuds de faible degré. Ce phénomène est légèrement moins marqué pour les réseaux à invocation partielle.

TAB. 21 - Corrélation des degrés des composantes géantes des réseaux d'interaction d'opérations.

	Syntaxique		Sémantique			
	égal	Jaro-Winkler	exact	plugin	subsume	fitin
Invocation Totale	-0,45	-0,45	-0,51	-0,50	-0,61	-0,45
Invocation partielle	-0,36	-0,36	-0,43	-0,48	-0,51	-0,39

4.3.5 Conclusion

Les réseaux d'interaction d'opérations exhibent des propriétés observées dans la plupart des réseaux complexes réels, à savoir une structure en composantes avec la présence d'une composante géante et une petite distance moyenne. Cette caractéristique nous permet de conclure que les réseaux possèdent la propriété petit monde. Les réseaux présentent par ailleurs une distribution des degrés qui ne suit pas une loi de puissance. Contrairement aux réseaux sans échelle, ils sont plutôt caractérisés par une présence non négligeable de nœuds de fort degré. Les valeurs importantes de la corrélation des degrés témoignent de la présence de

nœuds importants qui se connectent avec des nœuds à faible degré. Les réseaux ne sont par ailleurs pas transitifs. Le mode d'invocation totale est plus restrictif que le mode d'invocation partielle. Ceci se traduit par des composantes géantes qui comprennent plus de nœuds et de liens pour l'invocation partielle. Ceci permet d'accroître l'efficacité du réseau en termes de composition. Il faut remarquer néanmoins que ce gain est obtenu en relâchant les contraintes sur la qualité des solutions de composition. En effet, rien n'assure qu'un service qui peut fournir un seul des paramètres nécessaire pour invoquer un autre service puisse conduire à une solution satisfaisante.

En ce qui concerne la répercussion des propriétés sur la recherche de compositions, nous pouvons dresser les mêmes conclusions que pour les réseaux d'interaction de paramètres. La composante géante regroupe la majorité des opérations en interaction et étant susceptibles d'entrer dans des compositions. La petite distance moyenne traduit la présence de compositions de relativement petite taille en moyenne, ce qui permet d'atteindre un but à un coût réduit. Enfin, les nœuds importants peuvent servir de guide pour un processus de recherche de compositions.

La comparaison entre les réseaux syntaxiques et les réseaux sémantiques montre qu'une plus grande proportion de nœuds et de liens est intégrée dans les composantes syntaxiques, ce qui traduit la présence d'opérations connectées mal à propos. Bien que les composantes géantes exact et plugin contiennent moins de liens, leur inter connexion est plus efficace ; elle conduit à une plus petite distance moyenne et à un plus petit diamètre, ce dernier correspondant à la taille de la plus grande composition. Comme pour les réseaux de paramètres, nous concluons que l'introduction de la sémantique dans les descriptions permet une représentation plus précise des interactions entre les opérations. Ceci aboutit d'une part à rendre des processus de recherche de compositions plus efficaces et d'autre part à récupérer des compositions sémantiquement fiables.

Une comparaison entre les réseaux sémantiques exact, plugin et subsume fait apparaître une claire distinction entre le réseau à interactions exact et les réseaux à interactions plugin et subsume. Ces derniers conduisent notamment à des valeurs de diamètre et de distance moyenne plus petites. En effet, dans cette situation, un certain nombre de nouvelles interactions sont possibles grâce à la mise en correspondance plus souple qui intègre les relations hiérarchiques entre concepts. Cette caractéristique se traduit par l'existence de compositions impliquant moins de services. Notons néanmoins que le réseau subsume ne permet pas de satisfaire pleinement une requête. En ce sens, il représente des interactions qui restent moins souhaitables que l'interaction exact ou plugin. Le réseau fitin regroupe les services qui entrent en interaction exact et plugin. C'est à ce titre le réseau qui regroupe le plus de nœuds et de liens. Il est donc normal que nous observions des valeurs plus grandes que dans les réseaux exact et plugin pour la taille de la composante géante, pour la distance moyenne et pour le diamètre. Dans un processus de synthèse de compositions, il permet donc d'explorer plus de solutions mais à un coût plus élevé que ses homologues.
Nous n'observons pas de différence notable entre les propriétés des deux réseaux syntaxiques. Ceci est dû au choix de faire ajouter dans le réseau approximatif uniquement des interactions qui restent sémantiquement correctes. Dans ce cas, nous constatons que l'apport n'est pas conséquent.

4.4 Comparaison des réseaux de paramètres et d'opérations

4.4.1 Caractéristiques de base

Taille et nombre de liens

Les deux types de réseaux d'interaction partagent la même structure: un ensemble de nœuds isolés, un ensemble de petites composantes et une composante géante. Dans les réseaux d'interaction de paramètres, les nœuds représentent des archétypes de nom ou de concept de paramètres et les liens représentent des opérations. Dans les réseaux d'opérations, les nœuds représentent les opérations et les liens représentent les possibilités d'invocation entre opérations. Dans le premier cas, la taille du réseau représente la taille du vocabulaire utilisé pour désigner les paramètres alors que dans le second cas, il représente le nombre d'opérations disponibles dans la collection.

Nœuds isolés

La proportion de nœuds isolés est beaucoup plus élevée dans les réseaux d'opérations que dans les réseaux de paramètres. On en dénombre de 4 à 5% dans les réseaux de paramètres contre 44 à 50% dans les réseaux d'opérations, ce qui représente près de douze fois plus. Ils traduisent néanmoins des situations très différentes.
Dans un réseau de paramètres, les nœuds isolés correspondent à des paramètres issus d'opérations ayant uniquement des paramètres en entrée ou en sortie. De plus, le nom ou le concept associé à ces paramètres ne doivent pas être utilisés par d'autres services qui possèdent eux des paramètres en entrée et en sortie. Par exemple, le paramètre nommé `_DUTYTAX` n'apparaît qu'une fois dans la collection comme paramètre en sortie de l'opération `Camerataxedpricedutytax`. Par conséquent, il est représenté comme un nœud isolé. La faible proportion de nœuds isolés dans les réseaux de paramètres indique que peu de paramètres possèdent ces caractéristiques. La grande majorité d'entre eux sont partagés par plusieurs services. Néanmoins, cela ne signifie pas qu'ils peuvent participer à une composition.
Dans un réseau d'opérations, les nœuds isolés représentent des opérations qui ne peuvent pas invoquer ou être invoquées par une autre opération. Par conséquent, elles ne peuvent être invoquées qu'en tant qu'opération atomique et ne représentent aucune valeur ajoutée en termes de composition.

Petites composantes

La structure en composantes des réseaux reflète la décomposition de la collection en plusieurs groupes de paramètres ou d'opérations qui n'ont pas de relation entre eux. Le nombre de petites composantes est globalement plus élevé dans les réseaux de paramètres que dans les réseaux d'opérations. On compte quinze petites composantes dans le réseau de paramètres syntaxique contre cinq dans le réseau d'opérations syntaxique par exemple, ce qui représente trois fois plus. Il y a d'autre part plus de très petites composantes dans les réseaux de paramètres. Par exemple, dans le réseau de paramètres syntaxique, 13 petites composantes sur

16 contiennent seulement de 2 à 6 nœuds. En dehors des différences de taille et d'effectifs, les petites composantes n'ont pas la même signification. Dans un réseau d'opérations, une composante représente nécessairement une composition. La plus petite composante possible, qui contient deux nœuds, contient deux opérations dans une relation d'interaction. Ce n'est pas le cas dans un réseau de paramètres où une composante peut représenter une opération unique. Si dans un réseau de paramètres une composante contient plusieurs opérations, cela signifie qu'elles partagent certains paramètres, mais cela ne signifie pas qu'elles sont liées par une relation de composition. C'est ce qu'illustre l'exemple de la figure 27, avec les opérations `Bookmedicaltransport` (Gauche) et `Providemedicaltransportinformation` (Droite). Ces deux opérations partagent un paramètre en entrée nommé `ProvideMedicalTransportInformation_DesiredTransportVehicle` (Milieu). Par conséquent, aucune relation de composition ne se dégage de cette composante à 11 nœuds.

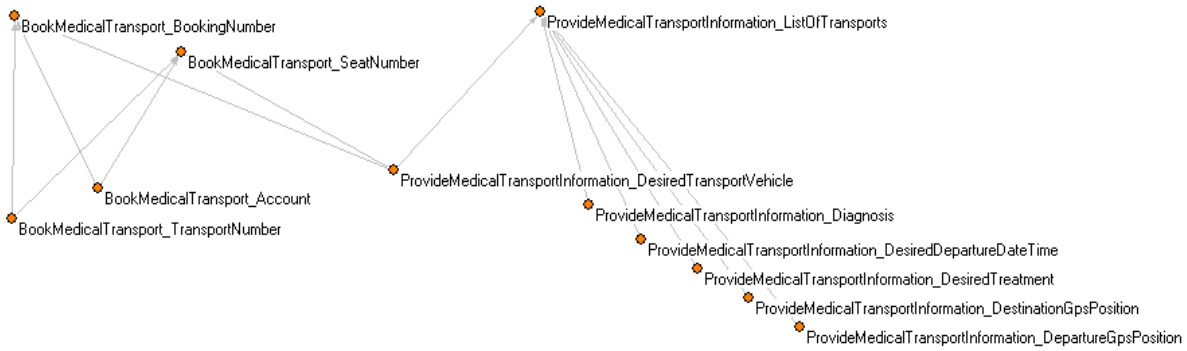


FIG. 27 - Petite composante à 2 opérations dans un réseau d'interaction de paramètres. Opération `getBookMedicalTransport_SeatNumberBookMedicalTransport_BookingNumber` (Gauche) avec 5 paramètres : 3 entrées (Bas), 2 sorties (Haut). Opération `getProvideMedicalTransportInformation_ListOfTransports` (Droite) avec 7 paramètres : 6 entrées (Bas), 1 sortie (Haut). 1 paramètre d'entrée partagé : `ProvideMedicalTransportInformation_DesiredTransportVehicle` (Milieu).

Composante géante

La présence d'une composante géante dans les deux cas est une bonne propriété. C'est le reflet d'un nombre élevé d'interactions entre les services de la collection, permettant à une grande partie d'entre eux de participer à une composition. Nous portons une attention toute particulière aux liens. En effet, les liens sont un élément clé lorsqu'il s'agit de composition. Plus les liens sont nombreux, plus il y a de possibilités de pouvoir composer des opérations. La densité du réseau est un indicateur de la proportion des liens. Comme le montre le tableau 22, les réseaux de paramètres sont moins denses que les réseaux d'opérations. Sachant que les deux types de réseaux représentent les mêmes données, nous pouvons penser que la recherche dans un réseau moins dense est moins coûteuse. Ceci est à pondérer par le fait que les liens n'ont pas la même signification. Dans les réseaux d'opérations en invocation totale, les liens indiquent qu'une opération fournit l'ensemble des paramètres pour en invoquer une autre de façon plus ou moins satisfaisante. Dans les réseaux d'interaction à invocation partielle,

l'opération invoquante ne peut fournir qu'une partie des paramètres en offrant aussi différents niveaux de satisfaction pour ces paramètres. Dans les réseaux de paramètres, un lien indique que deux paramètres sont liés par une relation entrée/sortie. Tous ces réseaux portent donc une information complémentaire qui représente autant de facettes de la composition.

TAB. 22 - Densité des composantes géantes des réseaux d'interaction.

Paramètres		Opérations					
Syntaxique	Sémantique	Syntaxique		Sémantique			
égal	exact	égal	Jaro-Winkler	exact	plugin	subsume	fitin
0.0087	0.0086	0.0235	0,0233	0,0295	0,0180	0,0358	0,0358
Invocation partielle		0,0178	0,0176	0,019	0,022	0,018	0,028

4.4.2 Propriétés

Distance et propriété petit monde

Les deux types de réseaux d'interaction possèdent la propriété petit monde. Ils exhibent en effet une petite distance moyenne. Ils signifient que de nombreux raccourcis existent dans les réseaux. Dans un réseau de paramètres, on peut produire certains paramètres en utilisant un nombre relativement faible d'opérations. Dans un réseau d'opérations, on peut trouver des compositions mettant en œuvre une fonctionnalité demandée en utilisant un nombre relativement faible d'opérations.
Néanmoins, nous pouvons observer que globalement, toutes choses égales par ailleurs, la distance moyenne est plus grande pour les réseaux de paramètres. En effet, nous devons garder à l'esprit que pour représenter une composition minimum, c'est-à-dire impliquant deux opérations, deux liens sont nécessaires dans un réseau de paramètres alors qu'un lien seulement est nécessaire dans un réseau d'opérations. Les valeurs du diamètre sont toutes relativement faibles par rapport à la taille des réseaux. Ainsi, dans tous les cas, la plus grande composition implique peu d'opérations de la collection.

Distribution des degrés et propriété sans échelle

Les réseaux de paramètres possèdent une distribution en loi de puissance. Les valeurs des coefficients observées sont conformes avec ce qui est généralement observé dans des grands graphes de terrain. Les résultats expérimentaux d'adéquation de la loi de puissance pour les réseaux d'opérations inclinent à rejeter cette hypothèse. Nous soupçonnons que cette situation puisse être liée au ré-échantillonnage de la collection. Afin de juger du caractère plus ou moins artefactuel de cette constatation, nous avons analysé la collection purement syntaxique Public Web Services. Les résultats portant sur la propriété sans échelle sont reportés dans le tableau 23. Il apparaît que pour cette collection, la distribution en loi de puissance soit une hypothèse plausible. Notons néanmoins que les valeurs estimées des coefficients de la loi de puissance sont inférieures dans tous les cas à 2. Ceci se traduit par une plus forte proportion de nœuds à fort degré et une plus faible proportion de nœuds à faible degré. Nous pouvons remarquer que l'on observe le même phénomène pour la collection SAWSDL-TC1. Celle-ci est en effet caractérisée par une proportion non négligeable de nœuds de forts degrés. Ces

résultats donnent ainsi plus de poids à l'hypothèse que le ré-échantillonnage ait modifié la distribution des degrés de façon significative. Le phénomène est plus visible sur les réseaux d'opérations car dans ce cas, il agit sur les liens alors que dans les réseaux de paramètres, il agit sur les nœuds du réseau.

TAB. 23 - Exposant de la loi de puissance de la distribution des degrés estimée dans les réseaux syntaxiques de la collection syntaxique Public Web Services et probabilité associée au test d'adéquation de Kolmogorov Smirnov.

	Invocation totale	Invocation partielle
Exposant de la loi de puissance		
Degrés entrants γ_{in}	1,8	1,66
Degrés sortants γ_{out}	1,59	1,11
Degrés globaux γ_{all}	1,66	1,26
Probabilité du test d'adéquation		
Degrés entrants p_{in}	0.47	0.41
Degrés sortants p_{out}	0.26	0.23
Degrés globaux p_{all}	0.79	0.74

Transitivité et corrélation des degrés

Tous les réseaux étudiés présentent des valeurs très faibles de transitivité. Autrement dit, la proportion de cliques d'ordre trois est négligeable. De ce point de vue, ils sont comparables aux réseaux aléatoires. Les réseaux de paramètres sont un peu plus transitifs que les réseaux d'Erdös- Réyni comparables. On observe la tendance inverse pour les réseaux d'opérations.

Comme de nombreux réseaux du monde réel, les réseaux d'interaction de services Web présentent tous une corrélation des degrés disassortative. Dans une telle configuration, les nœuds fortement connectés sont préférentiellement liés à des nœuds faiblement connectés. Ceci traduit la présence de hubs et d'autorités. Ce phénomène est nettement plus marqué dans les réseaux d'opérations. Dans un réseau d'opérations, ces nœuds à forte connectivité représentent des opérations qui peuvent entrer dans de nombreuses compositions. Dans un réseau de paramètres, ils représentent des paramètres partagés par un grand ensemble d'opérations.

4.5 Conclusion

Dans ce chapitre, nous avons étudié les propriétés topologiques des réseaux d'interaction de services Web. Nous avons comparé les réseaux issus des deux types de descriptions (syntaxique et sémantique) et les réseaux au sein d'un même type de description. Nous avons également comparé les réseaux selon la granularité (paramètre, opération). Deux modes d'invocation (totale, partielle) ont été évalués pour les réseaux d'opérations. Quatorze réseaux ont été construits à partir de la seule collection disponible qui se présente sous les formes de descriptions syntaxiques et sémantiques, à savoir la collection SAWSDL-TC1.

L'étude des propriétés topologiques de ces réseaux a montré que les réseaux sémantiques permettent une structuration plus efficace de l'espace des services dans le cadre de la composition.

Bien qu'ils représentent les capacités d'interaction entre services sous des aspects différents, les réseaux d'interaction de paramètres et d'opérations sont caractérisés par des propriétés topologiques similaires :

- La même organisation en composantes avec une composante géante, un ensemble de petites composantes et des nœuds isolés.
- La propriété petit monde qui traduit le fait que chaque paramètre ou chaque opération peut être atteint par un court chemin depuis n'importe quel autre paramètre ou opération.
- Une corrélation des degrés disassortative qui signifie que les nœuds fortement connectés sont liés à des nœuds faiblement connectés.
- Une faible transitivité reflétant une faible proportion de triangles dans les réseaux.
- L'existence de hubs et d'autorités qui sont des paramètres ou des opérations importants dans le réseau, c'est-à-dire partageant des connexions avec de nombreux autres nœuds du réseau.
- La propriété sans échelle est seulement observée dans les réseaux de paramètres. Néanmoins, une analyse plus approfondie révèle certains points qui permettent de nuancer cette conclusion.

Pour terminer cette étude, nous pouvons positionner les réseaux de services Web parmi les autres grands graphes de terrain présentés au chapitre introductif sur les réseaux complexes.

Les réseaux d'interaction de paramètres ont un degré moyen comparable au réseau technologique des systèmes autonomes et au réseau de collaborations scientifiques. Leur distance moyenne est de l'ordre de grandeur de celle des réseaux de protéines et des systèmes autonomes. Leur faible transitivité s'apparente à celle des réseaux de routeurs. La distribution des degrés entrants / sortants du réseau sémantique est proche de celle du réseau représentant le Web.
Les réseaux d'interaction d'opérations ont une taille comparable à celles du réseau d'information Gnutella et du réseau métabolique. La petite distance moyenne des réseaux d'opérations est comparable à celle des réseaux de protéines. Leur très faible transitivité s'apparente à celles du réseau de routeurs et du réseau de protéines

A l'instar des réseaux de systèmes autonomes, Gnutella et des réseaux biologiques, les réseaux d'interaction de paramètres et d'opérations ont une corrélation des degrés disassortative.

5. TOPOLOGIE DES RESEAUX DE SIMILITUDE

Dans ce chapitre, nous présentons l'analyse des réseaux de similitude de services Web. Ce travail vise tout d'abord à comparer les réseaux générés à partir des différentes fonctions de similitude que nous avons définies. De plus, nous voulons aussi pouvoir mener une étude comparative entre les représentations syntaxiques et sémantiques des descriptions de services Web.

Nous donnons en première partie la méthodologie adoptée. Nous justifions le choix de la collection retenue pour cette étude. Nous présentons ensuite l'ensemble des réseaux extraits de cette collection à partir des différentes définitions que nous avons données de la similitude au chapitre 2. Puis nous donnons les éléments d'analyse de la topologie des réseaux.
Dans une deuxième partie, nous nous intéressons aux caractéristiques structurelles de base des réseaux à description syntaxique et sémantique. Nous observons comment les différentes fonctions de similitude influent sur la présence de nœuds isolés et la création de liens dans les réseaux. Nous analysons également la distribution de la taille des composantes. Nous déterminons par ailleurs les relations entre la transitivité des réseaux et les fonctions de similitude.
Enfin, en dernière partie, nous menons une étude détaillée de la structure des composantes afin d'identifier la présence de motifs élémentaires caractéristiques. Nous faisons aussi le lien entre les composantes des réseaux de similitude et la notion de domaine utilisée dans la catégorisation des services.

5.1 Méthodologie

La méthodologie suivie ici est en partie similaire à celle suivie au chapitre 4 pour les réseaux d'interaction. Après avoir choisi une collection, nous donnons les profils des réseaux étudiés et précisons la démarche d'expérimentation adoptée.

5.1.1 Choix d'une collection

La première exigence est que la collection doit être formée d'ensembles conséquents de services aux fonctionnalités similaires. De plus, tout comme pour l'analyse des réseaux d'interaction, nous devons pouvoir disposer d'une collection de services à même de satisfaire les exigences suivantes :

- Les services doivent ensuite être décrits syntaxiquement et sémantiquement, afin de permettre une étude comparée basée sur les types de descriptions.

- Ces services doivent autant que possible être issus du monde réel.

A notre connaissance il n'existe pas de collection dédiée pour tester les modèles et algorithmes de classification. On peut néanmoins penser que des collections organisées en domaines d'intérêt contiennent un ensemble suffisant de services ayant les mêmes fonctionnalités. Si l'on se restreint à l'étude des services à description syntaxique, les

collections Public Web Services et Full Dataset peuvent être considérées. En effet, ces collections sont issues du monde réel. Malheureusement ces deux collections ne satisfont pas la première exigence. La collection Public Web Services a été construite afin d'établir un instantané de l'espace des services Web. Elle contient donc des services de nature très différente. La collection Full Dataset est une collection de services réels qui a été définie afin de répondre à des problématiques d'annotation. Elle regroupe 816 services organisés dans une structure arborescente correspondant à trois niveaux. Le premier niveau contient 26 classes dont une classe qui regroupe 368 descriptions non classifiés. Si la structure de cette collection est très intéressante, la distribution des effectifs dans chacune des classes n'est pas suffisamment équilibrée pour nos besoins. Ainsi, 16 classes contiennent moins de 10 services.

TAB. 24 - Classification des descriptions de la collection SAWSDL-TC1

Domaine	Taille	Sous domaines	Taille
Communication	26	communication-title_comedyfilm_service.	14
		communication-title_videomedia_service	12
Economy	230	economy-book_price_service	37
		economy-bookpersoncreditcardaccount_service	16
		economy-bookpersoncreditcardaccount_price_service	21
		economy-car_price_service	40
		economy-dvdplayermp3player_price_service	14
		economy-maxprice_cola_service	13
		economy-preparedfood_price_service	25
		economy-recommendedprice_coffeewhiskey_service	18
		economy-shoppingmall_cameraprice_service	18
		economy-userscience-fiction-novel_price_service	28
Education	170	education-country_skilledoccupation_service	74
		education-governmentdegree_scholarship_service	40
		education-novel_author_service	22
		education-researcher-in-academia_address_service	16
		education-university_lecturer-in-academia_service	18
Food	27	food-grocerystore_food_service	27
Medical	19	medical-hospital_investigating_service	19
Travel	145	travel-citycountry_hotel_service	23
		travel-geographical-regiongeographical-region_map_service	15
		travel-geopolitical-entity_weatherprocess_service	23
		travel-surfing_destination_service	32
		travel-surfinghiking_destination_service	39
		travel-surfingorganization_destination_service	13
Weapon	37	weapon-governmentmissile_funding_service	37

Tout comme pour les réseaux d'interaction, la collection SAWSDL-TC1 reste, parmi les collections dont nous disposons, la plus adaptée à nos exigences. En effet les services de cette collection possèdent une description syntaxique et sémantique. Elle est partiellement composée de services réels. Elle est par ailleurs organisée en catégories de services de même domaine. Elle contient 894 services mono opération dont 654 sont classifiés en 7 domaines et dont certains sont organisés en sous-domaines. Le tableau 24 représente l'architecture arborescente à deux niveaux de cette collection et fournit les effectifs de chacune des classes.
Au vu de ces données, il apparaît que trois domaines contiennent plus de 80% des effectifs (*economy*, *education*, *travel*). Les effectifs des quatre autres classes (*communication*, *food*,

medical, *weapon*) ne sont pas très élevés et relativement uniformes. Par ailleurs, parmi les domaines à fort effectif, le domaine *economy* semble le plus hétérogène. Ce qui le caractérise c'est que les fonctionnalités des services sont relatives au prix.

5.1.2 Extraction des réseaux de similitude

Les réseaux de similitude sont donc extraits à partir de la collection SAWSDL-TC1 à l'aide de l'outil WS-NEXT que nous avons présenté au chapitre 4. L'arborescence des profils utilisés est présentée sur la figure 28. Chacun des 8 réseaux extraits correspond à un chemin depuis la racine jusqu'à une feuille soulignée. Pour chaque fonction de similitude, nous avons donc une déclinaison syntaxique et une déclinaison sémantique du réseau. Nous avons par ailleurs extrait les réseaux à partir des deux collections à description syntaxique (Full Dataset, Public Web Services). Il s'est avéré que ces deux collections sont inexploitables. En effet, la grande majorité des services se présentent sous la forme de nœuds isolés. Autrement dit, il existe très peu de services qui ont des paramètres de sortie similaires.

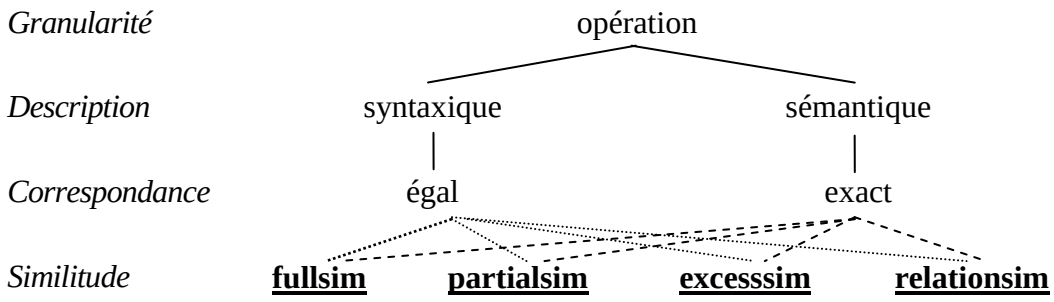


FIG. 28 – Ensemble des réseaux de similitude.

5.1.3 Démarche d'analyse

Les expérimentations sur les descriptions syntaxiques et sémantiques de la collection permettent deux axes d'études comme indiqué sur la figure 29. L'axe vertical permet de mettre en avant les différences topologiques entre les réseaux générés à partir des différentes fonctions de similitude. L'axe horizontal permet quant à lui d'établir des comparaisons liées au type de description de services (syntaxique, sémantique).

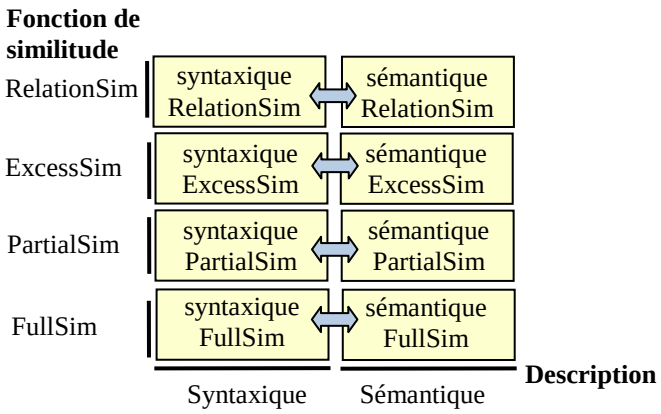


FIG. 29 - Comparaison inter description de la topologie des réseaux de similitude de services Web (Flèches horizontales bleues) et comparaison inter fonction (Axe vertical).

5.2 Caractéristiques de base

Tous les réseaux de similitude présentent la même structure. Un ensemble de petites composantes jouxte des nœuds isolés. Contrairement aux réseaux d'interaction, aucune composante géante n'apparaît. Cette structure reflète la décomposition de la collection en un nombre raisonnable de « communautés ». C'est une propriété intéressante en termes de classification. En effet, des réseaux composés uniquement de nœuds isolés ou des réseaux avec la présence d'une composante géante aurait conduit à une répartition en « communautés » inefficace.

Nœuds isolés

Le tableau 25 résume les informations sur les caractéristiques de base de ces réseaux. Ils contiennent chacun 785 nœuds qui correspondent aux 785 opérations de la collection. Le nombre de nœuds isolés est particulièrement important dans tous les réseaux. La collection ne contient pas un échantillon de services similaires du même domaine assez important pour que les fonctions de similitude incorporent un nombre conséquent d'opérations dans les composantes.

TAB. 25 - Caractéristiques de base des réseaux de similitude.

	Syntaxique				Sémantique			
	Full	Partial	Excess	Relation	Full	Partial	Excess	Relation
Taille	785	785	785	785	785	785	785	785
Nœuds isolés								
Nombre	604	447	486	237	593	449	489	242
Proportion	77%	57%	62%	30%	75%	57%	62%	31%
Nœuds restants								
Nombre	181	338	299	548	192	336	296	543
Nombre de liens	310	412	307	2254	320	399	291	2267
Densité	0,0095	0,0036	0,0034	0,0075	0,0087	0,0035	0,0033	0,0077
Petites Composantes								
Nombre	38	61	67	123	42	59	66	121

Concentrons-nous tout d'abord sur les réseaux issus des descriptions syntaxiques. Le réseau RelationSim se démarque largement des trois autres réseaux. Il est celui qui contient la plus faible proportion de nœuds isolés (31%). On dénombre par ailleurs de 5 à 7 fois plus de liens que dans les trois autres réseaux. Cela signifie que dans la collection, beaucoup d'opérations produisent des sorties identiques avec des entrées complètement différentes. Viennent ensuite les réseaux PartialSim et ExcessSim qui contiennent respectivement 57% et 62% de nœuds isolées. Ces deux types de réseaux orientés ont des contraintes symétriques sur les sorties. Ainsi, les ensembles des paramètres de sortie doivent être « plus ou moins » identiques. On crée un lien d'un service vers un autre dans la situation où le service cible possède plus de paramètres en sortie lorsque la fonction de similitude utilisée est Excesssim. Partialsim décrit quant à lui le cas où le service cible possède moins de paramètres en sortie. En ce qui concerne les contraintes sur les paramètres d'entrée, elles sont moins strictes sur le réseau

PartialSim. En effet, la similitude partielle nécessite qu'un paramètre d'entrée soit partagé alors que dans la similitude avec excès, il faut en plus que le service cible ne dispose pas d'entrées supplémentaires par rapport au service source. Autrement dit, il a au plus le même ensemble d'entrées. Cette asymétrie au niveau des contraintes peut ainsi expliquer la différence observée. Ces deux réseaux ont des caractéristiques très proches à ce niveau. Le réseau FullSim contient quant à lui la plus grande proportion de nœuds isolés. Les opérations produisant les mêmes sorties avec des ensembles de paramètres en entrée se recouvrant sont donc peu nombreuses dans la collection.

En ce qui concerne les réseaux sémantiques, on observe le même type de comportement. Pour des fonctions de similitude identiques, les proportions de nœuds isolés sont très proches. Les différences observées sont liées à des appariements inappropriés de paramètres dans les réseaux syntaxiques. Ceux-ci peuvent se traduire par des fluctuations du nombre de nœuds isolés dans les deux sens. Un appariement inapproprié de paramètres peut permettre de créer un lien dans un réseau syntaxique réduisant ainsi le nombre de nœuds isolés des réseaux syntaxiques par rapport aux réseaux sémantiques équivalents. A contrario, lorsque deux paramètres auraient dû être appariés mais ne le sont pas par la mise en correspondance syntaxique, ceci se traduit par une augmentation du nombre de nœuds isolés dans le réseau syntaxique par rapport au réseau sémantique équivalent. C'est l'effet combiné des faux positifs et des faux négatifs qu'on observe ainsi.

Liens

Observons maintenant le nombre de liens global. On peut remarquer que la comparaison sur l'axe syntaxique-sémantique fait apparaître peu de différences. La variation entre deux réseaux qui utilisent la même fonction de similitude est de l'ordre de la dizaine de liens. Si l'on compare les fonctions de similitude, le réseau RelationSim contient le plus grand nombre de liens. On dénombre de 5 à 7 fois plus de liens que dans les trois autres réseaux. Le réseau FullSim qui contient pourtant la plus grande proportion de nœuds isolés, possède un nombre de liens similaire à celui des réseaux PartialSim et ExcessSim. Grâce à la relaxation des contraintes sur les ensembles de paramètres en sortie des opérations, les réseaux PartialSim et ExcessSim incorporent plus de nœuds dans les composantes que le réseau FullSim. Cependant, ce n'est pas pour autant que le nombre de liens augmente dans les mêmes proportions. En effet, contrairement aux deux autres, une relation FullSim peut être transitive, ce qui offre plus de possibilités de créer des liens.

Si l'on se réfère à la densité globale, là encore les réseaux syntaxique et sémantique ont des comportements très similaires dès lors qu'ils utilisent la même fonction de similitude. Le réseau FullSim présente la plus grande valeur de densité suivi par le réseau RelationSim. Les deux autres réseaux sont environ deux fois moins denses. Ce paramètre est néanmoins à manipuler avec précaution car il ne caractérise pas là une composante, mais des ensembles de composantes assez différents.

Petites composantes

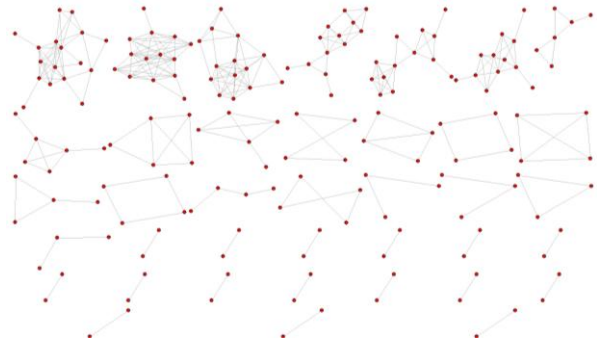
La comparaison du nombre de composantes selon l'axe de la fonction de similitude fait apparaître une grande variabilité. Le réseau RelationSim là encore est celui qui possède le plus grand nombre de composantes. Entre le réseau FullSim et le réseau RelationSim, ce nombre est dans un rapport trois. Les deux autres réseaux toujours très proches, possèdent quant à eux deux fois moins de composantes. Globalement, la comparaison selon l'axe de la description ne fait là encore pas apparaître de différences significatives.

Afin de pouvoir effectuer une comparaison visuelle entre les réseaux de similitude syntaxiques et sémantiques, nous les avons représentés sur la figure 30, sans les nœuds isolés. La comparaison selon l'axe syntaxique-sémantique fait très clairement apparaître la grande similitude des réseaux. Pour mener une analyse plus fine, nous présentons les histogrammes de la distribution de la taille des composantes pour l'ensemble des réseaux sur la figure 31. Tous les réseaux présentent une distribution asymétrique qui s'apparente à une loi de puissance (ou exponentielle). Autrement dit, une très grande proportion de composantes est de petite taille et une faible proportion est de grande taille. Pour pouvoir effectuer aisément la comparaison sur l'axe syntaxique-sémantique nous avons reporté sur un même graphe les histogrammes pour une même fonction de similitude. On voit très clairement que les histogrammes sont très proches ce qui traduit de très faibles fluctuations entre les deux types de description. Afin de quantifier le degré de proximité entre ces distributions empiriques, nous avons calculé le coefficient de corrélation entre les effectifs de chacun des deux histogrammes (syntaxique, sémantique) pour les quatre fonctions de similitude. Les résultats sont reportés dans le tableau 26. Ces résultats mettent en évidence la corrélation quasi parfaite entre les distributions des descriptions syntaxique et sémantique.

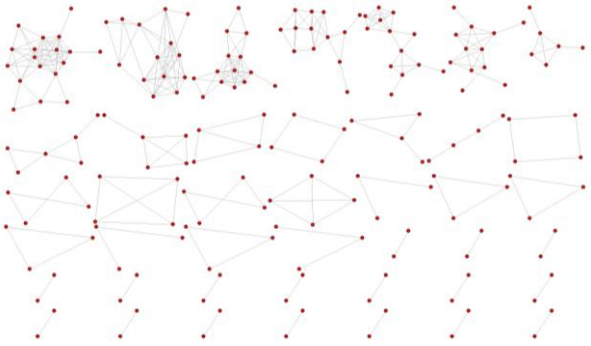
TAB. 26 - Coefficient de corrélation entre les effectifs des histogrammes de la taille des petites composantes syntaxiques et sémantiques.

FullSim	PartialSim	ExcessSim	RelationSim
0,965	0,970	0,978	0,997

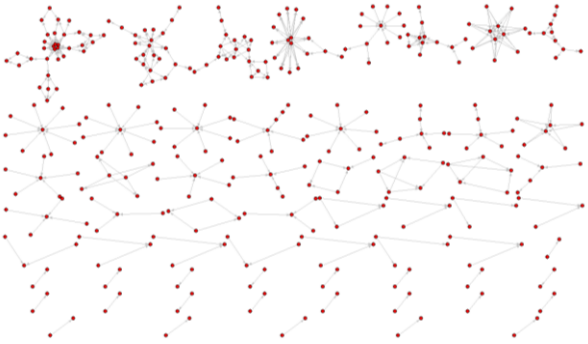
Les caractéristiques statistiques de centralité et de dispersion des distributions de la taille des communautés sont reportées dans le tableau 27. Tout d'abord on peut noter que si l'on effectue la comparaison sur l'axe syntaxique-sémantique, les différences observées sont relativement faibles. Néanmoins, bien que les histogrammes soient très proches, les différences semblent être plus accentuées au niveau de la moyenne et de l'écart type. Cela est dû au fait que ces paramètres sont plus adaptés pour les distributions symétriques. Dans le cas de distributions asymétriques la valeur moyenne et l'écart type sont beaucoup plus sensibles à de faibles variations dans les données.



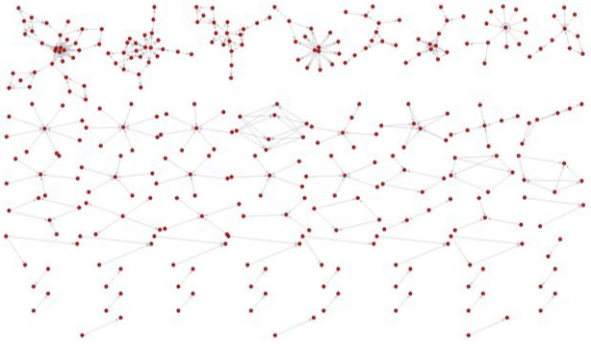
FullSim syntaxique



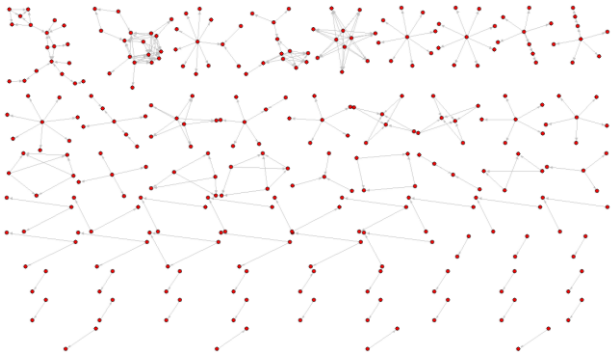
FullSim sémantique



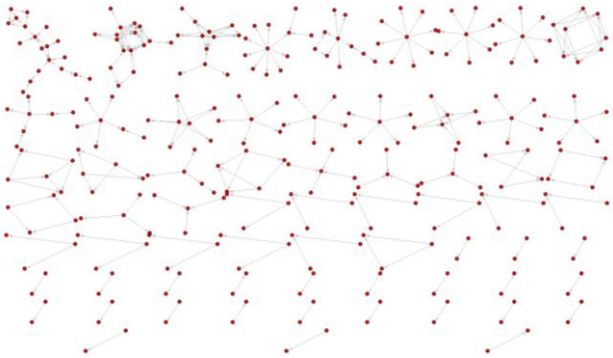
PartialSim syntaxique



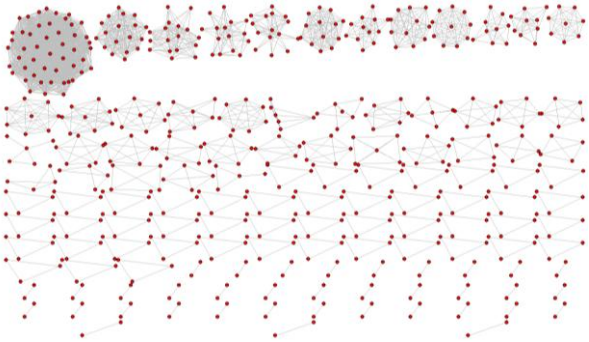
PartialSim sémantique



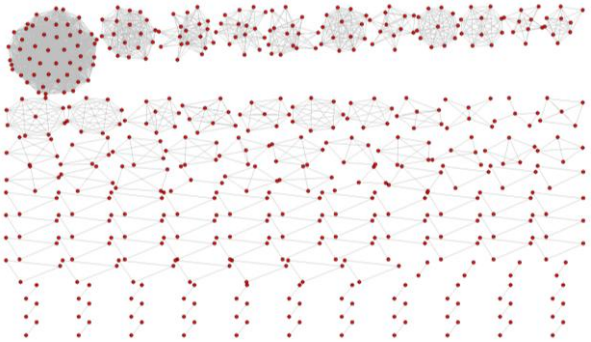
ExcessSim syntaxique



ExcessSim sémantique

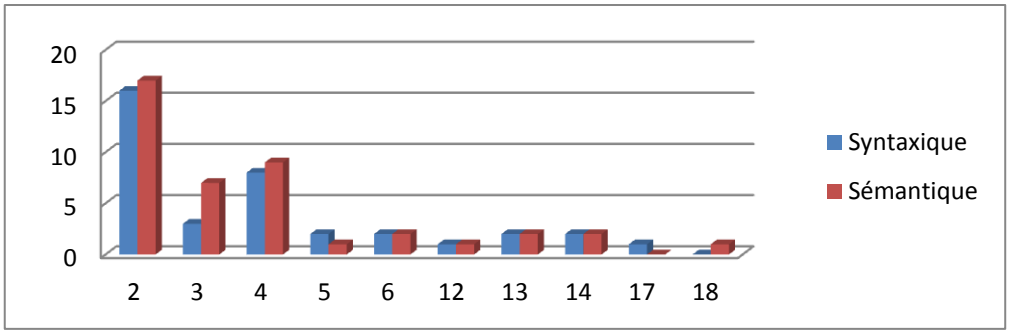


RelationSim syntaxique

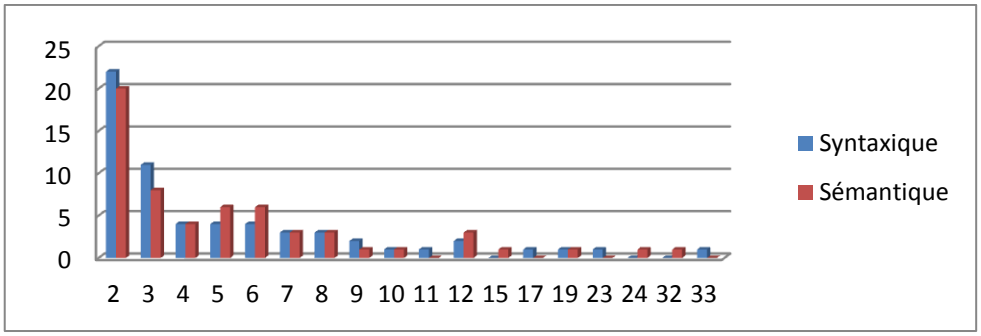


RelationSim sémantique

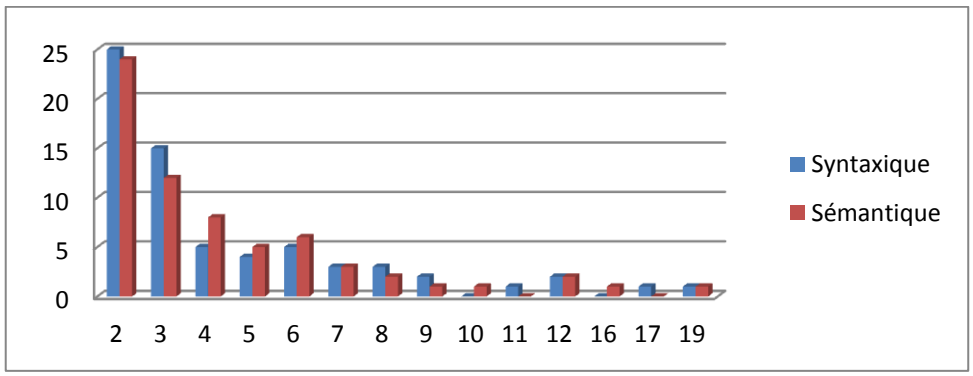
FIG. 30 – Réseaux de similitude.



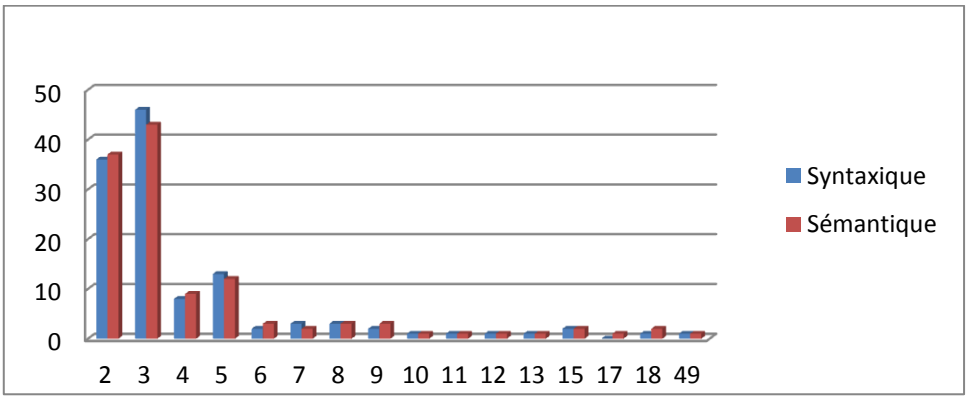
Réseau FullSim



Réseau Partialsim



Réseau ExcessSim



Réseau RelationSim

FIG. 31 - Histogrammes de la taille des petites composantes. L'axe des abscisses représente la taille et l'axe des ordonnées les effectifs.

La comparaison sur l'axe des fonctions de similitude fait clairement apparaître deux types de réseaux. Le premier type correspond au réseau RelationSim caractérisé par une valeur élevée de la taille moyenne et de l'écart type des composantes. Le second type regroupe les trois autres réseaux avec des caractéristiques assez proches. Il faut néanmoins pondérer ce distinguo car lorsque l'on observe les distributions, il apparaît très clairement que l'on peut ordonner les réseaux en fonction de l'effectif et de la taille des grandes composantes. Le réseau RelationSim vient en premier suivi par le réseau PartialSim. Les deux autres réseaux sont très similaires.

TAB. 27 - Valeur moyenne et écart type de la taille des composantes dans les réseaux.

	Syntaxique				Sémantique			
	Full	Partial	Excess	Relation	Full	Partial	Excess	Relation
Moyenne	17,8	18,72	21,35	32,8	19,2	18,66	20,85	35
Ecart type	11,04	11,76	13,94	34,5	12,03	13,36	13,93	31,26

Transtivité

Le coefficient de transitivité des réseaux de similitude est reporté dans le tableau 28. On peut distinguer deux types de comportements à ce niveau :

- Les réseaux FullSim et RelationSim sont fortement transitifs. En effet, les valeurs de la transitivité relevées sont proches de 1 et bien largement supérieures à celles des réseaux d'Erdős-Réyni correspondants (taille et nombre de liens). Ces valeurs suggèrent une organisation sous forme de cliques. La présence de triangles dans les réseaux FullSim traduit le fait qu'au sein d'une composante, de nombreuses opérations qui ont leurs ensembles de paramètres en sortie similaires, ont des ensembles de paramètres en entrée qui se recoupent. Dans le réseau RelationSim, la forte transitivité signifie que de nombreuses opérations, au sein d'une même composante, ont le même ensemble de paramètres en sortie et des ensembles de paramètres en entrée totalement différents.

- Les réseaux PartialSim et ExcessSim ne sont pas transitifs. Les différences observées avec les réseaux d'Erdős-Réyni correspondants ne sont pas très importantes. Elles témoignent tout juste d'une organisation plus structurée que les réseaux aléatoires. L'observation des graphes de ces réseaux fait d'ailleurs apparaître que la structure dominante est plutôt étoilée. Dans le réseau PartialSim on peut distinguer deux types d'étoiles selon l'orientation des liens. Ceux-ci peuvent converger vers le centre de l'étoile ou diverger à partir du centre. Dans le premier cas, la structure étoilée représente la situation où l'opération au centre de l'étoile a moins de paramètres de sortie que les services périphériques. Dans le second cas, ce sont les opérations en périphérie qui ont moins de paramètres en sortie que l'opération située au centre de l'étoile. Les paramètres d'entrée de ces opérations quant à eux se recouvrent. Dans le réseau ExcessSim, on retrouve ces deux situations. Lorsque les liens convergent au centre de l'étoile, l'opération centrale a plus de paramètres en sortie que les opérations périphériques. Lorsque les liens pointent vers l'extérieur, les opérations en périphérie

ont plus de paramètres en sortie que l'opération située au centre. Les paramètres d'entrée de ces opérations sont quant à eux plus contraints.

TAB. 28 - Transitivité dans les réseaux de similitude syntaxiques et sémantiques.

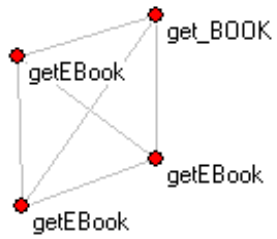
	FullSim	PartialSim	ExcessSim	RelationSim
Syntaxique				
Réseau	0.72	0.02	0.04	0.93
Erdős-Réyni	0,01	0,01	0,01	0,015
Sémantique				
Réseau	0,72	0,025	0,05	0,93
Erdős-Réyni	0,02	0,009	0,005	0,008

Le passage de la description syntaxique à la description sémantique n'exerce aucune influence sur les résultats précédents. Les différences entre les deux types de descriptions étant négligeables, nous ne considérons que les réseaux sémantiques dans ce qui suit.

5.3 Structure des composantes

Réseau FullSim

Les composantes du réseau FullSim sont organisées en cliques. Cette organisation en clique confère au réseau une transitivité élevée. Afin d'illustrer cette caractéristique nous présentons trois composantes extraites de ce réseau.
La première composante représentée sur la figure 32 (Gauche) contient les opérations `get_BOOK` et `getEBook`. Ces quatre opérations forment une clique d'ordre quatre. Ces opérations produisent toutes un seul et même paramètre de concept `Book` et elles partagent au moins un paramètre d'entrée. En fait, les opérations `getEbook` sont identiques et donc parfaitement substituables alors que `getBook` et `getEbook` possédent une seule entrée commune (`Title`). Cette composante regroupe des services qui sont tous similaires au sens de la définition FullSim.



Opération	Entrées		Sorties	
nom	*nom*	*concept*	*nom*	*concept*
`getEBook`	`EBookRequest` `UserAccount`	`Title` `User`	`EBook`	`Book`
`get_BOOK`	`_TITLE`	`Title`	`_BOOK`	`Book`

FIG. 32 - Une composante à une clique d'ordre 4 dans le réseau FullSim (Gauche). Signature des opérations (Droite).

La deuxième composante représentée sur la figure 33 (Gauche) contient six opérations nommées `get_LENDING`. Elles sont organisées en trois cliques dont deux 3-cliques et une 2-clique. Les six opérations ont toutes en sortie un seul et même paramètre nommé `LENDING` de concept `Lending`. Nous reprenons cette clique sur la figure 33 (Droite) en désignant chaque opération par un chiffre et en étiquetant les liens par les paramètres en entrée communs à deux opérations.

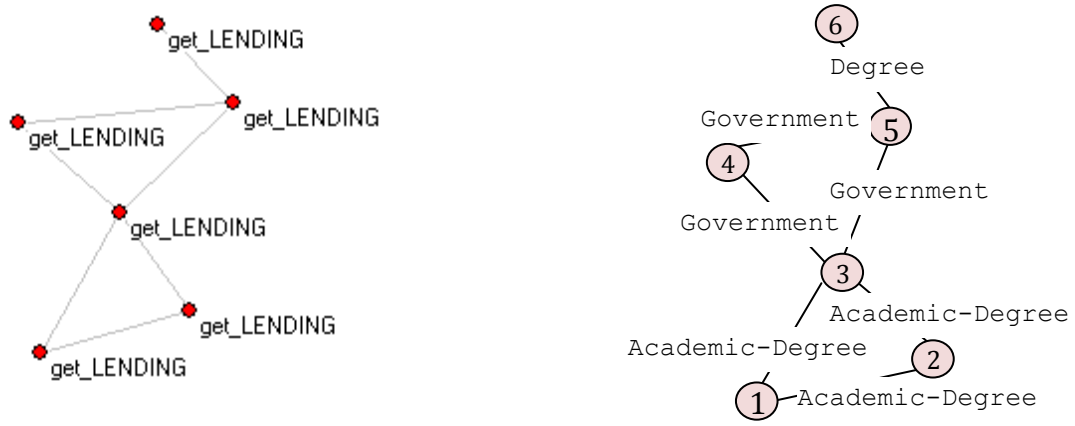


FIG. 33 – Une composante à trois cliques (deux 3-cliques et une 2-clique) du réseau FullSim.

.

Contrairement au cas précédent, dans cette composante, toutes les opérations ne sont pas similaires entres elles au sens de la définition Fullsim. Seules les opérations qui sont au sein d'une même clique respectent cette définition. Cette composante regroupe donc trois ensembles d'opérations similaires qui sont organisées sous forme de cliques. Deux opérations qui ne sont pas dans la même clique sont similaires au sens de RelationSim. Elles ont des sorties communes mais des entrées qui ne se recouvrent pas.

La troisième composante représentée sur la figure 34 contient treize opérations nommées `get_PRICE` et réparties en cinq cliques (une 6-clique, une 4-clique, une 3-clique, deux 2-cliques). Autrement dit, nous avons cinq ensembles d'opérations qui fournissent un prix avec des entrées qui se recoupent dans chacun des ensembles.

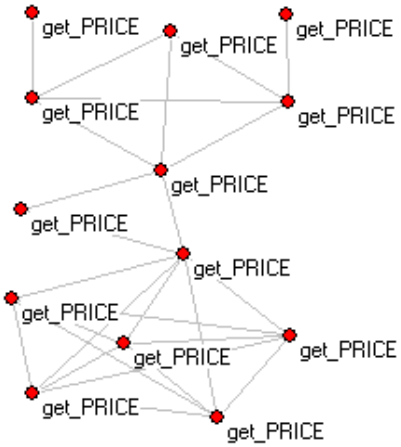


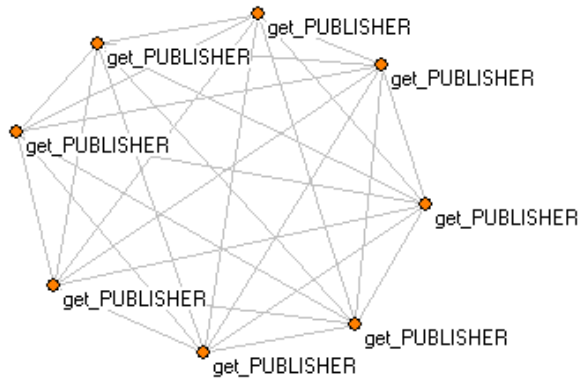
FIG. 34 – Une composante à cinq cliques (1 6-clique, 1 4-clique, 1 3-clique et 2 2-cliques).

Dans le réseau, nous dénombrons deux composantes regroupant des opérations nommées `get_PRICE`. Certaines opérations `get_PRICE` figurent aussi parmi les nœuds isolés. Les opérations `get_PRICE` sont nombreuses dans la collection. Elles appartiennent aux services du domaine *economy*. Dans ce domaine, les services sont répartis en différentes sous catégories (*book*, *food*, *car*, *electronic device*). La répartition en composantes des opérations semblent correspondre à ces catégories.

On voit donc que la répartition en composantes d'opérations similaires permet de classifier les opérations selon leur fonctionnalité. Une composante regroupe un ensemble de services qui ont des sorties identiques. Les services qui possèdent au moins une entrée commune s'organisent sous forme de clique dans cette composante. Une composante n'est donc pas un bloc monolithique de services similaires. Elle peut se décomposer en un ensemble de communautés de services caractérisé par une structure en cliques. Tirer parti de l'existence des cliques dans les composantes permet donc une caractérisation plus fine de la notion de communauté d'opérations.

Réseau RelationSim

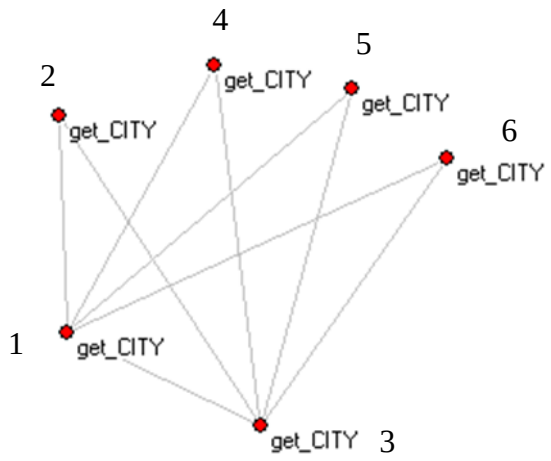
Le réseau RelationSim contient des composantes de grande taille qui sont très densément connectées. L'organisation sous forme de clique est plus marquée. D'ailleurs, certaines composantes de taille conséquente forment un graphe complet. Ceci se traduit par la valeur la plus élevée du coefficient de transitivité observée. La composante de la figure 35 (Gauche) illustre le cas du graphe complet. Les huit opérations `get_PUBLISHER` de la collection ont en sortie le paramètre nommé `_PUBLISHER` de concept `Publisher`. Elles n'ont aucun paramètre d'entrée en commun, figure 35 (Droite). On peut néanmoins remarquer en observant les concepts associés aux paramètres d'entrée, que visiblement les services appartiennent au même domaine.



Nom et concept des paramètres en entrée
{_BOOK, Book}
{_ENCYCLOPEDIA, Encyclopedia}
{_MONOGRAPH, Monograph}
{_NOVEL, Novel}
{_PUBLICATION, Publication}
{_SCIENCE-FICTION-NOVEL, Science-Fiction-Novel}
{_SCIENCEFICTIONBOOK, ScienceFictionBook}
{_SHORT-STORY, Short-Story}

FIG. 35 – Une composante à 8 nœuds du réseau RelationSim sémantique représentée par un graphe complet (Gauche). Nom et concept des paramètres en entrée des opérations `get_PUBLISHER` (Droite).

La figure 36 représente une situation où la composante n'est pas un graphe complet. Les 6 opérations qui forment cette composante ont toutes le même ensemble de paramètres en sortie, constitué dans ce cas d'un seul paramètre de concept `City`.



Opération	Nom et concept des paramètres en entrée
1	{_ACTIVITY, Activity}
2	{_HIKING, Hiking} {_SURFING, Surfing}
3	{_SIGHTSEEING, Sightseeing}
4	{_GENERIC-AGENT, Generic-Agent} {_SURFING, Surfing}
5	{_SURFING, Surfing} {_HIKING, Hiking}
6	{_ORGANIZATION,Organization} {_SURFING, Surfing}

FIG. 36 – Une composante non complète du réseau RelationSim sémantique à 6 nœuds (Gauche). Noms et concepts des paramètres en entrée des opérations (Droite).

L'opération 1 est similaire à toutes les autres opérations car elles ne partagent aucun de leurs paramètres d'entrée. L'opération 3 est également similaire aux cinq autres pour les mêmes raisons. Les opérations 2, 4, 5 et 6 sont quant à elles seulement similaires avec les opérations 1 et 3. Elles n'ont par ailleurs pas de relation de similitude entre elles car elles ont des paramètres d'entrée en commun.

Dans le réseau RelationSim, la plus grande composante contient les 49 opérations `get_PRICE` de la collection avec un degré moyen de 46. Cette composante correspond à un seul domaine si l'on considère le domaine *economy*. Toutefois, on peut distinguer dans ce domaine les sous domaines *car*, *food*, *book* et *electronic device*.

Réseau PartialSim

La structuration des composantes du réseau PartialSim se démarque nettement de celle des deux précédents. Alors que pour les réseaux FullSim et RelationSim on observe une organisation sous forme de cliques, les composantes du réseau FullSim sont plutôt organisées sous forme d'étoiles. Cette organisation confère au réseau une transitivité faible.
La figure 37 présente une composante en étoile du réseau PartialSim. L'opération centrale `get_FILM` produit moins de paramètres en sortie que les opérations périphériques. Elle produit seulement le paramètre de concept `Film` tandis que les cinq autres produisent d'autres paramètres en plus de celui-ci. Elle présente par ailleurs des paramètres d'entrée en commun avec les 5 opérations périphériques. En l'occurrence, ces six opérations ont toutes en commun le même et unique paramètre de concept `Title`. Toutes ces opérations appartiennent au même domaine.

Opération	Nom et concept des paramètres en sortie
1	{_FILM, Film}
2	{_FILM, Film} {_MAXPRICE, MaxPrice} {_QUALITY, Quality}
3	{_FILM, Film} {_TAXEDFREEPRICE, TaxedFreePrice} {_QUALITY, Quality}
4	{_FILM, Film} {_PRICE, Price} {_QUALITY, Quality}
5	{_FILM, Film} {_RECOMMENDEDPRICE, RecommendedPrice} {_QUALITY, Quality}
6	{_FILM, Film} {_TAXEDPRICE, TaxedPrice} {_QUALITY, Quality}

FIG. 37 – Une composante non complète du réseau PartialSim sémantique à 6 nœuds. (Gauche). Noms et concepts des paramètres en sortie des opérations (Droite).

La figure 38 présente une composante à quinze nœuds dans laquelle on retrouve une organisation sous forme d'étoiles imbriquées. L'étoile est ici un motif élémentaire. On distingue quatre étoiles dont les centres sont constitués par les opérations `get_DESTINATION_HOTEL`, `get_ACTIVITY_HOTEL` , `get_SPORTS_HOTEL` , `get_SPORTS_HOTEL`.

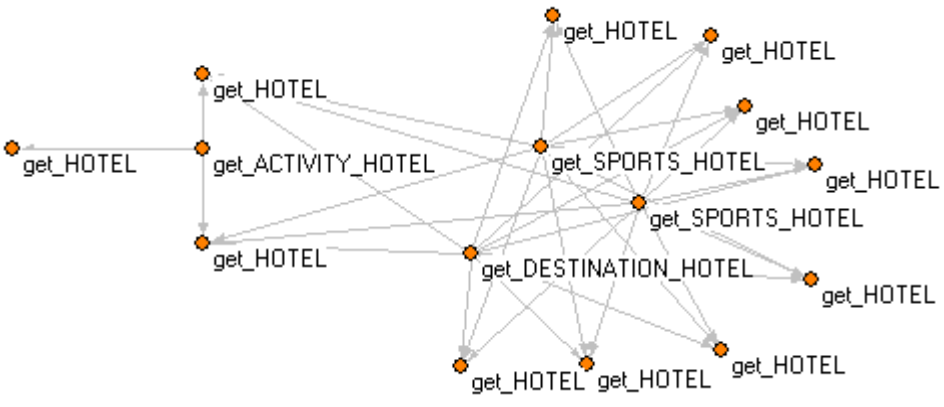


FIG. 38 – Une composante à 15 nœuds du réseau PartialSim sémantique regroupant les opérations `get_HOTEL`.

Dans cet exemple, les opérations appartiennent toutes au domaine *travel*. Ceci n'est pas toujours le cas. En effet elles peuvent aussi appartenir à des domaines différents. La composante qui regroupent les opérations `get_FUNDING`, figure 39, illustre ce cas.

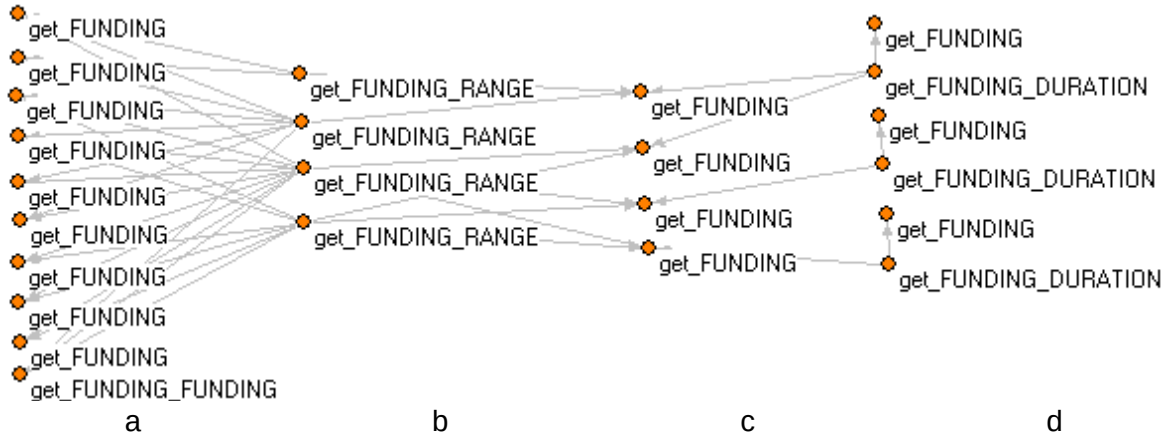


FIG. 39 – Une composante à 24 nœuds du réseau PartialSim sémantique regroupant les opérations `get_FUNDING`.

Dans cette composante à 24 nœuds, les opérations des colonnes étiquetées a et b proviennent du domaine *weapon*. Celles qui sont dans la colonne étiquetée d sont issues du domaine *education*. Enfin, celles qui sont dans la colonne étiquetée c sont issues du domaine *education* et elles ont à la fois un paramètre d'entrée en commun avec deux des opérations `get_FUNDING_RANGE` et une des opérations `get_FUNDING_DURATION`. Nous voyons donc qu'au sein d'une même composante on peut former des sous-ensembles en fonction des domaines. Autrement dit, la notion de similitude peut être affinée au sein d'une même composante. Toutefois, d'autres choix pour l'organisation en domaines des services de la collection peuvent être envisagés. En effet, le domaine *weapon* et le sous domaine *education-governmentdegree_scholarship_service* de *education* d'où proviennent les opérations contenues dans cette composante peuvent très bien être regroupées dans un même domaine. Toutes deux concernent les financements liés à des organisations gouvernementales et c'est ce qui les rassemble dans la même composante.

Réseau ExcessSim

A l'instar des opérations des composantes des réseaux PartialSim, celles des réseaux ExcessSim sont plutôt structurées sous forme hiérarchique avec la présence de hubs et d'autorités.
La figure 40 présente une composante du réseau ExcessSim à six noeuds. Cette étoile est identique à l'étoile formée par ces mêmes opérations dans le réseau PartialSim présentée sur la figure 37. Seule l'orientation diffère. Dans le réseau ExcessSim, les liens sont orientés vers les nœuds périphériques alors que dans le réseau PartialSim, ils convergent vers le centre de l'étoile. Cette situation se produit lorsqu'aucune des opérations périphériques n'a plus de paramètres en entrée que l'opération du centre de l'étoile. Bien entendu, le centre de l'étoile et les opérations périphériques doivent avoir au moins un paramètre d'entrée commun. En l'occurrence, toutes les opérations n'ont qu'un seul et même paramètre d'entrée (`Title`). Dans l'hypothèse où une opération périphérique possède plus de paramètres d'entrée que l'opération du centre de l'étoile, la branche existant entre ces deux opérations dans le réseau PartialSim disparaît dans le réseau ExcessSim.

Une seconde situation permet d'obtenir un cas de figure analogue. Une étoile du réseau PartialSim dont les liens sont orientés vers la périphérie devient une étoile dont les liens sont orientés vers le centre dans un réseau ExcessSim. Ce cas de figure apparaît lorsque l'opération au centre de l'étoile ne possède pas plus de paramètres en entrée que les opérations périphériques. Si l'opération au centre de l'étoile possède plus de paramètres d'entrée qu'une opération périphérique, la branche existant entre ces deux opérations dans le réseau PartialSim disparaît dans le réseau ExcessSim.

Opération	Nom et concept des paramètres en sortie
1	{_FILM, Film}
2	{_FILM, Film} {_MAXPRICE, MaxPrice} {_QUALITY, Quality}
3	{_FILM, Film} {_TAXEDFREEPRICE, TaxedFreePrice} {_QUALITY, Quality}
4	{_FILM, Film} {_PRICE, Price} {_QUALITY, Quality}
5	{_FILM, Film} {_RECOMMENDEDPRICE, RecommendedPrice} {_QUALITY, Quality}
6	{_FILM, Film} {_TAXEDPRICE, TaxedPrice} {_QUALITY, Quality}

FIG. 40 – Une composante du réseau ExcessSim à 6 nœuds (Gauche). Noms et concepts des paramètres en sortie des opérations (Droite).

De la même façon que pour le réseau PartialSim, certaines composantes sont formées d'étoiles « imbriquées ». La composante de la figure 41 illustre la présence des hubs et des autorités.

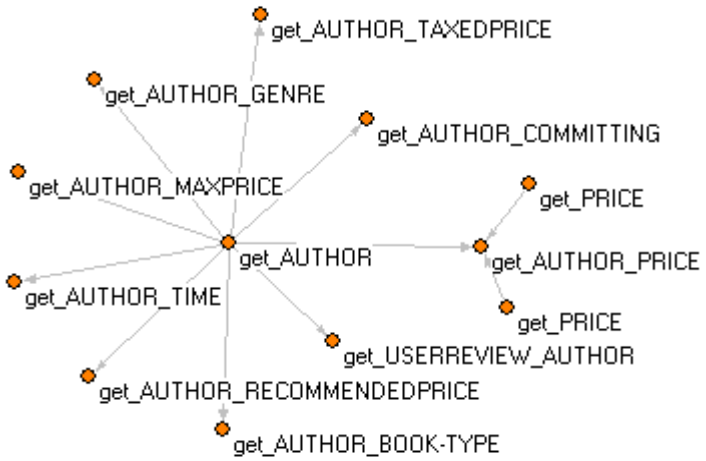


FIG. 41 – Une composante à 12 nœuds du réseau ExcessSim sémantique avec hub et autorité.

L'opération `get_AUTHOR` pointe vers de nombreuses opérations qui lui sont similaires avec excès, tandis que l'opération `get_AUTHOR_PRICE` est pointée par plusieurs opérations parce qu'elle fournit des paramètres supplémentaires. Notons que les opérations sont regroupées dans les composantes par domaine.

Globalement, la contrainte supplémentaire introduite sur les paramètres d'entrée du réseau ExcessSim par rapport au réseau PartialSim se traduit par des composantes de taille plus petite et un plus grand nombre de nœuds isolés. Ainsi, la composante à vingt quatre nœuds présentée pour le réseau PartialSim s'est scindée en un ensemble de petites composantes dans le réseau ExcessSim.

5.4 Conclusion

Dans ce paragraphe, nous avons analysé la topologie des réseaux de similitude extraits de la collection SAWSDL-TC1. Nous avons ainsi comparé la structure des réseaux obtenus avec différentes fonctions de similitude en considérant les deux types de descriptions des opérations de services Web (syntaxique, sémantique).

Notons tout d'abord que tous ces réseaux sont caractérisés par un nombre important de nœuds isolés. Celui-ci évolue de 30% à 75 % en fonction du caractère plus ou moins contraignant de la définition de la similitude adoptée. L'ensemble des nœuds restant s'organise en un grand nombre de petites composantes qui regroupent un ensemble d'opérations similaires. Globalement, la comparaison entre les descriptions syntaxiques et sémantiques des réseaux utilisant des fonctions de similitude analogue ne fait pas apparaître des différences aussi marquées que dans le cas des réseaux d'interaction. Ceci s'explique par une organisation radicalement différente. Alors que les réseaux d'interaction s'organisent autour d'une composante géante dans laquelle les effets cumulés de la correction apportée par la représentation sémantique étaient visibles, l'organisation en de multiples composantes des réseaux de similitude se traduit par un effet de dilution de cette même correction. La distribution de la taille de ces composantes est très asymétrique avec une grande majorité de composantes de petite taille et une petite minorité de composantes de taille plus conséquente. Pour fixer les idées, on peut noter que la plus grande composante observée contient moins de 50 nœuds. La détermination des coefficients de transitivité fait apparaître deux classes de réseaux. Les réseaux RelationSim et FullSim qui sont très transitifs et les réseaux PartialSim et ExcessSim qui ne le sont pas. Dans les deux premiers, le motif élémentaire d'organisation est la clique alors que dans les seconds, c'est l'étoile. L'analyse plus fine de la structure des composantes de chaque réseau confirme cette hypothèse.

Si l'on considère les réseaux organisés sous forme de cliques, on peut établir les constats suivants. Dans le réseau FullSim, une composante regroupe un ensemble de cliques. Toutes les opérations dans une clique partagent des paramètres de sortie identiques et des paramètres d'entrée qui se recouvrent. Par contre, deux services qui n'appartiennent pas à la même clique ont des paramètres d'entrée disjoints. Cette structuration met ainsi en valeur deux niveaux de similitude. Le premier au niveau de la composante et le second au niveau de la clique. Dans

les réseaux de similitude relationnelle, les composantes sont fortement connectées, voire même sont des graphes complets. Ceci traduit le fait que de nombreuses opérations qui possèdent les mêmes paramètres de sortie n'ont pas de paramètres d'entrée commun.

En ce qui concerne les réseaux organisés sous forme d'étoiles, les deux réseaux sont très analogues. En effet, ils possèdent des contraintes symétriques sur les paramètres de sortie. ExcessSim est néanmoins plus contraint sur les paramètres d'entrée. On peut construire ExcessSim à partir du réseau PartialSim en considérant la contrainte supplémentaire introduite sur les paramètres d'entrée. Celle-ci conduit à briser des composantes du réseau PartialSim et à accroître le nombre de nœuds isolés.

L'analyse des réseaux issus de la collection SAWSDL-TC1 nous a permis de mieux appréhender les relations de similitude entre services Web. Néanmoins, pour aller plus loin il est nécessaire de disposer d'une collection de plus grande taille mieux adaptée à nos besoins.

6. STRUCTURE COMMUNAUTAIRE DANS LES RESEAUX COMPLEXES

La détection de communautés est une branche de recherche des plus actives dans le domaine des réseaux complexes. C'est en effet un problème qui a des répercussions considérables sur la compréhension de la structure et du fonctionnement macroscopique des grands graphes de terrain [103], [104]. C'est un problème qui s'apparente au clustering de données et à la partition de graphes. Il en diffère néanmoins par la taille et la structure particulière des grands graphes de terrain. Ces particularités se doivent d'être prises en compte pour développer et évaluer des techniques de détection de communautés. Outre le fait qu'ils permettent une analyse plus fine des réseaux, les algorithmes de découverte de communautés peuvent être utilisés comme éléments de base pour des tâches complémentaires. Ainsi, afin de répondre à la problématique liée à la taille des grands graphes de terrain, on peut envisager d'utiliser les communautés pour diviser le graphe afin de paralléliser les traitements. Ils peuvent aussi être utilisés dans le cadre de la visualisation pour mieux appréhender les structures du graphe à l'échelle macroscopique. Ce large éventail d'applications a conduit à l'élaboration de nombreux algorithmes pour la détection des communautés.

Dans ce chapitre, nous précisons la notion de communauté dans le cadre des réseaux complexes, puis nous définissons les mesures spécifiques pour analyser la structure communautaire des grands graphes de terrain. Nous nous attachons ensuite à présenter un échantillon d'algorithmes de détection de communautés issus des approches les plus fécondes de cette problématique. Le foisonnement des méthodes proposées dans la littérature fait que nécessairement, il est très difficile de couvrir le domaine de façon exhaustive. Néanmoins, afin d'illustrer l'éventail des solutions proposées, nous nous sommes attachés à sélectionner des méthodes pour leur aspect fondamental ou pour leur efficacité reconnue. Nous présentons ensuite la mesure de qualité de la structure communautaire qui est utilisée de façon prépondérante dans ces algorithmes. Pour plus d'information à ce sujet on pourra se reporter à [59] qui dresse un panorama assez complet des mesures alternatives. Enfin, nous présentons les méthodes d'évaluation des performances pour comparer les algorithmes de détection de communautés. En effet, suivant la nature des données, les performances des algorithmes peuvent être très différentes, ce qui nécessite de pouvoir les comparer dans le cadre spécifique des réseaux d'interaction de services Web.

6.1 Définition et propriétés topologiques des communautés

6.1.1 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 [105]. 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 [106]. 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 42. Le réseau de gauche possède une structure communautaire. Le réseau de droite est dépourvu de structure communautaire.

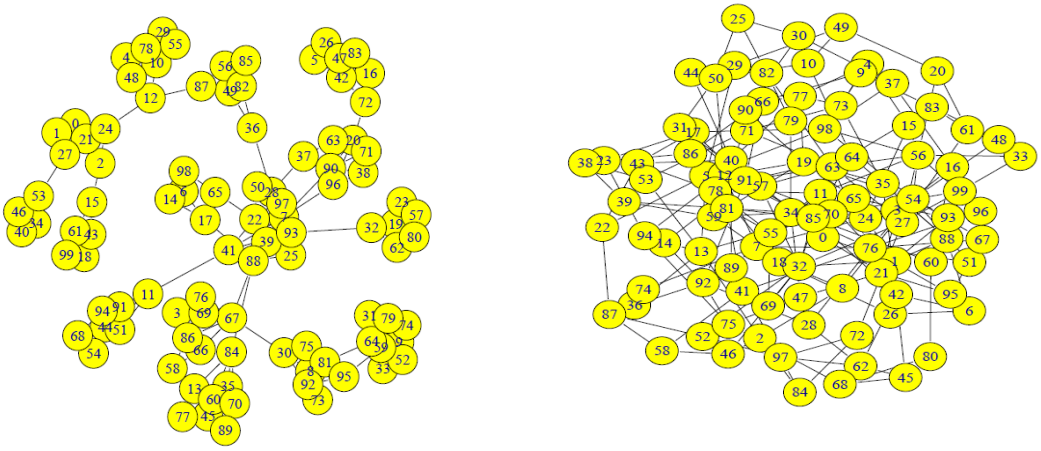


FIG. 42 – Réseaux artificiels générés avec le modèle de Lancichinetti et al. [107]. Réseau avec une structure communautaire (Gauche). Réseau sans structure communautaire (Droite).

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 [108].

6.1.2 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 intra communautaire 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.

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 [106] [109] 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 k_{int} et le degré total k du nœud considéré [110] :

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

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

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

$$\tilde{\rho}(C) = \rho(C)n_C = \frac{2m_C}{n_C - 1}$$

La densité normalisée (scaled density) $\tilde{\rho}$, 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 [111].

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 [111]. 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 [111].

6.2 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 [103] « The first problem in graph clustering is to look for a quantitative definition of community. No definition is universally accepted ». Pour formaliser le problème, on peut définir la détection de 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, on pourra se référer à [112] [103]. 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.

6.2.1 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 (Fast Greedy) [113] 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 $O(mn)$.

L'algorithme de Bondel et al. (Louvain) [114] 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, le nœud 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.

L'algorithme de Girvan et Newman (EdgeBetweenness) [110] 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 $O(m^2n)$.

L'algorithme de Radicchi et al. (***Radetal***) [115] 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 $O(m^2)$.

6.2.2 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) [116] 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 $O(n^3)$.

L'algorithme de Pons et Latapy (Walktrap) [117] 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 $O(mn\log(n))$.

L'algorithme de Rosvall et Bergstorm (Infomap) [118] 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.

6.2.3 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 de diverses façons afin d'aboutir à une partition adaptée.

L'algorithme de Newman (Leading Eigenvector) [119] 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.

L'algorithme de Donetti et Muñoz (Comfind) [120] 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 retenu.

6.2.4 Autres algorithmes

Optimisation par recuit simulé

Reichardt et Bornholdt ont proposé un algorithme d'optimisation appelé ***Spinglass*** (Verre de Spin) [121] basé sur une analogie entre le modèle de Potts appliqué aux verres de spin et la structure de la communauté. Le modèle de Potts est un modèle d'interaction dans un réseau dans lequel les spins peuvent prendre plusieurs états q. Les verres de spin sont des alliages métalliques comportant un petit nombre d'impuretés magnétiques disposées au hasard dans l'alliage. A chaque impureté est associé un spin. Le couplage entre ces différents spins peut être plus ou moins intense (attractif ou répulsif) en fonction de la distance qui les sépare. Ces interactions peuvent être modélisées par des fonctions d'énergie non convexe. Par analogie, on suppose que chaque nœud du graphe représente un verre de spin qui peut appartenir à l'une des q communautés possibles. On recherche alors la meilleure configuration pour étiqueter chacun des nœuds en minimisant une fonction d'énergie non convexe. Cette fonction comprend un terme d'adéquation aux données favorisant les nœuds voisins qui portent la même étiquette et un terme de pénalité pour les trop grandes communautés. Dans ce travail, la technique d'optimisation du recuit simulé est appliquée sur le modèle défini. Cette technique nécessite de se donner un majorant du nombre de communautés. Sa complexité est en $O((m+n)n)$.

L'algorithme de Raghavan (LabelPropagation) [122] utilise le concept de voisinage des nœuds et simule la diffusion d'information dans le réseau pour identifier les communautés. Initialement, chaque nœud est étiqueté avec une valeur unique. Puis, dans un processus itératif, chaque nœud prend l'étiquette qui est la plus répandue dans son voisinage. Ce processus se poursuit jusqu'à convergence, c'est à dire lorsque chaque nœud a l'étiquette majoritaire de ses voisins. Les communautés sont obtenues en considérant les groupes de nœuds avec la même étiquette.

Modularité

La modularité est la mesure de référence dans le contexte de la détection de communautés pour évaluer la qualité d'une partition du réseau. Elle a été introduite par Newman et Girvan [123]. Elle compare la proportion effective d'arêtes internes aux communautés à la proportion d'arêtes attendues dans l'hypothèse où les liens sont distribués de façon aléatoire :

$$Q = \sum_{C \in P} e(C) - \ a(C)^2$$

où $e(C)$ représente la proportion effective d'arêtes internes à une communauté C et $a(C)$ la proportion d'arêtes liées à une communauté C. Une arête est dite liée à une communauté si au moins l'une de ses extrémités appartient à la communauté. Si C est un ensemble aléatoire de nœuds et si les liens sont distribués de façon aléatoire, alors la proportion de liens internes attendue est $a(C)^2$. En effet, chacune des extrémités d'une arête interne a une probabilité $a(C)$ d'appartenir à la communauté.

La modularité est comprise dans l'intervalle [-1, 1]. Lorsque le premier terme est beaucoup plus grand que le second, cela signifie qu'il y a beaucoup plus de liens à l'intérieur d'une communauté que l'on pourrait attendre du hasard. Il y a donc de grandes chances que l'on ait effectivement détecté une communauté. Lorsqu'un réseau ne présente pas de structure communautaire ou lorsque les communautés ne sont pas meilleures qu'une partition aléatoire, la valeur de la modularité est négative ou nulle. En pratique, une modularité comprise entre 0,3 et 0,7 est considérée comme élevée [124]. La modularité n'est pas une mesure de performance mais plutôt une propriété des données, car elle ne permet pas de quantifier une bonne ou une mauvaise partition. Son principal avantage est qu'elle peut être calculée en utilisant la connectivité du réseau, en l'absence de toute étiquette de nœud ou de toute autre information. Elle s'est imposée comme un standard de facto pour mesurer la qualité d'une partition. Ceci commence à soulever quelques questions. Ainsi les travaux de [125] ont mis en évidence son inaptitude à détecter de petites communautés. En effet, la modularité est une somme de termes où chaque terme correspond à une communauté. Trouver la modularité maximale est alors équivalent à la recherche du compromis idéal entre le nombre de termes dans la somme, c'est à dire le nombre de communautés, et la valeur de chaque terme. Une augmentation du nombre de communautés ne correspond pas nécessairement à une augmentation de la modularité, car les communautés étant plus petites, chaque terme de la somme diminue.

La modularité a été adoptée pour une grande variété d'utilisations. Elle est utilisée pour la validation et la comparaison des structures communautaires. Elle constitue aussi un critère d'optimisation des algorithmes de détection de communautés. Elle a donné lieu à des solutions algorithmiques qui l'utilisent comme critère d'optimisation de façon explicite. Elle est par ailleurs utilisée pour choisir une partition dans les techniques de partitionnement hiérarchique. En effet, il existe une multitude de partitions en communautés possibles à partir du dendrogramme. Le choix parmi ces partitions se fait généralement par une fonction de qualité capturant les propriétés souhaitées. Le tableau 29 reporte les principales propriétés des algorithmes présentés précédemment tout en précisant l'usage qui est fait de la modularité le cas échéant. Ceci illustre bien la place prépondérante de la modularité dans la problématique de la découverte de communautés.

TAB. 29 – Principales propriétés des algorithmes de détection de communautés.

Nom	Classification Hiérarchique	Complexité	Paramètres requis	Utilisation de la modularité
Edge Betweenness	Oui	$O\ (m^2n)$	Aucun	Coupure du dendrogramme
Radetal	Oui	$O\ (m^2)$	Critère de transitivité	Coupure du dendrogramme
Fast Greedy	Non	$O\ (mn)$	Aucun	Optimisation
Louvain	Non	$O\ (n)$	Aucun	Optimisation
Spinglass	Non	$O\ ((m+n)n)$	Nombre de Spins	Optimisation
Eigenvector	Oui	N/A	Aucun	Coupure du dendrogramme
Comfind	Non	N/A	Nombre de vecteurs propres	Coupure du dendrogramme
Markov Cluster	Non	$O\ (n^3)$	Expansion/ Inflation	Coupure du dendrogramme
Walktrap	Oui	$O\ (mn\log(n))$	Longueur de la marche	Coupure du dendrogramme
Label Propagation	Non	$O\ (n)$	Aucun	Non
Infomap	Non	N/A	Aucun	Non

6.3 Mesures de performance des algorithmes

Pour évaluer les performances des algorithmes de détection de communautés, on considère usuellement la qualité de la partition sous l'aspect classification. Lorsque l'on dispose d'une vérité terrain, autrement dit lorsque l'on connaît les communautés réelles, on compare les communautés estimées par l'algorithme avec les communautés de référence. A cette fin, plusieurs mesures existent, la plupart d'entre elles reposent sur la matrice de confusion. L'élément m_{ij} de cette matrice représente le nombre de nœuds provenant d'une communauté de référence *i* qui sont affectés par l'algorithme à une communauté *j*. La matrice de confusion pour un réseau contenant *l* communautés de référence et un algorithme ayant estimé *k* communautés est représentée dans la figure 43.

		Estimation					
		1	i	2	...	k	Total
Référence	1		m_{11}	m_{12}	...	m_{1k}	m_{1+}
	j						
	2	m_{21}		m_{22}	...	m_{2k}	m_{2+}
	⋮	⋮		⋮	⋱	⋮	⋮
	l		m_{l1}	m_{l2}	...	m_{lk}	m_{l+}
	Total		m_{+1}	m_{+2}	...	m_{+k}	$m_{++} = n$

FIG. 43 - Une matrice de confusion.

n désigne le nombre total de nœuds dans le réseau, m_{i+} représente le nombre total de nœuds dans la communauté de référence et m_{+j} représente le nombre total de nœuds dans la communauté estimée. Cette matrice est généralement rectangulaire car l'algorithme n'estime pas forcément le nombre exact de communautés ($k \neq$ l).

Information mutuelle normalisée

La mesure de l'information mutuelle normalisée (IMN) est la métrique la plus couramment utilisée dans la littérature. Elle a été définie dans le contexte du clustering classique pour comparer deux partitions différentes d'un même ensemble de données. Elle s'exprime à partir des éléments de la matrice de confusion :

$$I = \frac{-2\sum_i \sum_j m_{ij} \log\left(n m_{ij} / m_{i+} m_{+j}\right)}{\sum_i m_{i+} \log(m_{i+}/n) + \sum_j m_{+j} \log\left(m_{+j}/n\right)}$$

Si les communautés estimées correspondent parfaitement aux communautés de référence, la mesure prend la valeur 1. Elle est nulle si les deux partitions sont indépendantes.

Proportion de nœuds bien classés

La proportion de nœuds bien classés a été utilisée par plusieurs auteurs [110]. Un nœud est considéré comme étant correctement classé si sa communauté de référence est identique à la moitié au moins des nœuds présents dans la même communauté estimée. Dans le cas où aucune communauté ne représente 50% des effectifs, tous les nœuds concernés sont considérés comme des nœuds mal classés. Le nombre total de nœuds correctement classés est normalisé par la taille du réseau afin d'obtenir une valeur comprise entre 0 et 1. Cette mesure est aussi déduite de la matrice de confusion.

L'indice de Rand

L'indice de Rand (IR) est un indice de concordance entre partitions [126]. Il est utilisé quand on dispose de deux partitions effectuées sur les mêmes individus par deux algorithmes pour savoir si ces deux partitions sont en accord ou bien si elles diffèrent significativement en un sens à préciser. Lorsque l'on croise deux partitions, on va s'intéresser aux paires d'individus qui restent ou ne restent pas dans les mêmes classes. Pour une paire donnée, il y a accord lorsque les deux nœuds appartiennent à la même communauté, ou à des communautés différentes dans les deux structures communautaires à comparer. Par conséquent, il y a désaccord si les nœuds sont dans la même communauté pour une partition et dans des communautés différentes pour l'autre. L'index de Rand est donné par :

$$IR = \frac{\sum_{tu}\binom{m_{tu}}{2}}{\binom{n}{2}}$$

où le numérateur correspond au nombre d'accords et le dénominateur correspond au nombre maximal d'accords possibles pour un réseau de taille *n*. La valeur de l'indice de Rand est de 1 pour des partitions identiques et de 0 pour des partitions indépendantes.

L'indice de Rand ajusté

L'indice de Rand ajusté (IRA) est une version corrigée de l'indice de Rand [127]. L'idée derrière cette correction est qu'une partie de l'accord mesuré par l'indice de Rand est due au hasard. Elle doit donc être soustraite afin d'obtenir la valeur de l'agrément indépendant du hasard. La mesure est définie par :

$$IRA = \frac{\sum_{tu}\binom{m_{tu}}{2} - \left[\sum_{t}\binom{m_{t+}}{2}\sum_{u}\binom{m_{+u}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{t}\binom{m_{t+}}{2} + \sum_{u}\binom{m_{+u}}{2}\right] - \left[\sum_{t}\binom{m_{t+}}{2}\sum_{u}\binom{m_{+u}}{2}\right]/\binom{n}{2}}$$

Le premier terme du numérateur correspond au nombre d'accords observés et le second au nombre d'accords dus au hasard. Le dénominateur permet de normaliser la mesure afin qu'elle varie de -1 à 1. La valeur nulle correspond à un accord de pur hasard. Les valeurs négatives traduisent la situation où l'accord observé est inférieur à celui qui serait dû au hasard. Les valeurs positives représentent le cas où l'accord observé est supérieur à celui dû au pur hasard.

6.4 Evaluation des algorithmes de détection

Etant donné le nombre sans cesse croissant d'algorithmes de détection de communautés, il est nécessaire d'établir une cartographie objective des performances des méthodes proposées. Pour ce faire, nous pouvons considérer deux voies complémentaires :

L'évaluation sur des réseaux réels qui suppose de connaître préalablement la structure communautaire du réseau. Déterminer les communautés de référence nécessite de faire appel à des experts. C'est une tâche qui peut s'avérer coûteuse et difficile lorsque la taille des réseaux est significative. Le principal avantage est que les performances sont déterminées sur des réseaux on ne peut plus réalistes. Néanmoins, les grands graphes de terrain ont des propriétés structurelles bien spécifiques et les performances d'un algorithme dans une situation donnée ne peuvent pas être extrapolées à d'autres réseaux aux propriétés topologiques différentes. Afin de pouvoir dresser des conclusions plus générales, il convient de disposer d'un échantillon représentatif étiqueté de grands graphes de terrain.

L'évaluation sur des réseaux artificiels qui suppose de disposer de générateurs de graphes à même de générer des échantillons de graphes réalistes. Les réseaux artificiels semblent surmonter les limitations des réseaux réels, car il est possible de générer aléatoirement plusieurs d'entre eux, tout en contrôlant leurs propriétés. Tout ce qui est nécessaire est un modèle capable de produire des réseaux avec des caractéristiques similaires à celles des réseaux du monde réel. Le modèle le plus populaire pour tester des algorithmes de détection de communautés a été défini par Newman et Girvan [110]. Cependant, bien que largement utilisé à des fins comparatives [110], [120], [112], [128], il est limité en termes de réalisme. En effet, il génère de petits réseaux avec des communautés de taille égale et des degrés de nœuds qui sont approximativement les mêmes. Pour pallier ces inconvénients, plusieurs variantes permettant de produire de plus grands réseaux et des communautés avec des tailles hétérogènes ont été définies [129]. Plus récemment, le modèle proposé par Lancichinetti et al. [107] permet de générer des graphes plus réalistes avec une distribution des degrés et la taille des communautés en loi de puissance. Par ailleurs, un paramètre spécifique permet de contrôler la netteté de la structure communautaire, c'est-à-dire le nombre de liens inter communautés. Cependant, ces générateurs ne visent pas à modéliser les processus de formation des grands graphes de terrain. Ils génèrent des graphes avec des propriétés similaires sans néanmoins atteindre le niveau de complexité des graphes réels. Les réseaux artificiels ne doivent donc pas être considérés comme un substitut aux réseaux du monde réel, mais plutôt comme un complément.

Les premières traces d'évaluation objective de la structure communautaire apparaissent dans [110]. Les auteurs utilisent des réseaux artificiels peu réalistes et des réseaux sociaux et biologiques de petite taille (< 300 nœuds) pour qualifier leur algorithme. Le nombre de communautés étant fixé, ils calculent la proportion de nœuds correctement classifiés. Dans [123] les auteurs comparent des méthodes hiérarchiques divisives en utilisant une méthodologie et des données comparables à celles de leurs travaux antérieurs.

Dans [130], l'auteur compare son algorithme (Walktrap) aux algorithmes EdgeBetweenness, Fast Greedy, Comfind, Markov Cluster ainsi qu'à quatre autres algorithmes que nous n'avons pas présentés. Il s'agit d'une version améliorée de Comfind [131], de Netwalk [132], une méthode hiérarchique basée sur le temps moyen d'atteinte d'un sommet par des marches aléatoires, de l'algorithme de Duch Arenas [133], qui utilise le recuit simulé pour optimiser la similitude et de Cosmoweb [134] basé sur une approche gravitationnelle. La comparaison porte sur des données artificielles et réelles. Pour les données artificielles, l'indice de performance utilisé est l'indice de Rand modifié, et la modularité est utilisée pour les données

réelles. Les expérimentations sur les données artificielles sont menées en faisant varier le nombre, la taille et la densité des communautés ainsi que la modularité espérée du graphe. Deux cas sont considérés en ce qui concerne la distribution de la taille des communautés, distribution uniforme et distribution en loi de puissance.
Dans le premier cas, l'auteur montre que Walktrap et Comfind sont les plus performants. Fast Greedy et Cosmoweb sont les moins performants. Les autres approches obtiennent des résultats intermédiaires. De plus, hormis pour Markov Cluster, Comfind et Walktrap, les performances diminuent lorsque la taille des graphes augmente. En ce qui concerne la taille des communautés, Comfind et sa version modifiée ainsi que Walktrap, trouvent des communautés de tailles comparables à celles qui sont simulées. Fast greedy a tendance à privilégier les grandes communautés alors que Cosmoweb, et dans une moindre mesure Markov Cluster, ont tendance à privilégier les très petites communautés, et ce visiblement de manière indépendante des tailles des communautés de la partition de référence. Ceci laisse à penser que ces approches possèdent une échelle intrinsèque à laquelle elles détectent des communautés. Elles peuvent ainsi obtenir de très bons résultats lorsque les communautés à détecter ont des tailles adaptées à leur échelle intrinsèque. A l'opposé, lorsque les communautés ne correspondent pas à leur échelle intrinsèque, ces algorithmes risquent de fournir une partition de mauvaise qualité.
Dans le second cas, cas où la taille des communautés est hétérogène, on observe trois comportements différents. Walktrap et EdgeBetweenness se comportent relativement bien, contrairement à Markov Cluster et Comfind dont les performances se dégradent sensiblement. Pour les autres algorithmes, les performances se dégradent dans des proportions plus raisonnables. Fast Greedy est le seul algorithme dont les performances s'améliorent.
Les comparaisons sur graphes réels font surtout apparaître que des partitions très différentes peuvent atteindre le même niveau de modularité. On peut donc penser que la modularité peut conduire à des partitions peu pertinentes, ce qui soulève la question de son utilisation comme mesure de qualité.

Dans [135], les auteurs utilisent trois réseaux réels de petite taille pour comparer Fast Greedy, Walktrap, Markov Cluster et un algorithme de marche aléatoire qu'ils proposent. Pour qualifier les performances des algorithmes, ils utilisent l'ensemble des mesures présentées précédemment (indice de Rand, indice de Rand modifié, Information Mutuelle Normalisée, proportion de bien classés) ainsi que la modularité. Ils mettent aussi en évidence des situations où la modularité maximale est obtenue avec une structure communautaire très différente de la réalité. Ils mettent ainsi en garde contre l'utilisation de la modularité comme critère pour l'évaluation et la comparaison des algorithmes de détection de communautés et préconisent d'utiliser une combinaison de mesures évaluant à la fois la topologie de la structure communautaire et les performances de classification objectives. Globalement, les mesures considérées sont en accord en ce qui concerne le classement des algorithmes. Leur algorithme se classe premier, suivi par Markov Cluster puis Fast Greedy, Walktrap fermant le rang. Notons que les différences ne sont pas très significatives, et ce d'autant plus que la taille des réseaux est très limitée.

Plus récemment, deux études [136], [137], relativement proches sont apparues dans la littérature. Elles utilisent toutes deux le modèle de génération de réseaux artificiels proposé par Lancichinetti et al. [107]. Parce que les graphes générés par ce modèle ont des propriétés

plus réalistes, il permet de revisiter le problème de la comparaison d'algorithmes de découverte de structure de communautés à partir de données artificielles. La mesure de performance utilisée dans les deux cas est l'Information Mutuelle Normalisée.

Dans [136], les auteurs évaluent les performances des algorithmes Fast Greedy, Infomap, Louvain, Walktrap et d'un algorithme basé sur une recherche de motifs locaux appelé CFinder [138] sur données réelles et artificielles. Sur les données artificielles, ils ont mené des expérimentations en faisant varier la taille des réseaux, le degré moyen des nœuds, la taille des communautés ainsi que la netteté de la structure communautaire. Leurs résultats montrent que les algorithmes ont un comportement très variable selon les propriétés des réseaux générés. De façon globale, Infomap, Louvain et Walktrap sont les plus performants. Lorsque les communautés sont moins nettes, Louvain a tendance à sur diviser les communautés alors que Walktrap a un comportement inverse. Ce comportement se retrouve pour Louvain et CFinder dans le cas des graphes peu denses. CFinder est particulièrement pénalisé lorsqu'il s'agit de trouver des grosses communautés. Louvain est lui moins à l'aise dans les situations où les graphes sont peu denses. Fast Greedy a toujours un comportement assez éloigné des autres algorithmes. Les résultats sur réseaux réels mettent en valeur des comportements très différents. Les différences sont de plus très variables en fonction des réseaux utilisés. Les algorithmes sont en accord lorsque les communautés sont relativement faciles à identifier (graphes peu denses avec de nombreuses petites composantes connexes), mais sont en désaccord lorsque les communautés sont plus difficiles à distinguer. Contrairement au cas des données artificielles, Fast greedy est plus en accord avec les autres algorithmes alors que CFinder a un comportement plus singulier.

Dans [137], les auteurs comparent Louvain, Fast Greedy, MarkovCluster, Infomap et Walktrap uniquement sur des données artificielles. Les expérimentations sont similaires à celle de Navarro. Les résultats de cette étude montrent que Walktrap et Infomap s'avèrent être les plus consistants. Ils produisent les meilleurs résultats pour toutes les situations étudiées. Infomap donne cependant généralement les meilleurs résultats, plus particulièrement lorsque les communautés sont moins nettement identifiées. Markof Cluster et Louvain donnent aussi des résultats satisfaisants. Fast Greedy donne les résultats les moins bons et semble dépassé par rapport aux algorithmes plus récents. Un point commun à tous les algorithmes est l'amélioration des performances avec l'élévation du degré. De plus, les performances de tous les algorithmes décroissent lorsque les communautés sont moins bien séparées. Les auteurs comparent par ailleurs les propriétés topologiques des communautés de référence avec celles des communautés estimées par les algorithmes. Les cinq propriétés présentées précédemment sont considérées (distribution de la taille des communautés, distribution de l'enracinement, densité normalisée, distance moyenne, dominance de hub). L'analyse de la distribution de la taille des communautés révèle qu'Infomap est proche de la référence tout en trouvant plus de petites communautés. Walktrap trouve la distribution la plus proche de la référence, bien que l'Information Mutuelle Normalisée soit plus basse que celle d'Infomap. En ce qui concerne la distribution de l'enracinement, c'est Infomap qui s'approche le plus de la référence. Quant aux autres paramètres (densité normalisée, distance moyenne et hub dominance), Infomap, Walktrap et Markov Cluster sont les plus proches de la référence. Bien que les auteurs aient pris soin de générer des réseaux dont les propriétés soient proches de celles observées dans les réseaux réels, ils en diffèrent néanmoins. Ainsi, les petites communautés sont organisées sous

forme de clique alors qu'elles sont plutôt organisées sous forme d'arbre dans les réseaux réels. De plus, la distance moyenne des petites communautés artificielles est plus petite que celle des petites communautés réelles. Ils concluent que globalement, Infomap est l'algorithme le plus performant. Il identifie des communautés dont les propriétés sont les plus proches de la référence. Les propriétés des communautés découvertes par Walktrap sont proches de la référence. Bien que plus performant au sens de l'Information Mutuelle Normalisée, Markov Cluster aboutit à une mauvaise distribution de la taille des communautés. Les communautés identifiées par Louvain sont les plus différentes de la référence.

6.5 Conclusion

Dans ce chapitre nous avons introduit la notion de structure communautaire dans les réseaux complexes. Nous avons présenté les mesures qui permettent d'apprécier les propriétés topologiques des communautés ainsi que les valeurs observées de ces mesures dans les grands graphes de terrain. Nous avons par ailleurs dressé un éventail des solutions proposées pour la détection de communautés. L'analyse des quelques travaux concernant la comparaison des algorithmes nous a permis de constater que l'évaluation objective de la qualité des algorithmes est un problème complexe pour lequel il n'existe pas de solution universelle. Bien que des tendances générales se dégagent, il semble préférable de ne négliger aucune solution. En effet, dans certaines situations où les communautés sont facilement identifiables, les performances des algorithmes convergent alors qu'elles sont très différentes dans d'autres situations. On remarque de plus, qu'extrapoler le comportement d'un algorithme sur des données artificielles à des données réelles n'est pas chose aisée. On aboutit parfois à des situations contradictoires, ce qui souligne la différence structurelle entre des réseaux dont les propriétés sont simulées et les réseaux réels. Un des acquis majeurs de l'étude bibliographique sur la comparaison des performances des algorithmes, est la confirmation qu'il convient d'observer simultanément les performances au sens usuel des mesures issues de la classification, tout en analysant les propriétés structurelles des communautés identifiées.

7. DETECTION DE COMMUNAUTES DANS LES RESEAUX D'INTERACTION DE SERVICES WEB

7.1 Introduction

Dans ce chapitre, nous présentons les résultats de l'analyse de la structure communautaire des réseaux d'interaction de services Web. Ce travail s'inspire des travaux de [48] dans lesquels les communautés sont des ensembles de services Web qui entrent souvent dans les mêmes compositions et donc qui interagissent de façon préférentielle. Notre objectif est d'identifier les algorithmes de détection de communautés qui soient adaptés au domaine spécifique des services Web. En effet, les études comparatives laissent supposer qu'il n'existe pas d'algorithme universel. Les performances des algorithmes de détection de communautés sont étroitement liées à la structure du graphe et par là même, à la nature des données traitées.

Dans une première partie, nous exposons la méthodologie adoptée pour conduire cette analyse. Nous présentons les collections choisies, l'ensemble des réseaux sur lesquels sont conduites les expérimentations, les algorithmes de détection de communautés retenus ainsi que les mesures de performance utilisées dans notre analyse. Nous spécifions également notre démarche d'analyse.
Dans la deuxième partie, nous fournissons des éléments de l'analyse topologique des réseaux étudiés issus des collections retenues. Ceci nous permet de comparer les collections et les propriétés topologiques des communautés identifiées avec celles des communautés réelles lorsque nous possédons cette référence.

La troisième partie est consacrée à la détection de communautés dans les réseaux de paramètres et d'opérations syntaxiques dont la structure communautaire est plus ou moins connue. Nous dressons une comparaison des résultats obtenus par les algorithmes sur ces réseaux, tant du point de vue de leurs performances que des propriétés structurelles des communautés détectées.
La quatrième partie est consacrée à l'évaluation des algorithmes sur les réseaux de paramètres et d'opérations syntaxiques et sémantique sans structure communautaire connue. Nous comparons les propriétés topologiques des communautés détectées par les différents algorithmes ainsi que la similitude des partitions mesurée par l'Information Mutuelle Normalisée (IMN). Nous établissons également le lien entre communauté et domaine.

7.2 Méthodologie

Afin de mener une étude comparative des algorithmes de détection de communautés, nous disposons de trois degrés de liberté : les collections de services Web, les algorithmes de détection proprement dit et les mesures de performances. Dans ce qui suit, nous précisons et justifions les éléments retenus dans le cadre de nos expérimentations. Nous présentons également la démarche suivie pour les expérimentations et l'analyse des résultats.

7.2.1 Recherche d'une collection

L'évaluation des performances des algorithmes de découverte de communautés nécessite de disposer d'une collection de services Web qui satisfasse les mêmes exigences que celles que nous avons préalablement définies pour l'analyse des réseaux d'interaction à savoir :

- Les services préférablement issus du monde réel doivent constituer un échantillon représentatif de l'espace des services utilisés dans le cadre de la composition.

- On doit pouvoir disposer des descriptions syntaxiques et sémantiques des mêmes services afin de conduire une étude comparative basée sur les types de descriptions.

Nous avons de plus une exigence supplémentaire :

- La structure communautaire du réseau construit à partir de la collection doit être connue. Autrement dit chaque nœud du réseau doit être identifié comme faisant partie d'une communauté.

A l'heure actuelle, il n'existe pas à notre connaissance, de collection qui permette de satisfaire toutes ces exigences. La collection ICEBE05 permet de les satisfaire partiellement. Dans cette collection, chaque service est mono opération. Autrement dit, les réseaux de services et d'opérations sont confondus. Elle ne permet de satisfaire que partiellement la première exigence. En effet, dans cette collection les données sont artificielles. De plus elle est purement syntaxique. Néanmoins, ces services artificiels ont été générés avec en vue des problématiques de composition et de découverte. En ce sens, les réseaux d'interaction générés à partir de cette collection sont bien plus spécifiques que les réseaux artificiels utilisés dans la littérature pour tester les algorithmes de détection de communautés. ICEBE05 est organisée en dix-huit jeux de test permettant d'évaluer différents scénarii de composition. Elle comprend un ensemble de requêtes et des sous-ensembles de descriptions syntaxiques qui diffèrent par la taille des solutions, le nombre de descriptions et le nombre de paramètres dans une description. Elles sont désignées en utilisant la terminologie suivante *CollectionX1-X2-X3*. X1 se rapporte au nombre minimal de services dans une solution de composition ; (1 signifie qu'il y en a entre 2 et 4, 2 signifie qu'il y en a entre 6 et 8). Ces deux ensembles contiennent chacun 9 sous-ensembles qui diffèrent par les deux derniers paramètres. X2 porte l'information de la taille du sous ensemble en nombre de descriptions ; (20 correspond à 3356, 50 correspond à 5356 et 100 correspond à 8356). Le troisième chiffre X3 indique le nombre de paramètres dans les descriptions, entrées et sorties confondues ; (4 signifie que le nombre de paramètres est compris entre 4 et 8, 16 signifie qu'il est compris entre 16 et 20 et 32 signifie qu'il est compris entre 32 et 36).
La collection peut satisfaire la dernière exigence qui est d'ailleurs la plus impérative pour pouvoir mener une analyse comparative des algorithmes. En effet, les sous-ensembles de cette collection fournissent des réseaux d'interaction qui sont organisés sous forme de composantes. Chaque composante constitue un ensemble de services qui interagissent de façon préférentielle. Nous pouvons ainsi considérer chaque composante comme une communauté.

Nous avons sélectionné le sous ensemble Composition-2-20-4 car les propriétés topologiques des réseaux d'opérations et de paramètres de ce sous ensemble ont été préalablement étudiés par Oh. Nous pouvons ainsi nous assurer de nos résultats expérimentaux en ce qui concerne l'analyse topologique de cette collection. Il contient 3356 descriptions ayant entre 4 et 8 paramètres.

La collection SAWSDL-TC1 satisfait quant à elle les exigences concernant la nature et la description des services. En effet, bien qu'en partie ré-échantillonnée, elle est composée de services réels. Nous disposons à la fois des descriptions syntaxiques et sémantiques pour cette collection. Par contre, nous ne disposons pas de vérité terrain en ce qui concerne la structure communautaire. Nous l'utilisons donc afin d'étudier l'existence éventuelle d'une structure communautaire dans les réseaux d'interaction de services réels. Elle permet aussi de pouvoir évaluer l'influence de la description sur la découverte de communautés.

En résumé, nous avons retenu la collection ICEBE05 afin de pouvoir comparer les algorithmes en ayant une référence sur la structure communautaire. La collection SAWSDL-TC1 nous permet, quant à elle, de valider ou non la structuration communautaire des réseaux d'interaction et de mettre en évidence les différences liées au type de description.

7.2.2 Extraction des réseaux

Dans ce travail, nous désirons essentiellement étudier la structure communautaire des réseaux d'interaction sans nécessairement évaluer toute la panoplie des réseaux que nous avons définis. Nous concentrons donc notre attention sur les réseaux de paramètres et d'opérations pour la collection ICEBE05. Le profil des réseaux extrait est donné sur la figure 44. On peut noter que pour cette collection, les noms des paramètres ont été générés aléatoirement. Ce sont des chaînes alphanumériques dépourvues de sens. On ne peut donc pas utiliser une fonction de mise en correspondance approximative pour ce type de nom.

Granularité	opération	paramètre
Mode d'invocation	total	
Description	syntaxique	syntaxique
Correspondance	**égal**	**égal**

FIG. 44 - Profils des réseaux d'interaction extraits de la collection ICEBE05 en vue de la détection de communautés.

En ce qui concerne la collection SAWSDL-TC1, nous extrayons à l'aide de WS-NEXT les quatre réseaux correspondant aux profils donnés dans la figure 45 afin de pouvoir en outre effectuer une comparaison entre les descriptions syntaxiques et sémantiques.

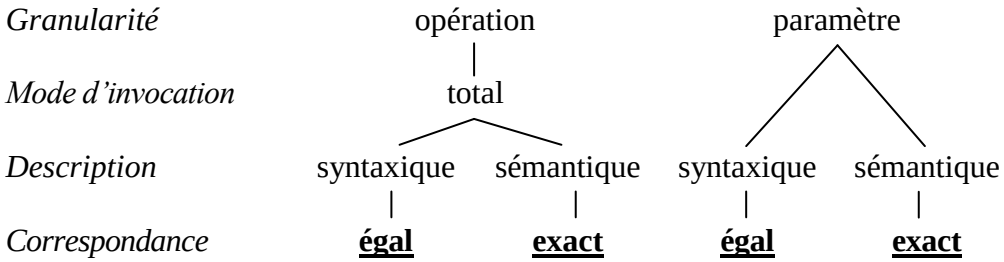


FIG. 45 - Profils des réseaux d'interaction extraits pour la collection SAWSDL-TC1 en vue de la détection de communautés.

7.2.3 Algorithmes retenus

Nous retenons un éventail d'algorithmes qui tienne compte à la fois d'un souci de représentativité des différentes catégories et des conclusions des travaux de comparaison étudiés. Pour la classe des algorithmes hiérarchiques agglomératifs, nous retenons Louvain pour son aspect plus stable que Fast Greedy face à la nature des données [136]. Il apparaît également bien plus performant que Fast Greedy sur données artificielles dans [137]. Pour les algorithmes hiérarchiques divisifs, nous retenons EdgeBetweenness qui se comporte relativement bien dans le cas de données artificielles où la taille des communautés est hétérogène [130]. Dans la classe des algorithmes basés sur les marches aléatoires, nous retenons Walktrap et Infomap qui présentent généralement tous deux de bonnes performances. Il est montré que Walktrap se comporte relativement bien sur données artificielles dans [130] bien qu'il ferme le rang dans [135] où rappelons-le, les réseaux réels considérés sont de petite taille. Walktrap et Infomap s'avèrent être les plus performants sur réseaux artificiels dans les comparaisons menées dans [136] et [137]. Pour la classe des algorithmes basés sur les propriétés spectrales nous utiliserons Eigenvector. En ce qui concerne les algorithmes non classifiés, nous retenons LabelPropagation et Spinglass.

7.2.4 Mesures de performance utilisées

Pour comparer deux partitions, nous utilisons l'information mutuelle normalisée (IMN), l'index de Rand (IR), l'index de Rand ajusté (IRA) et le pourcentage de nœuds bien classés (PBC).
Dans les expériences conduites sur les réseaux de la collection ICEBE05, nous comparons la partition obtenue à l'aide des algorithmes de détection de communautés avec la partition réelle. Pour les réseaux extraits de la collection SAWSDL-TC1, nous comparons entre elles les partitions obtenues par les différents algorithmes, afin de pouvoir juger de la concordance des ces partitions.
Dans tous les cas, nous calculons la modularité afin de pouvoir la mettre en parallèle avec les mesures de comparaison de partition.

7.2.5 Démarche d'expérimentation et d'analyse

Nous menons tout d'abord nos expérimentations sur les données artificielles issues de la collection ICEBE05. Comme nous le verrons par la suite, les réseaux issus de cette collection sont formés d'un ensemble de composantes.

Dans un premier temps, nous formons pour chacun des réseaux étudiés (paramètre, opération) un réseau global en connectant les composantes. Afin de pouvoir faire une comparaison objective des algorithmes, nous considérons ainsi que chaque composante est une communauté que les algorithmes doivent détecter.
Nous conduisons également une étude sur les propriétés structurelles des communautés identifiées dans ces réseaux. La comparaison entres les propriétés des communautés réelles et celles des communautés découvertes par les algorithmes nous permet d'aborder la notion de performance de manière plus qualitative. En effet, deux algorithmes peuvent générer des partitions très différentes et obtenir le même score en termes de mesures de performance. La comparaison des propriétés structurelles permet donc de discerner ces algorithmes. Cet aspect introduit une dimension supplémentaire dans la notion de performance. L'algorithme le plus performant est ainsi celui qui maximise les mesures de performances tout en minimisant la distance entre les propriétés structurelles des communautés réelles et estimées.

Dans un second temps, nous utilisons les algorithmes sur chacune des composantes afin d'évaluer le bien fondé de notre hypothèse et d'étudier leur comportement dans cette situation non maîtrisée.

Nous menons enfin nos expérimentations sur les données issues de la collection SAWSDL-TC1 afin de mettre en évidence la structure communautaire des réseaux d'interaction. Dans ces réseaux dont le partitionnement n'est pas connu à priori, nous comparons les propriétés topologiques des communautés identifiés par les algorithmes utilisés pour un type de description donné. Nous comparons par ailleurs ces propriétés en fonction du type de description et de la granularité du réseau.

7.3 Caractéristiques topologiques des réseaux

7.3.1 Présentation des réseaux ICEBE05

Les figures 46 et 47 représentent respectivement les réseaux élagués (sans nœuds isolés) de paramètres et d'opérations extraits à partir de la collection ICEBE05. Tous deux présentent une structure en composantes. On recense onze composantes pour le réseau de paramètres et douze pour le réseau d'opérations, dont deux de taille négligeable (taille 2). Ces composantes sont néanmoins très différentes. En effet, le réseau de paramètres contient des composantes de petite taille relativement denses alors que celles du réseau d'opérations sont de plus grande taille et peu dense.

Caractéristiques des composantes

Les tableaux 30 et 31 récapitulent respectivement l'ensemble des caractéristiques topologiques de base des composantes pour le réseau des paramètres et pour le réseau d'opérations. Nous pouvons remarquer qu'au sein d'un même réseau, les composantes sont structurellement très homogènes.
Dans les deux cas, la distribution de la taille des communautés est quasi uniforme. Les valeurs du degré moyen des composantes du réseau de paramètres sont très supérieures à celles des composantes du réseau d'opérations. Elles sont en moyenne dans un rapport sept. Ces deux réseaux se démarquent aussi au niveau de la transitivité. Aucune des composantes du réseau

d’opérations ne comporte de triangle. En effet les opérations sont représentées sous la forme de flux d’activités qui représentent des compositions. Les réseaux de paramètres sont quant à eux plus transitifs.

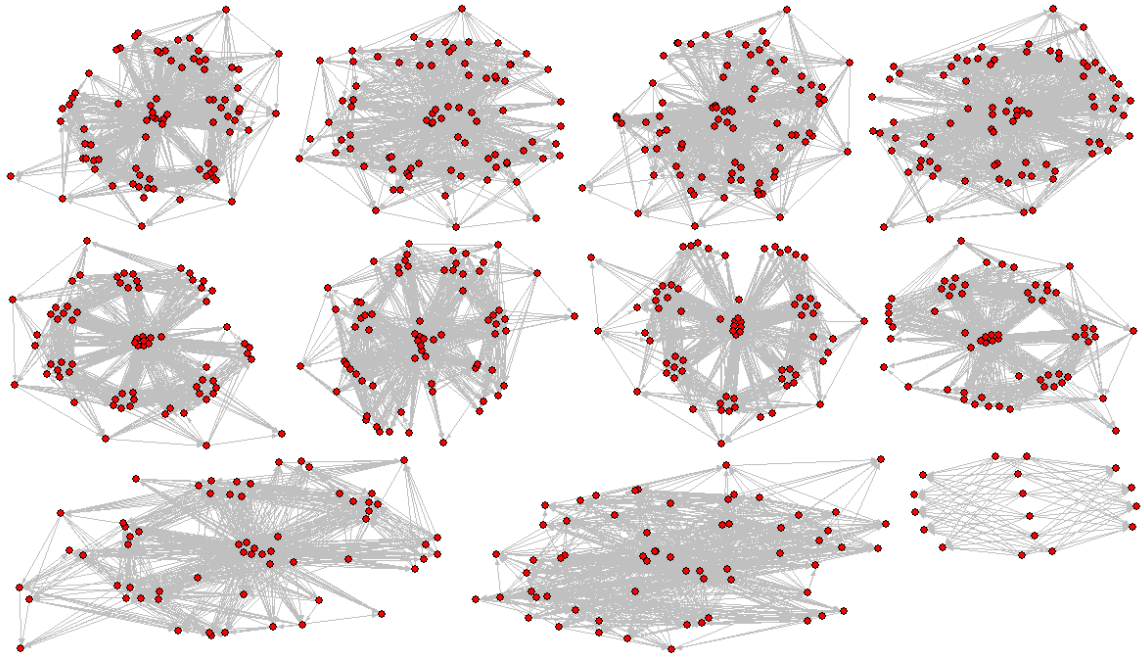


FIG. 46 – Réseau d’interaction de paramètres extrait à partir du sous-ensemble Composition-2-20-4 de la collection ICEBE05. Les nœuds isolés ne sont pas représentés.

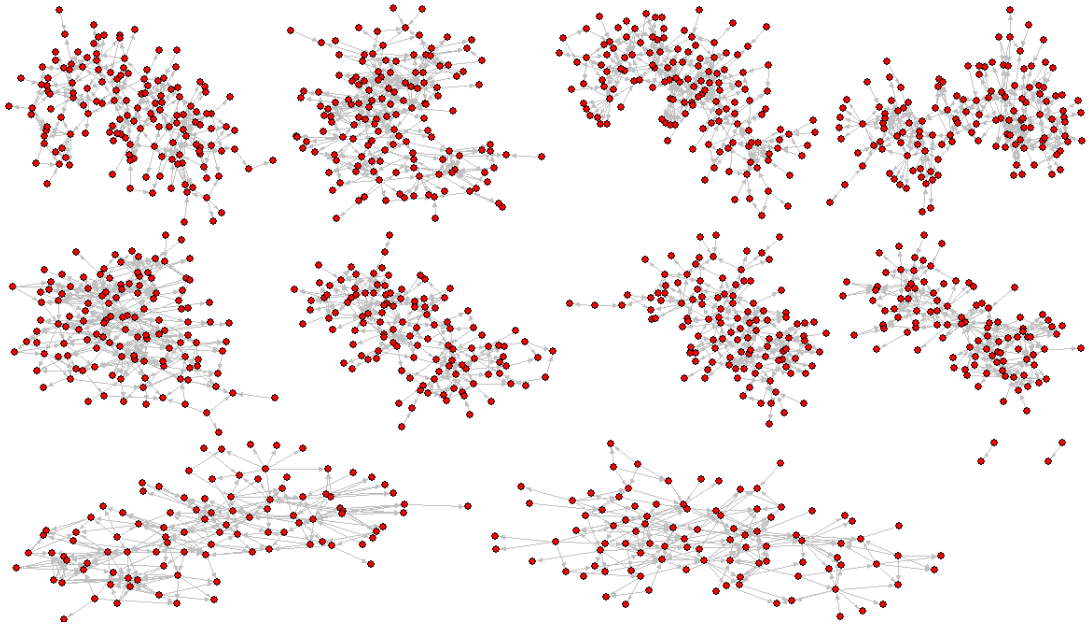


FIG. 47 – Réseau d’interaction d’opérations extrait à partir du sous-ensemble Composition-2-20-4 de la collection ICEBE05. Les nœuds isolés ne sont pas représentés.

TAB. 30 – Propriétés des composantes du réseau des paramètres de la collection ICEBE05. Taille *n*, nombre d'arêtes *m*, densité *d*, degré moyen <*k*>, transitivité *C*.

	1	2	3	4	5	6	7	8	9	10	11
n	80	80	80	80	71	71	71	62	62	62	17
m	1130	1122	1104	1089	936	931	888	831	777	748	72
<k>	28,47	28,05	27,6	27,22	36,36	26,22	25,01	26,8	25,06	25,6	8,00
C	0,38	0,35	0,36	0,35	0,35	0,35	0,31	0,41	0,36	0,32	0

TAB. 31 – Propriétés des composantes du réseau des opérations de la collection ICEBE05. Taille *n*, nombre d'arêtes *m*, densité *d*, degré moyen <*k*>, transitivité *C*.

	1	2	3	4	5	6	7	8	9	10
n	96	137	100	141	142	118	149	116	101	126
m	225	287	235	303	308	231	249	227	229	336
<k>	4,21	4,18	4,7	4,29	4,33	3,91	3,34	3,91	4,53	5,33
C	0	0	0	0	0	0	0	0	0	0

Distribution des degrés dans les composantes

résultats de l'étude de la distribution des degrés de chacune des composantes du réseau des paramètres et du réseau d'opérations sont reportés respectivement dans les tableaux 32 et 33. Les valeurs relativement faibles des probabilités associées au test d'adéquation laissent supposer que les composantes du réseau d'opérations ne suivent pas une loi de puissance. Pour ce qui est du réseau de paramètres, les résultats sont plus mitigés. L'hypothèse d'une distribution en loi de puissance semble plausible pour les degrés sortants.

TAB. 32 – Distribution des degrés des composantes du réseau des paramètres de la collection ICEBE05. Exposant de la loi de puissance estimée/valeur-p du test d'adéquation pour les degrés entrants (in), sortants (out), globaux (all) pour les composantes numérotées de 1 à 11.

	1	2	3	4	5	6	7	8	9	10	11
in	2,5/0,31	2,59/0,33	2,63/0,37	2,67/0,36	2,73/0,37	2,74/0,35	2,9/0,41	2,63/0,34	2,83/0,34	2,6/0,40	-
out	3,12/0,45	3,5/0,61	2,5/0,56	2,9/0,47	3,5/0,61	3,5/0,61	3,5/0,62	3,5/0,56	3,17/0,43	3,5/0,61	-
all	3,5/0,2	3,03/0,22	3,09/0,26	3,14/0,25	3,24/0,26	3,27/0,22	3,5/0,28	3,1/0,26	3,41/0,24	3,5/0,32	-

TAB. 33 – Distribution des degrés des composantes du réseau des opérations de la collection ICEBE05. Exposant de la loi de puissance estimée/valeur-p du test d'adéquation pour les degrés entrants (in), sortants (out), globaux (all) pour les composantes numérotées de 1 à 10.

	1	2	3	4	5	6	7	8	9	10
in	3,5/0,12	3,5/0,07	3,5/0,11	2,2/0,14	2,6/0,11	3,5/0,11	3,5/0,08	3,5/0,08	3,5/0,11	3,5/0,11
out	2,0/0,15	2,5/0,12	2,1/0,12	2,7/0,11	2,2/0,1	1,9/0,17	2,32/0,12	3,5/0,17	2,1/0,11	1,91/0,2
all	2,9/0,13	3,5/0,1	3,1/0,12	2/0,13	3,5/0,06	3,5/0,1	3,5/0,09	2,2/0,13	2,4/0,13	3,5/0,1

7.3.2 Propriétés topologiques des réseaux ICEBE05 et SAWSDL-TC1

Afin de pouvoir évaluer les algorithmes de détection, nous formons à partir des composantes, un réseau connexe. Pour ce faire, nous rajoutons aléatoirement dix liens entre chaque

composante dans le réseau des opérations et cinquante liens pour les réseaux de paramètres. Cette différence tient au fait que le réseau de paramètres est beaucoup plus dense. Si le nombre de liens n'est pas suffisant, tous les algorithmes n'ont aucune difficulté à détecter les composantes comme des communautés différentes. Dans ce cas, on ne peut donc pas les différencier. Dans le tableau 34, nous reportons les propriétés topologiques des deux réseaux connexes ainsi formés à partir de la collection ICEBE05, ainsi que celles des quatre réseaux issus de la collection SAWSDL-TC1. Nous n'avons pas reporté les valeurs estimées de la loi de puissance dans ce tableau lorsque les valeurs de probabilités associées au test d'adéquation sont trop faibles pour accepter l'hypothèse que la distribution des degrés suive une loi de puissance. Tous ces réseaux possèdent la propriété petit monde (distance moyenne faible). Ils ne sont pas (ou faiblement) transitifs. On observe de grandes différences pour ce qui est du degré moyen entre réseaux similaires des deux collections. Seul le réseau des paramètres de la collection SAWSDL-TC1 possède la propriété sans échelle. Tout ceci nous permet de conclure que, bien que partageant quelques caractéristiques communes, les réseaux construits à partir des deux collections sont structurellement assez différents. Ces différences font qu'il est nécessaire de prendre des précautions afin d'extrapoler les conclusions tirées de l'analyse des performances des algorithmes de détection de communautés d'une collection à l'autre. En effet, on ne peut exclure que certains algorithmes soient relativement sensibles à ces différences.

TAB. 34 – Propriétés topologiques des réseaux utilisés pour la détection de communautés.

Réseau	Taille n	Degré Moyen <k>	Distance Moyenne L	Transitivité C	Exposant Loi puissance γ	Corrélation des degrés
ICEBE (OH) Para-Synt	736	26,18	2,3	0,3188	-	-
SAWSDL-TC1 Para-Synt	385	2,3	2,75	0,039	3,15/2 ,01	-0,21
SAWSDL-TC1 Para-Sema	357	2,3	1,97	0,03	2,99/3,45	-0,22
ICEBE Oper-Total	1229	4,24	3,25	0	-	-
SAWSDL-TC1 Oper-Total-Synt	785	9,28	2,19	0,032	-	-0,45
SAWSDL-TC1 Oper-Total-Sema	785	10,04	1,87	0,022	-	-0,51

7.4 Comparaison des algorithmes sur les réseaux de la collection ICEBE05

Tester un algorithme de détection de communautés sur des réseaux dont le partitionnement est connu est une étape essentielle. L'utilisation de données artificielles permet de construire de tels réseaux. Idéalement, ceux-ci doivent avoir des propriétés proches des réseaux réels. L'étude des propriétés topologiques des réseaux extraits à partir de la collection ICEBE05 a mis en valeur les points de divergence et de convergence de cette collection artificielle avec la collection SAWSDL-TC1, que nous pouvons qualifier de réelle. Bien que ces réseaux ne soient pas équivalents, il est néanmoins intéressant de mesurer l'efficacité des algorithmes

dans cette situation. Cela nous apporte en effet certainement plus d'information que si les données artificielles étaient indépendantes de notre domaine d'étude.
Dans un premier temps, nous présentons les résultats de l'analyse menée sur le réseau global rendu connexe par l'ajout de liens entre les composantes de façon aléatoire.
Nous étudions ensuite le comportement des algorithmes sur les composantes prises séparément. Dans un dernier temps, à partir des partitions obtenues sur le réseau global, nous comparons les propriétés topologiques des communautés de référence (les composantes originelles) avec celles qui sont détectées par les algorithmes. Ceci nous permet d'avoir un éclairage supplémentaire sur la qualité comparée des algorithmes étudiés.

7.4.1 Réseau des opérations

Détection de communautés dans le réseau global

Dans ce cas, la structure communautaire de référence représente la situation où chaque composante forme une communauté distincte. La détection des communautés est effectuée par les algorithmes retenus sur le réseau d'opérations global obtenu en reliant les communautés. Dans un premier temps, on compare les performances des algorithmes indépendamment du nombre de communautés détectées.
Dans un second temps, nous imposons tant que faire se peut le nombre de communautés à détecter. Ceci nous permet de nous assurer que les algorithmes sont bien à même de découvrir la structure en composantes du réseau

Nombre de communautés variable
Le tableau 35 reporte les valeurs des mesures de performance obtenues sur le réseau d'opérations global sans contrainte sur le nombre de communautés à détecter. On remarque qu'à ce niveau, les algorithmes ont des comportements extrêmement variables. Alors que pour l'algorithme LabelPropagation le réseau est composé d'une seule communauté, Eigenvector en détecte soixante-dix-huit. Globalement, le nombre moyen de communautés évolue autour de 28 avec un écart type de 35. Ces valeurs reflètent bien la grande disparité des estimations.
Si l'on observe les valeurs des mesures de performance, on peut noter que les valeurs de l'indice de Rand sont généralement très élevées. Ce qui peut conduire à conclure que les algorithmes sont très performants. Nous avons aussi reporté dans ce tableau les rangs accordés à chacun des algorithmes par les mesures de performances. Cela permet de mettre en valeur leur jugement concordant pour les quatre algorithmes les moins performants. Ainsi, elles s'accordent toutes à donner le plus mauvais score à LabelPropagation suivi par Eigenvector, Infomap et finalement EdgeBetweenness. Pour les algorithmes restants, les avis divergent. On peut remarquer par ailleurs que, bien que les valeurs de l'indice de Rand et celles de l'indice de Rand ajusté soit très différentes, elles aboutissent au même classement. Il en est d'ailleurs de même pour le pourcentage de bien classés et l'information mutuelle normalisée. Le calcul du rang moyen permet néanmoins de départager les algorithmes. A ce titre, on peut considérer que Walktrap est le plus performant suivi par Spinglass, Louvain fermant la marche.

Rappelons que la modularité mesure plutôt la cohésion des partitions et non pas la similitude entre les partitions opérées par les algorithmes et les partitions de référence. Nous avons néanmoins reporté les valeurs mesurées de cette propriété de la structure communautaire avec

celles des indices de performance, ceci à titre comparatif. En effet, elle est souvent utilisée comme un critère de performance ou d'optimisation des algorithmes de détection de communautés. Il est donc intéressant d'étudier la corrélation pouvant exister entre cohésion et qualité de la partition. Si l'on compare le classement des algorithmes selon la modularité avec celui obtenu à partir des mesures de performance, on remarque qu'elle aboutit au même constat que celle-ci pour les trois algorithmes les moins performants. Si l'on intègre le score de la modularité avec celui des mesures de performance, l'ordonnancement des algorithmes est pratiquement inchangé. Walktrap et Spinglass obtiennent dans ce cas des scores identiques.

TAB. 35 – Détection de communautés sur le réseau d'interaction d'opérations ICEBE05. Nombre de communautés, modularité et mesures de performance des algorithmes : valeur des mesures de performance et classement effectué à partir de ces valeurs.

		VALEURS					RANGS				
	Nb com	MOD	IMN	PBC	IR	IRA	MOD	IMN	PBC	IR	IRA
EDGEBETWEENNESS	26	0,654	0,761	0,585	0.927	0.518	2	4	4	4	4
LOUVAIN	17	0.664	0.854	0.686	0.956	0.721	4	2	2	3	3
SPINGLASS	22	0,672	0,842	0,672	0.965	0.782	1	3	3	1	1
EIGENVECTOR	78	0,467	0,635	0,331	0.906	0.286	6	6	6	6	6
WALKTRAP	28	0,652	0,859	0,753	0.960	0.746	3	1	1	2	2
INFOMAP	62	0.625	0.741	0.382	0.925	0.397	5	5	5	5	5
LABELPROPAGATION	1	0	0	0,121	0.100	0	7	7	7	7	7

Nous reportons dans le tableau 36, les valeurs de la matrice de corrélation entre les mesures de performance en y incluant aussi la modularité. Les valeurs élevées des éléments de cette matrice symétrique permettent d'apprécier le consensus entre toutes ces mesures. Nous avons aussi calculé la valeur moyenne des coefficients de corrélation et l'écart type pour chacune d'entre elles. Ceci nous permet d'apprécier le degré de consensus d'une mesure avec l'ensemble des autres. Il apparaît ainsi que la modularité a un comportement global très proche de l'ensemble des mesures de performance. Cette constatation tendrait à justifier

Tab 36 – Matrice des coefficients de corrélation entre les mesures de performance incluant la modularité. Les deux dernières lignes correspondent respectivement à la valeur moyenne et à l'écart type de la corrélation d'une mesure avec l'ensemble des autres mesures.

	IMN	PBC	IR	IRA	MOD
IMN	1	0,872	0,982	0,964	0,991
PBC	0,872	1	0,766	0,981	0,852
IR	0,982	0,766	1	0,796	0,970
IRA	0,964	0,981	0,796	1	0,875
MOD	0,991	0,852	0,970	0,875	1
Moyenne	0,962	0,894	0,903	0,923	0,937
Ecart Type	0,052	0,096	0,112	0,085	0,068

l'utilisation de la modularité comme mesure de performance. Elle est néanmoins à manipuler avec précaution. En effet, cette convergence entre mesures de performance et modularité n'est peut-être pas reproductible dans d'autres situations expérimentales.

Nombre de communautés fixé

Dans cette partie, nous contraignons les algorithmes à identifier autant de communautés que de composantes, afin de pouvoir juger de leur aptitude à mettre en évidence la structure communautaire élaborée à partir de celles-ci. Cette expérimentation est menée avec les seuls algorithmes pour lesquels nous pouvons fixer le nombre de communautés. Il s'agit d'EdgeBetweenness, Spinglass, Eigenvector et Walktrap.

Le tableau 37 rassemble les résultats sur la modularité et les mesures de performance des algorithmes dans cette situation. Notons que Louvain, Infomap et LabelPropagation ne peuvent pas être contraints. On peut tout d'abord remarquer que seul Spinglass améliore ses performances. Pour les trois autres algorithmes, les performances se dégradent de façon significative. L'explication tient au fait que Spinglass est un algorithme d'optimisation qui recherche la meilleure partition pour un nombre de communautés donné, alors que les autres algorithmes utilisent une approche de classification hiérarchique. La coupure du dendrogramme, lorsque l'on contraint le nombre de communautés, ne correspond alors à aucun critère d'optimisation. Ce qui se traduit par une dégradation des performances. Notons d'ailleurs, qu'outre les mesures de performance, les valeurs de la modularité sont aussi très affectées par cette contrainte. Ces résultats suggèrent que les techniques qui utilisent une approche hiérarchique ont tendance à surestimer le nombre de communautés. Cela suppose que les composantes ne se subdivisent pas elles-mêmes en communautés.
Nous avons aussi déterminé la corrélation entre les mesures de performance et la modularité. Globalement, on retrouve le même comportement que précédemment. A quelques variations près, les mesures de performance et la modularité sont très consensuelles pour qualifier la qualité des algorithmes évalués.

TAB. 37 – Détection de communautés sur le réseau d'interaction d'opérations ICEBE05. Nombre de communautés, modularité et mesures de performance des algorithmes : valeur des mesures de performance et classement effectués à partir de ces valeurs. La dernière colonne correspond au classement moyen.

	VALEUR					RANG					
	Mod	IMN	PBC	IR	IRA	Mod	IMN	PBC	IR	IRA	Moy
EDGEBETWEENNESS	0,475	0,587	0,403	0.692	0.186	2	2	2	3	2	2,2
SPINGLASS	0,674	0,907	0,868	0.973	0.845	1	1	1	1	1	1
EIGENVECTOR	0,332	0,284	0,256	0.695	0.066	3	4	4	2	4	3,4
WALKTRAP	0,294	0,459	0,352	0.498	0.115	4	3	3	4	3	3,4

Détection de communautés dans les composantes

Afin de déterminer si les composantes possèdent une structure communautaire, nous avons appliqué les cinq algorithmes les plus performants sur chacune des dix composantes du réseau d'opérations. La figure 48 représente le nombre de communautés détectées par chacun des

algorithmes dans les communautés numérotées de 1 à 10. On observe trois types de comportements très différenciés. Infomap a tendance à détecter peu ou pas de communautés dans les composantes. Ainsi, hormis pour la composante 6, le nombre de communautés détectées évolue entre 1 et 4. Walktrap et EdgeBetweenness identifient systématiquement un grand nombre de communautés (de 8 à 16). Spinglass et Louvain ont un comportement médian. Ils détectent tous deux entre 6 et 8 communautés dans chacune des composantes.

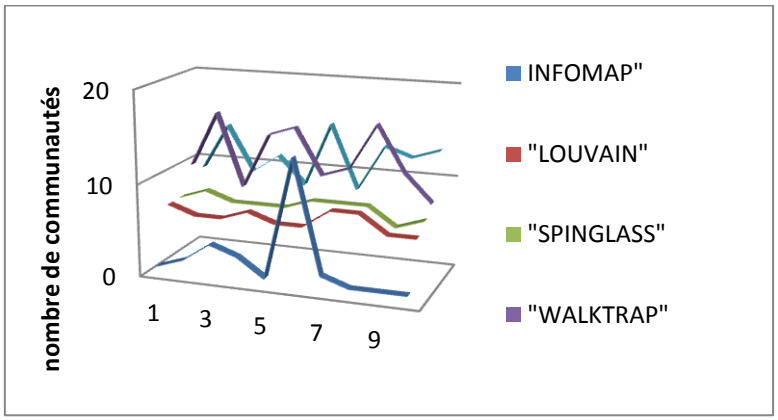


FIG. 48 - Nombre de communautés détectées dans chacune des dix composantes du réseau d'opérations ICEBE05.

Nous avons relevé la valeur de la modularité associée aux structures communautaires découvertes par les algorithmes dans chacune des composantes. Les résultats sont reportés dans la figure 49. Infomap se distingue des quatre autres algorithmes. La modularité associée à ses partitions est très faible. Elle est en étroite corrélation avec le nombre de communautés détectées. Les valeurs de la modularité évoluent dans une fourchette étroite pour les quatre autres algorithmes (entre 0,5 et 0,7). Spinglass présente toujours le plus haut niveau de modularité. Louvain est très légèrement en dessous de Spinglass. Il est suivi par Walktrap puis EdgeBetweenness avec lesquels les différences sont plus marquées. Bien que le nombre de communautés soit très variable, les valeurs de la modularité laissent à penser que les algorithmes aboutissent à des communautés dont la cohésion est globalement comparable.

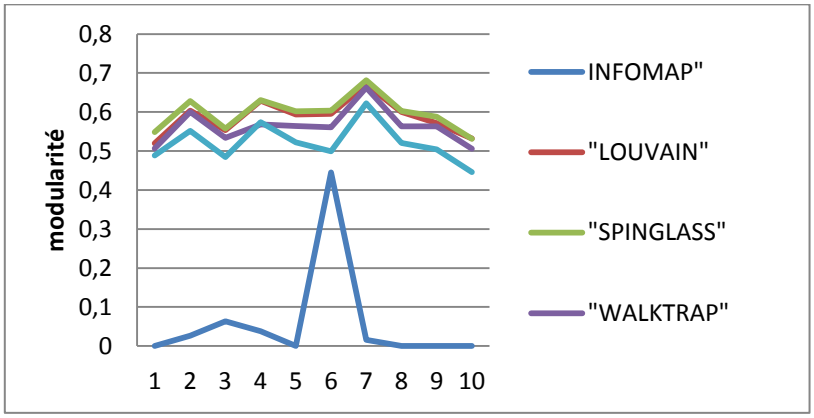
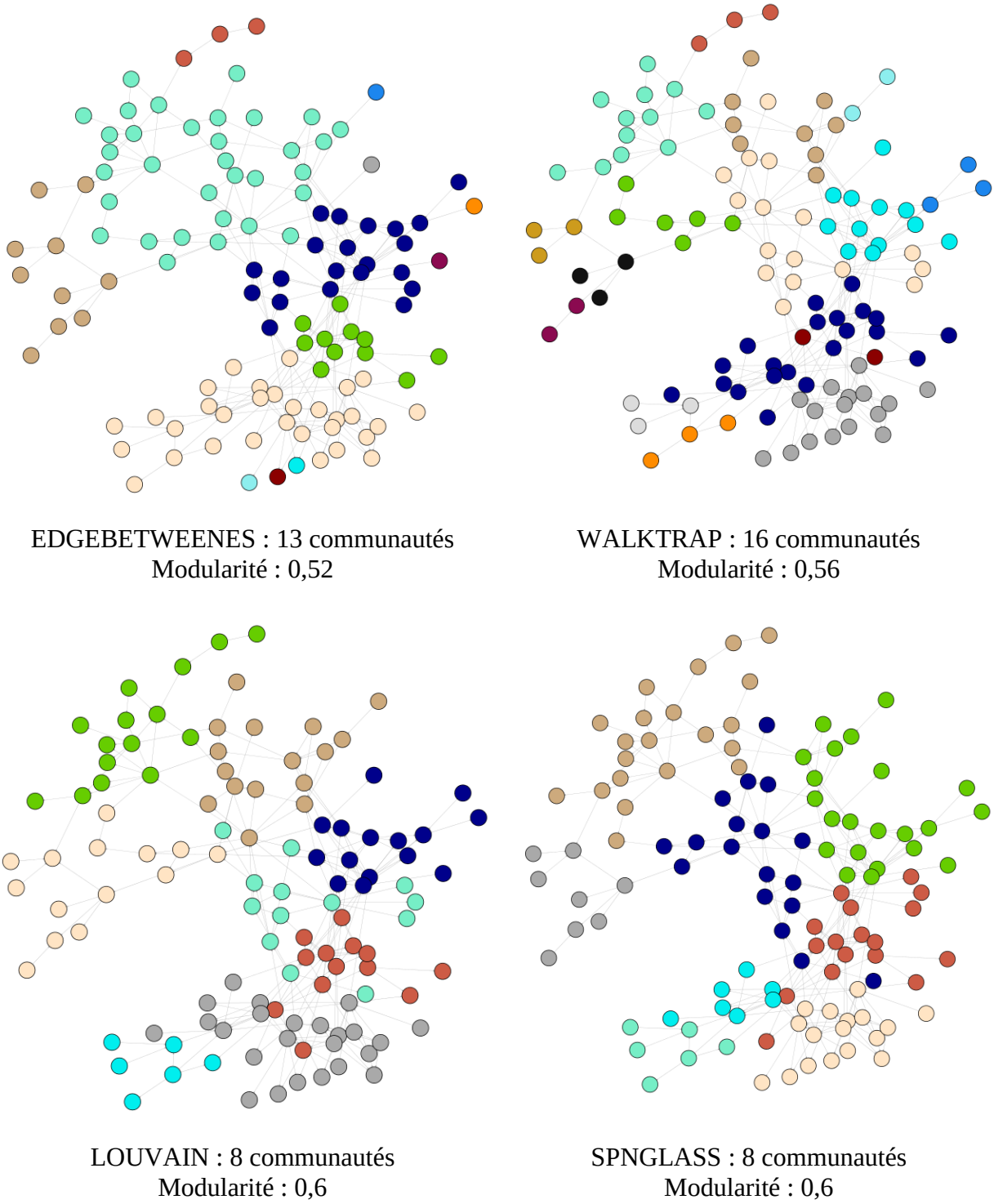


FIG. 49 - Modularité associée à la structure communautaire détectée dans les composantes 1 à 10 par les algorithmes Infomap, Louvain, Spinglass, Walktrap, EdgeBetweenness.

Pour examiner plus finement les différences entre ces algorithmes, nous avons représenté sur la figure 50, les communautés identifiées dans la composante 8 du réseau d'opérations. Pour celle-ci, Infomap ne détecte aucune communauté. EdgeBetweenness et Walktrap comptent respectivement 13 et 16 communautés. Les deux autres algorithmes partitionnent quant à eux la composante en 8 parties. Cette situation illustre bien les différences de comportement des algorithmes.



EDGEBETWEENES : 13 communautés
Modularité : 0,52

WALKTRAP : 16 communautés
Modularité : 0,56

LOUVAIN : 8 communautés
Modularité : 0,6

SPNGLASS : 8 communautés
Modularité : 0,6

FIG. 50 - Structure communautaire identifiée sur la composante 8 du réseau d'opérations de la collection ICEBE05.

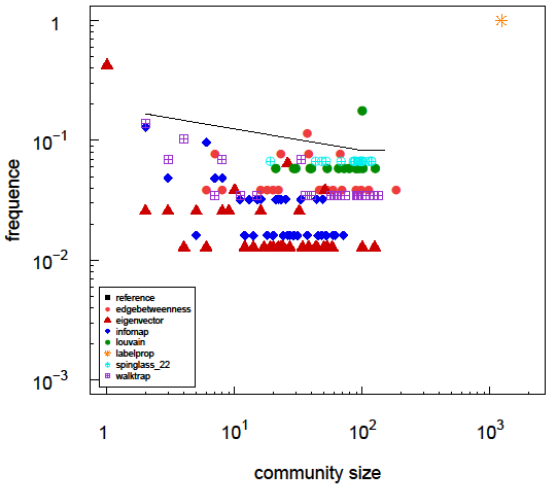
Spinglass et Louvain ont tendance à partitionner les composantes en communautés de taille comparable. Bien qu'aboutissant au même nombre de communautés, les partitions obtenues sont très différentes. Walktrap et EdgeBetweenness partitionnent les composantes de façon moins uniforme. Ainsi, dans l'exemple présenté, les 13 communautés d'EdgeBeetweenness se répartissent en deux grosses communautés (20 - 30 nœuds), trois communautés de taille moyenne (10 nœuds) et un ensemble de communautés de petite taille. De même, Walktrap répartit ses 16 communautés en 6 communautés de tailles moyennes et 10 autres de tailles plus petites. Tout ceci illustre bien qu'à modularité comparable, on peut avoir des structures communautaires très différentes. Cette constatation qui corrobore d'ailleurs les observations de Pons, montre bien que la modularité ne peut être considérée comme une mesure de qualité de partition.

Propriétés structurelles des communautés

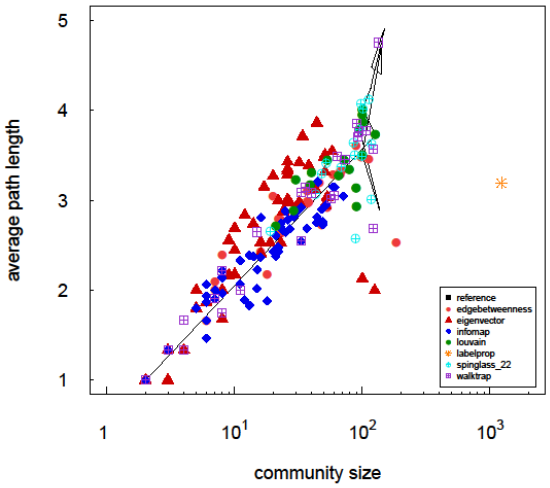
A ce niveau, le but visé est de comparer les propriétés structurelles des communautés identifiées par les algorithmes de découverte avec celles des communautés de référence. Cette comparaison ne peut s'opérer que sur le réseau global car nous n'avons aucune information en ce qui concerne la structure communautaire des composantes. La figure 51 représente la distribution de la taille des communautés ainsi que celle de l'enracinement. L'évolution de la distance moyenne, de la dominance de hub, de la densité normalisée en fonction de la taille des communautés sont aussi reportées.

Toutes ces mesures concernent les communautés identifiées par chacun des sept algorithmes étudiés, sans contrainte sur le nombre de communautés à détecter. Elles concernent également les valeurs mesurées dans les 10 composantes qui ont été utilisées pour former le réseau global. Au vu des résultats, on peut distinguer deux catégories d'algorithmes. La première catégorie regroupe les trois algorithmes les plus performants mais dans un ordre différent. Pour toutes les propriétés étudiées, Spinglass est celui qui identifie des communautés dont les propriétés topologiques sont les plus proches de celles des communautés de référence. Vient ensuite Louvain suivi de Walktrap. Ce qui pénalise Walktrap, c'est son aptitude à identifier plus de petites communautés. La deuxième catégorie regroupe les quatre algorithmes restants. Les communautés qu'ils mettent à jour ont des propriétés topologiques très hétérogènes, alors que les propriétés des composantes du réseau de référence sont très homogènes.

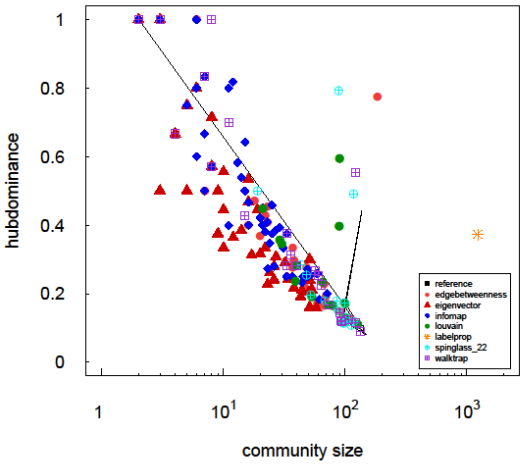
A partir de toutes ces analyses, il apparaît qu'un groupe de trois algorithmes se détache à savoir Spinglass, Louvain et Walktrap. Ils se distinguent par le fait que les deux premiers ont tendance à identifier moins de communautés mais de tailles comparables, alors que Walktrap détecte plus de communautés de tailles très différentes. Les composantes du réseau d'opérations étant très homogènes du point de vue des propriétés topologiques et peu denses, on peut donc conclure que ces trois algorithmes sont particulièrement adaptés dans ce cas de figure.



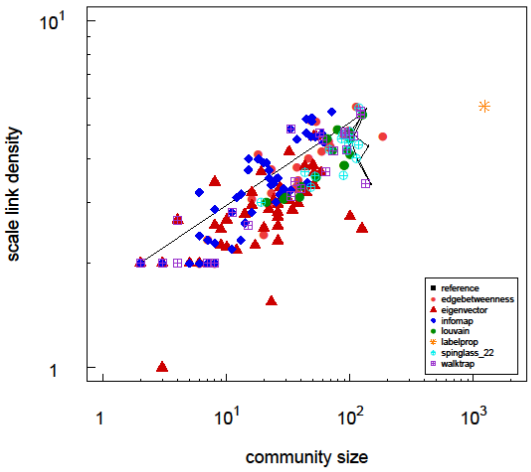
(a) Distribution de la taille des communautés



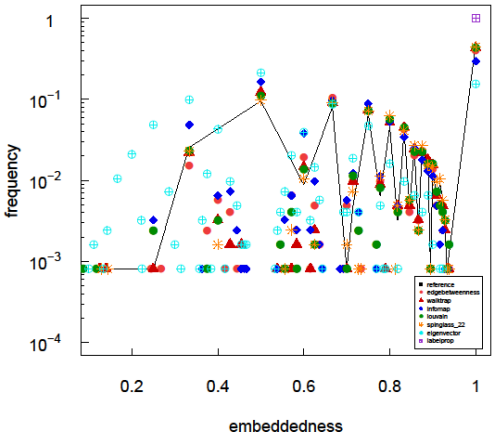
(b) Distance moyenne



(c) Dominance de hub



(d) Densité normalisée



(e) Distribution de l'enracinement

FIG. 51 - Evolution des propriétés structurelles des communautés en fonction de la taille des composantes (a, b, c, d) et distribution de l'enracinement (e) pour les sept algorithmes lorsque le nombre de communautés détecté est variable. Les valeurs de référence sont celles mesurées dans les composantes.

7.4.2 Réseau des paramètres

Détection de communautés dans le réseau global

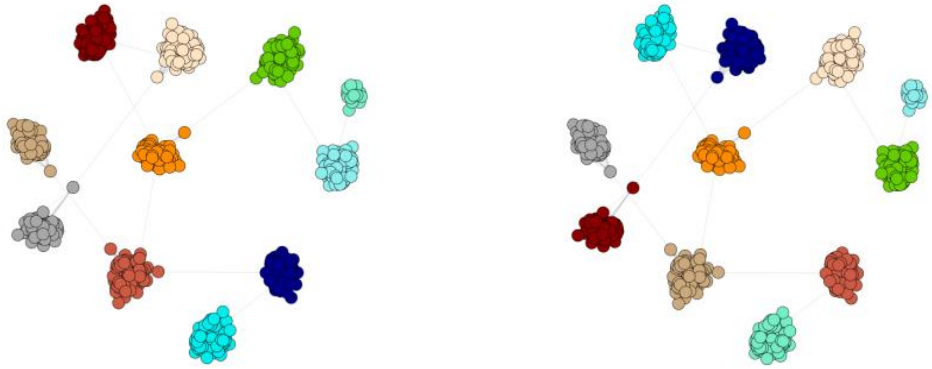
Les valeurs des mesures de performance obtenues sur le réseau de paramètres global, sans contrainte sur le nombre de communautés détectées, sont reportées dans le tableau 38. Tout d'abord, notons que les performances de tous les algorithmes sont très nettement supérieures à celles observées pour les réseaux d'opérations. Ceci tient au fait que les composantes sont bien plus denses que celles du réseau d'opérations. Cette situation plus favorable que la précédente se traduit donc par un gain de performance. Le nombre de communautés estimées est par ailleurs très proche du nombre de communautés réelles. On peut distinguer quatre groupes d'algorithmes en termes de performance. Tout d'abord un groupe formé de Infomap et LabelPropagation. En effet, ceux-ci détectent les 11 communautés sans erreur. Le second groupe est formé d'EdgeBetweenness, Louvain et Walktrap. Ces trois algorithmes détectent 10 communautés avec des performances identiques. Ce résultat vient du fait qu'ils ont tous agrégé la composante numéro 11 qui contient seulement 17 nœuds, à une composante de plus grande taille. A ce niveau, le fait d'avoir ajouté pratiquement autant de liens qu'en contenait à l'origine cette composante pour la rattacher aux autres, soulève la question de savoir si elle constitue toujours une communauté à part entière.

TAB. 38 – Détection de communautés sur le réseau d'interaction de paramètres ICEBE05. Nombre de communautés, modularité et mesures de performance des algorithmes

	Nb Communautés	MOD	IMN	PBC	IR	IRA
EDGEBETWEENNESS	10	0,847	0,986	0,976	0,994	0,971
LOUVAIN	10	0,847	0,986	0,976	0,994	0,971
SPINGLASS	13	0,845	0,982	0,972	0,994	0,967
EIGENVECTOR	1	0	0	0,1086	0,0957	0
WALKTRAP	10	0,847	0,986	0,976	0,994	0,971
INFOMAP	11	0,844	1	1	1	1
LABELPROPAGATION	11	0,844	1	1	1	1

Le troisième groupe contient Spinglass qui détecte 13 communautés dont deux de petite taille. Ce qui explique les différences insignifiantes des valeurs des mesures de performance observées avec les algorithmes précédemment cités. Eigenvector forme le dernier groupe. Il ne détecte, pour sa part, aucune communauté. Remarquons néanmoins que si le nombre de liens entre les composantes diminue, tous les algorithmes identifient clairement les 11 communautés. Cette situation intervient pour un nombre de liens égal à 30. La figure 52 représente les onze composantes détectées dans ce cas par les sept algorithmes.

En résumé, dans cette situation où les composantes sont très denses et faiblement interconnectées, à l'exception d'Eigenvector, les algorithmes sont tous très performants. De ce fait, observer les différences lorsque l'on contraint le nombre de communautés à détecter devient inutile.



(a) Référence (b) Communautés détectées

FIG. 52 – 11 Communautés de référence (a) et communautés détectées (b) par les sept algorithmes sur le réseau des paramètres ICEBE05.

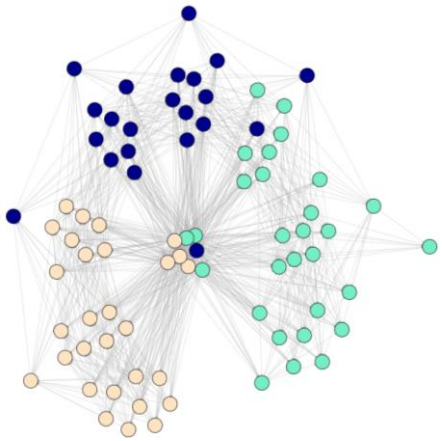
Détection de communautés dans les composantes

Les six algorithmes à même de détecter des communautés dans le réseau global sont utilisés pour étudier la structure communautaire des composantes. Les résultats obtenus permettent de scinder les algorithmes en deux groupes. Le premier groupe formé par Walktrap, Infomap et LabelPropagation ne détectent aucune communauté dans les 11 composantes. Le second groupe qui rassemble Louvain, EdgeBetweenness et Spinglass identifie une structure communautaire dans la majorité des composantes. Le tableau 39 récapitule le nombre de communautés détectées dans chacune des composantes pour ces trois derniers algorithmes. Tous s'accordent à ne pas partitionner la plus petite composante (composante 11 qui contient 17 nœuds). Pour le reste, le nombre de communautés détectées par Louvain et Spinglass est très corrélé. Il évolue entre deux et trois communautés. En ce qui concerne EdgeBetweenness, tous les cas de figure apparaissent. Dans certaines composantes, il ne détecte aucune communauté (composantes 3, 7, 8, 9, 10, 11). Il identifie un petit nombre de communautés dans les composantes 1 et 2. Il partitionne les composantes 4, 5, 6 en une multitude de composantes de tailles variables. On peut noter que la modularité calculée sur toutes ces partitions ne dépasse jamais 0,25. Elle est proche de zéro dans les situations où EdgeBetweenness détecte un grand nombre de communautés. Ce qui traduit une faible cohésion interne. Ceci nous incline à penser qu'il est plus plausible que les composantes ne possèdent pas de structure communautaire.

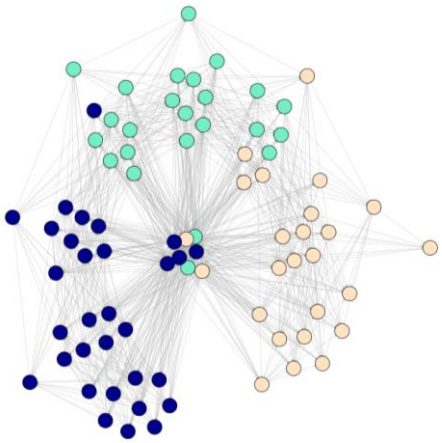
TAB. 39 - Nombre de communautés détectées dans chacune des onze composantes du réseau d'opérations par Louvain, Spinglass et EdgeBetweenness.

Composante	1	2	3	4	5	6	7	8	9	10	11
LOUVAIN	3	3	3	3	2	2	3	3	3	2	1
EDGEBETWEENNESS	4	3	1	54	33	49	1	1	1	1	1
SPINGLASS	3	3	3	3	3	3	3	3	3	2	1

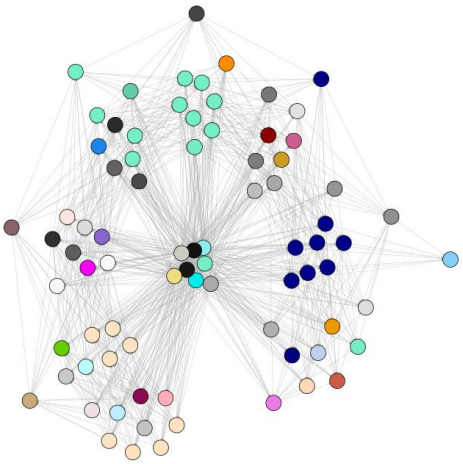
La figure 53 représente les partitions obtenues pour la composante 4 du réseau de paramètres. Dans ce cas, Louvain et Spinglass trouvent tous deux trois communautés. On peut voir que les deux partitions sont très proches. Cette figure illustre par ailleurs le fait que la taille des communautés détectées par EdgeBetweenness est très variable. On peut aussi observer que ces communautés sont très interconnectées.



SPINGLASS : 3 communautés
Modularité : 0,224



LOUVAIN : 3 communautés
Modularité : 0,217



EDGEBETWEENNESS : 54 communautés
Modularité : 0,008

FIG. 53 - Structure communautaire identifiée sur la composante 4 du réseau de paramètres par les algorithmes Spinglass, Louvain et EdgeBetweenness.

En résumé, les réseaux de paramètres sont caractérisés par des composantes très denses. Tout laisse supposer que celles-ci ne contiennent pas de communautés. C'est une situation très favorable pour les algorithmes de détection de communautés qui parviennent pratiquement tous (hormis Eigenvector), à mettre à jour la structure communautaire dans le réseau global.

7.5 Détection de communautés sur les réseaux issus de SAWSDL-TC1

Les réseaux d'interaction extraits de la collection SAWSDL-TC1 ont une structure en composantes avec une composante géante et des petites composantes. On ne s'intéresse dans ce qui suit qu'à l'étude de la structure communautaire de la composante géante. Etant donné que nous ne possédons pas d'information quant à la réalité d'une telle organisation, nous comparons les structures identifiées par les différents algorithmes. Pour ce faire, nous disposons de trois degrés de liberté : les sept algorithmes utilisés, le type de description (syntaxique, sémantique) et la granularité des réseaux (opérations, paramètres)

Nous analysons tout d'abord les différences observées sur les propriétés globales de la partition entre les algorithmes (nombre de communautés, modularité) dans les différents réseaux. Nous comparons ensuite les propriétés des communautés identifiées selon deux axes d'étude. Le premier axe consiste à évaluer les différences existantes sur des réseaux équivalents (paramètre ou opération) lorsque l'on passe d'une description syntaxique à une description sémantique des services pour chacun des algorithmes. Le second axe d'étude concerne les variations entre algorithmes toutes choses étant égales. Afin d'apprécier la cohérence entre les différents algorithmes, nous mesurons l'information mutuelle normalisée entre les partitions identifiées. En dernier lieu, nous faisons le lien entre les communautés identifiées et les domaines dans lesquels les services sont classifiés.

7.5.1 Nombre de communautés et modularité

Le tableau 40 répertorie le nombre de communautés ainsi que la valeur de la modularité associée aux structures communautaires découvertes par chacun des algorithmes dans les quatre réseaux étudiés.

Considérons tout d'abord la comparaison entre les réseaux de paramètres et les réseaux d'opérations utilisant le même type de description. Les tendances statistiques qui se dégagent tous algorithmes confondus, sont décrites par les mesures des valeurs moyennes et des écarts types. Globalement, les réseaux de paramètres contiennent moins de communautés que les réseaux des opérations. De plus, la variabilité du nombre de communautés détectées est bien plus grande dans ces derniers. Nous remarquons également que la modularité est plus élevée dans les réseaux de paramètres que dans les réseaux d'opérations. Autrement dit, les communautés détectées sont plus cohésives. Notons que ce sont bien les mêmes tendances que l'on a pu observer sur la collection ICEBE05.

Nous nous attachons maintenant à la comparaison sur l'axe de la description (syntaxique, sémantique). Pour les réseaux de paramètres, les différences sont statistiquement insignifiantes aussi bien pour la modularité moyenne que pour le nombre moyen de communautés détectées Dans les réseaux d'opérations le nombre moyen de communautés détectées est très comparable. La modularité est systématiquement supérieure dans le réseau des opérations sémantiques. Si l'on s'attache à comparer les différents comportements des algorithmes à partir de ces informations, on peut remarquer que Spinglass est, dans tous les cas, l'algorithme qui présente la plus forte valeur de la modularité. Dans les réseaux de

paramètres, EdgeBetweenness, Louvain et Spinglass semblent très proches du point de vue de la modularité et du nombre de communautés détectées. Walktrap, Eigenvector et Infomap sont un cran en dessous pour la modularité avec un nombre de communautés presque doublé. En ce qui concerne les réseaux d'opérations, Louvain, Spinglass et Infomap présentent les plus fortes valeurs de modularité. Cette valeur est atteinte avec 20 communautés pour Infomap, alors que pour les deux autres algorithmes, le nombre de partitions obtenues est de l'ordre de la dizaine.

TAB. 40 – Détection de communautés sur les réseaux de paramètres et d'opérations de la collection SAWSDL-TC1. Nombre de communautés et modularité.

	RESEAUX DE PARAMETRES				RESEAUX D'OPERATIONS			
	Syntaxique égal		Sémantique exact		Syntaxique égal		Sémantique exact	
	Nb	Mod	Nb	Mod	Nb	Mod	Nb	Mod
EDGEBETWEENNESS	9	0,621	14	0,624	48	0,477	43	0,506
LOUVAIN	10	0,618	10	0,619	8	0,492	9	0,53
SPINGLASS	9	0,637	12	0,63	12	0,508	12	0,53
EIGENVECTOR	15	0,6	12	0,596	6	0,434	5	0,479
WALKTRAP	17	0,609	16	0,618	23	0,479	20	0,478
INFOMAP	17	0,608	18	0,61	21	0,506	20	0,529
LABELPROPAGATION	10	0,593	13	0,581	9	0,361	13	0,506
Moyenne	12,4	0,612	13,5	0,611	18,14	0,465	17,42	0,508
Ecart type	3,7	0,014	2,7	0,017	14,69	0,052	12,52	0,022

7.5.2 Propriétés des communautés

Comparaison selon l'axe de la description des services

Nous voulons pouvoir comparer les propriétés des partitions obtenues par un algorithme sur les versions syntaxique et sémantique des réseaux. Pour cela nous avons calculé pour chacune des propriétés, la corrélation entre la série de données obtenue dans le cas syntaxique et celle obtenue dans le cas sémantique. Dans tous les cas, seules les huit communautés de plus grande taille sont considérées. En effet, pour un même algorithme, le nombre de communautés détectées diffère entre le cas syntaxique et le cas sémantique. De plus, les valeurs des propriétés peuvent être très variables dans les communautés de petite taille. Il est donc préférable de minimiser leur influence. Le tableau 41 contient les valeurs du coefficient de corrélation pour chacun des algorithmes utilisés dans les réseaux de paramètres et d'opérations. Les propriétés étudiées sont la taille des communautés, la distance moyenne dans les communautés, la densité normalisée et la dominance de hub. Globalement, on observe de grandes valeurs de la corrélation pour toutes les propriétés dans les deux types de réseaux. Ceci permet de penser que les communautés détectées dans le cas syntaxique sont très proches structurellement de celles issues des réseaux sémantiques. Aller plus loin dans l'interprétation de ces résultats est très malaisé, car les différences observées sont difficilement interprétables sur une si petite série de données.

TAB. 41 - Corrélation entre les distributions des propriétés des communautés des réseaux syntaxiques et sémantiques de paramètres et d'opérations de la collection SAWSDL-TC1.

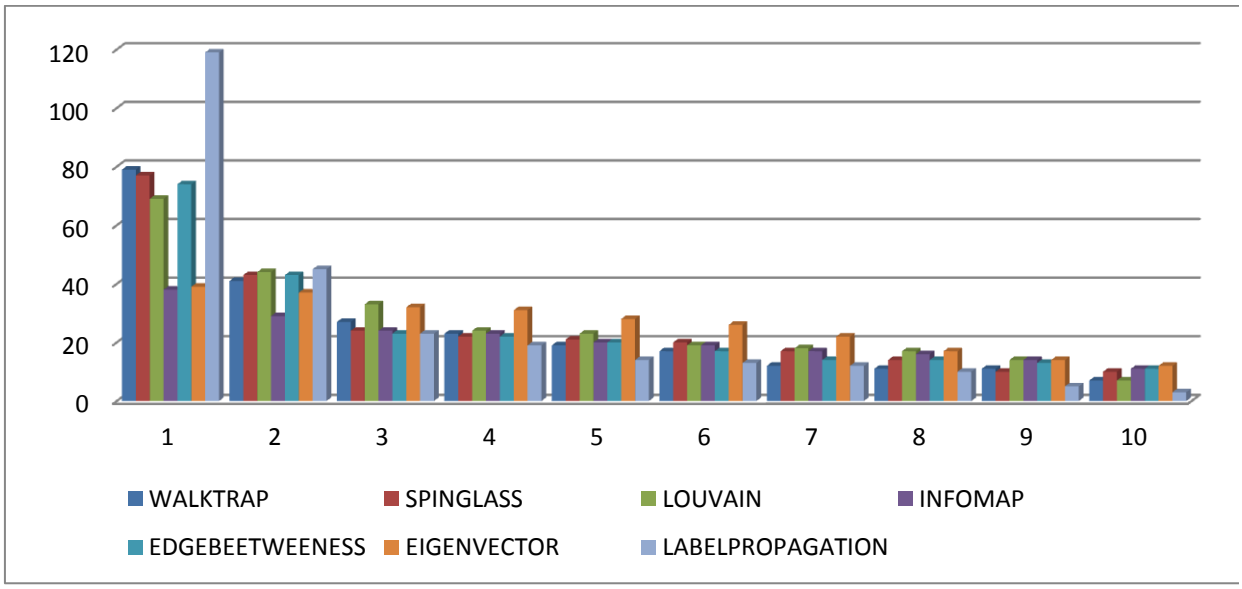
	RESEAUX DE PARAMETRES				RESEAUX D'OPERATIONS			
	Taille	Distance	Densité	Dom. Hub	Taille	Distance	Densité	Dom. Hub
WALKTRAP	0,991	0,974	0,952	0,917	0,993	0,895	0,983	0,973
SPINGLASS	0,985	0,943	0,965	0,881	0,954	0,910	0,987	0,909
LOUVAIN	0,877	0,924	0,931	0,896	0,951	0,852	0,984	0,963
INFOMAP	0,955	0,989	0,985	0,871	0,985	0,902	0,989	0,942
EDGEBETWEENNESS	0,993	0,947	0,966	0,736	0,996	0,835	0,994	0,947
EIGENVECTOR	0,932	0,968	0,911	0,971	0,903	0,813	0,883	0,922
LABELPROPAGATION	0,996	0,974	0,945	0,798	0,931	0,927	0,973	0,909

Comparaison entre les algorithmes

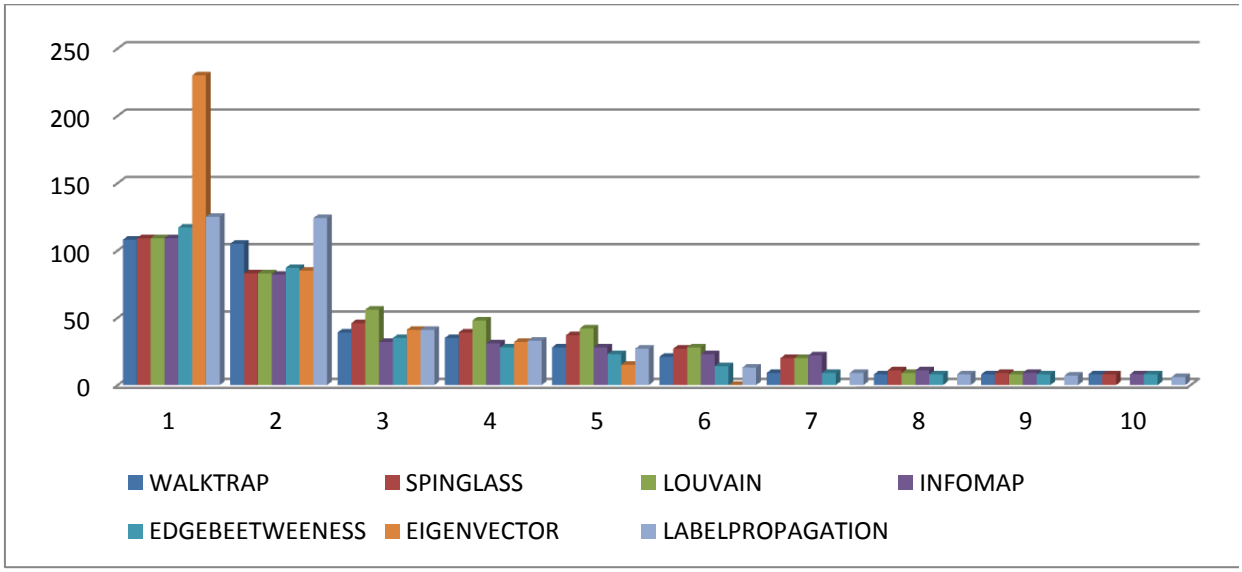
Afin d'illustrer les différences entre les partitions, nous donnons la taille des dix plus « grosses » communautés qui ont été découvertes par chacun des algorithmes dans les réseaux sémantiques de paramètres et d'opérations. En ce qui concerne les autres propriétés, on observe globalement des valeurs assez comparables pour la distance moyenne, et ce pour l'ensemble des algorithmes. Pour la dominance de hub et la densité à l'échelle, on remarque qu'elles sont caractérisées par une plus grande variabilité, et ce plus particulièrement, sur les communautés de petite taille. Sur la figure 54, on observe que les algorithmes s'accordent peu sur la taille de la plus « grosse » communauté dans le cas du réseau des paramètres. Pour les autres, les avis sont assez convergents. Notons qu'Infomap et Eigenvector aboutissent à une répartition plus uniforme de la taille des communautés. LabelPropagation a formé la plus « grosse » communauté. Dans le réseau d'opérations, les « avis » sont plus convergents, hormis pour Eigenvector qui se distingue par la taille de la plus « grosse » communauté.

7.5.3 Comparaison des partitions

Le tableau 42 donne les résultats de la comparaison des partitions générées par les algorithmes de détection de communautés dans les réseaux sémantiques. La mesure utilisée est l'information mutuelle normalisée mesurée entre les partitions identifiées par les sept algorithmes pris deux à deux. Ces résultats se présentent sous la forme d'une matrice symétrique qui évalue ainsi le degré de cohérence entre les différentes partitions. Il apparaît que même si le nombre et la taille des partitions sont globalement assez différents, les algorithmes s'accordent sur le contenu de ces partitions. Eigenvector est celui qui se démarque le plus des autres, indépendamment du type de réseau. LabelPropagation a aussi tendance à se distinguer des autres dans le réseau des opérations. Les algorithmes les plus consensuels sont Spinglass et Walktrap dans le réseau des paramètres. En effet, ils possèdent la valeur moyenne de l'information mutuelle normalisée la plus élevée (0,85). Dans le réseau des opérations, c'est Spinglass et Infomap qui occupent ce rôle. On observe le même type de comportement dans les réseaux syntaxiques à quelques nuances près. Globalement, le consensus est en moyenne légèrement moins élevé (de l'ordre de 5%) dans les réseaux d'opérations que dans les réseaux de paramètres. L'ordonnancement des algorithmes en termes de consensus est identique.



Réseau des paramètres sémantique



Réseau des opérations sémantique

FIG. 54 - Taille des dix communautés de plus grande taille détectées par les sept algorithmes dans les réseaux sémantiques.

TAB. 42 - Information Mutuelle Normalisée entre deux partitionnements dans les réseaux sémantiques des paramètres et des opérations. Chaque case donne l'IMN entre deux partitionnements. Le nom des algorithmes est abrégé dans les colonnes. La *iéme* colonne correspond à l'algorithme présenté sur la *iéme* ligne.

	RESEAUX DE PARAMETRES							RESEAUX D'OPERATIONS						
	SPI	WAL	INF	LOU	LAB	EIG	EDG	SPI	WAL	INF	LOU	LAB	EIG	EDG
SPINGLASS	1,00	0,85	0,86	0,87	0,82	0,73	0,88	1,00	0,84	0,91	0,93	0,80	0,58	0,83
WALKTRAP	0,85	1,00	0,85	0,83	0,83	0,75	0,85	0,84	1,00	0,89	0,80	0,79	0,47	0,82
INFOMAP	0,86	0,85	1,00	0,80	0,74	0,78	0,88	0,91	0,89	1,00	0,89	0,82	0,60	0,88
LOUVAIN	0,87	0,83	0,80	1,00	0,77	0,69	0,81	0,93	0,80	0,89	1,00	0,78	0,58	0,78
LABELPROPAGATION	0,82	0,83	0,74	0,77	1,00	0,70	0,80	0,80	0,79	0,82	0,78	1,00	0,52	0,80
EIGENVECTOR	0,73	0,75	0,78	0,69	0,70	1,00	0,74	0,58	0,47	0,60	0,58	0,52	1,00	0,59
EDGEBETWEENNESS	0,88	0,85	0,88	0,81	0,80	0,74	1,00	0,83	0,82	0,88	0,78	0,80	0,59	1,00

7.5.4 Lien entre communautés et domaines

Pour compléter cette étude, nous avons mené une analyse subjective des communautés identifiées par les différents algorithmes dans les réseaux. En ce qui concerne les réseaux des opérations, on observe que, globalement, les communautés et les domaines ne se recouvrent pas. Ainsi, si nous concentrons notre attention sur les trois plus « grosses » communautés, nous remarquons que dans toutes les partitions, elles rassemblent une grande partie des opérations du domaine *economy* avec des opérations du domaine *travel* et *education*. Pour les communautés de taille moyenne, la mixité est plus homogène. Dans les réseaux de paramètres, la structure communautaire est quelque peu différente. En effet, dans ce cas, les communautés sont plus centrées sur les domaines. Ceci s'explique par le fait que les réseaux s'organisent autour d'un vocabulaire commun (les paramètres) spécifique à chaque domaine. Quoi qu'il en soit, la notion de communauté est bien plus riche que la notion de domaine. En effet, une communauté regroupe les services à même d'entrer dans une composition, alors que la classification par domaine n'induit pas forcément de relation d'interaction.

7.6 Conclusion

Nous avons mené une analyse de la structure communautaire des réseaux d'interaction à partir de deux collections de descriptions de services. Nous avons tout d'abord utilisé une collection artificielle, ICEBE05, afin de qualifier des algorithmes représentatifs des diverses solutions apportées au problème de la détection de communautés dans les réseaux complexes. Dans un second temps, nous avons utilisé ces algorithmes afin de valider l'hypothèse d'une structure communautaire dans les réseaux issus d'une collection construite à partir de services réels, en l'occurrence SAWSDL-TC1.

Le principal enseignement qui se dégage des résultats obtenus sur la collection ICEBE05 est que la densité des communautés est l'élément prépondérant en ce qui concerne les performances des algorithmes. Ainsi, les réseaux de paramètres construits à partir de cette collection sont obtenus en formant une composante connexe à partir de composantes très

denses, en ajoutant cinquante liens entre les onze composantes. Dans cette situation, pratiquement tous les algorithmes permettent de retrouver les composantes originelles (à l'exception de la plus petite qui ne se distingue plus). De plus, si on leur demande de partitionner les composantes elles-mêmes, la réponse qui se dégage majoritairement est que ces composantes ne présentent pas de structure communautaire. En effet, seul trois algorithmes (Louvain, Spinglass, et EdgeBetweenness) détectent un petit nombre de communautés fort peu cohésives. Dans les réseaux d'opérations beaucoup moins denses, la situation est bien moins favorable, ce qui se traduit par un niveau de performance plus faible. Ceci permet de mettre en évidence des comportements très différents. Trois algorithmes se dégagent dans l'ordre suivant : Walktrap, Spinglass et Louvain, pour ce qui est des mesures de performances. Si l'on examine les propriétés structurelles des communautés, cet ordre se modifie. Spinglass identifie des communautés dont les propriétés structurelles sont plus proches de celles des communautés de référence. Il est suivi par Louvain puis par Walktrap dans ce classement. La recherche de communautés dans les composantes met en valeur une grande hétérogénéité. Néanmoins, les valeurs de modularité relevées pour les algorithmes qui détectent une structure communautaire rendent cette hypothèse plausible.

L'analyse de la collection SAWSDL-TC1 fait apparaître que le niveau de difficulté est plutôt moyen pour les algorithmes. Tous s'accordent à identifier des communautés dans les réseaux construits à partir de cette collection. Le réseau de paramètres là aussi soulève moins de difficultés que le réseau d'opérations qui possède des communautés moins cohésives. On observe peu de différence entre les réseaux syntaxique et sémantique. Bien que différentes, les partitions sont néanmoins très proches. Cette proximité est attestée tant par les propriétés topologiques des communautés que par la mesure de similitude des partitions. Notons que globalement, les communautés dans les réseaux d'opérations ont tendance à regrouper des opérations provenant de domaines différents, alors que les réseaux de paramètres sont plus centrés sur un domaine spécifique.

L'apport principal de cette étude est de valider un nouveau mode de classification de services Web reposant sur leur capacité à être composés. En effet, ces résultats montrent que la détection de communautés dans les réseaux d'interaction permet d'envisager une structuration de l'espace des services Web alternative à la notion de domaine d'intérêt.

CONCLUSION

Les services Web s'inscrivent en tant que briques de base dans le paradigme SOC (Service-Oriented Computing) pour le développement de systèmes d'information distribués. Cette nouvelle façon de concevoir les applications est une réponse à un changement fondamental dans le fonctionnement des entreprises et des organismes. Les applications propriétaires intégrées sont remplacées par des applications distribuées où chaque participant fournit des services spécialisés. Ces services ne cessent de gagner en popularité, ils sont de plus en plus nombreux et sont caractérisés par un aspect hautement dynamique. Dans un tel contexte, il convient d'envisager la structuration de l'espace que forment ces services.
La contribution principale de cette thèse réside dans l'utilisation du paradigme des réseaux complexes pour représenter l'espace de services Web. Cette approche permet de considérer les services comme un ensemble dynamique de grande taille composé d'entités autonomes qui peuvent entretenir différents types de relations. En inscrivant le système d'information que constituent les services Web dans le cadre des grands graphes de terrain, nous disposons d'une panoplie d'outils et de méthodes qui permettent de mieux comprendre et d'organiser efficacement cet ensemble. Ceci permet par exemple de fournir une vision à un moment donné de l'état du système et aussi d'envisager des modes de classification directement en liaison avec la structure du réseau sous-jacente.

Dans ce travail, les services sont considérés comme des entités fonctionnelles qui regroupent un ensemble d'opérations. Une opération est, quant à elle, caractérisée par un ensemble de paramètres d'entrée/sortie. Nous avons proposé au chapitre deux, deux modèles de réseaux de services Web. Le premier, un modèle d'interaction, représente les relations de composition entre les services. Nous avons modélisé ce phénomène de plusieurs façons. Pour cela, un premier degré de liberté réside dans le choix des nœuds qui peuvent être les services eux-mêmes, leurs opérations ou leurs paramètres. Un deuxième degré de liberté consiste à considérer la quantité d'information nécessaire pour relier deux opérations ou deux services. Un service peut ainsi fournir tout ou partie des paramètres d'une opération pour en invoquer un autre. Le troisième degré est relatif au choix de la fonction de mise en correspondance pour déterminer la similitude entre les paramètres. Cette fonction dépend de la description des services qui peut être syntaxique ou sémantique. Le modèle défini est inspiré de travaux préexistants sur l'utilisation des réseaux pour représenter un ensemble de services Web en interaction, ainsi que de travaux traitant de la mise en correspondance des services Web. Cette littérature connexe fait l'objet du premier chapitre.
Le deuxième modèle représente des relations de similitude qui peuvent exister entre services d'un point de vue fonctionnel, c'est-à-dire, dans notre cas, en ne considérant que les opérations et leurs paramètres. Deux opérations sont dites similaires si elles partagent des caractéristiques communes en termes de paramètres d'entrée/sortie. Nous avons défini un ensemble de fonctions qui traduisent divers degrés de similitude entre les opérations des services. Ces fonctions utilisent des métriques et des opérateurs de mise en correspondance présentés dans le cadre des travaux connexes au chapitre deux.

Une fois les modèles définis, nous les avons appliqués à une collection de descriptions de services Web afin d'obtenir un ensemble de réseaux. Nous avons pour cela mis au point un

outil pour l'extraction des réseaux à partir de collections de descriptions syntaxiques et sémantiques. L'étape suivante, qui fait l'objet des chapitres quatre et cinq, a consisté à analyser les propriétés topologiques de ces réseaux. La définition de ces propriétés ainsi que des moyens de les appréhender sont abordés au chapitre trois.

L'analyse topologique des réseaux d'interaction révèle que quelles que soient les entités sur les nœuds (paramètres ou opérations), les réseaux sont structurés autour d'une composante géante, traduisant ainsi la capacité des services à interagir. Cette composante possède des propriétés analogues à celles observées dans les grands graphes de terrain. Elle respecte ainsi la propriété petit monde. La propriété sans échelle est révélée sans ambiguïté pour les réseaux de paramètres et dans une moindre mesure pour les réseaux d'opérations. Cette étude montre ainsi le bien-fondé de l'approche réseaux complexes pour appréhender l'espace des services Web. Elle permet de positionner les réseaux d'interaction de services Web dans l'ensemble des autres grands graphes de terrain. Nous avons également montré que l'introduction de la sémantique dans les descriptions modifie la topologie des réseaux et produit une structuration plus efficace des services.

En ce qui concerne l'analyse topologique des réseaux de similitude, nous avons montré qu'ils sont constitués d'un ensemble de composantes qui regroupent des opérations similaires et que leur structure dépend de la fonction de similitude utilisée. Nous mettons ainsi en évidence deux types de motifs élémentaires : la clique et l'étoile. Nous avons par ailleurs montré l'existence de deux niveaux structurels au sein des composantes. Le premier niveau correspond au motif élémentaire d'organisation des nœuds dans la composante. Le deuxième niveau correspond à une juxtaposition de plusieurs exemplaires du motif élémentaire. Ce modèle d'organisation permet d'affiner la notion de domaine en permettant de distinguer dans une même composante, des services à travers les notions de voisinage.
Les travaux concernant les modèles de réseaux, leur analyse topologique ainsi que l'extracteur ont fait l'objet des publications suivantes [139] [140] [141] [102] [142] [143].

L'existence de communautés est une caractéristique non triviale commune à de nombreux grands graphes de terrain. Nous avons mené une étude de la structure communautaire sur les réseaux d'interaction de services Web. Les outils utilisés pour cette étude sont présentés au chapitre six. Dans le chapitre sept, une première série d'expérimentations conduites sur une collection de descriptions artificielles dont la structure communautaire était connue, nous a permis de qualifier les algorithmes de détection de communautés utilisés. A l'issue d'une deuxième série d'expérimentations pratiquées sur des descriptions de services réels, nous avons pu conclure que les réseaux d'interaction de services Web exhibent eux aussi une organisation en communautés. Cette structuration permet de classifier des services Web en fonction de leur aptitude à intervenir au sein d'une même composition. Ce travail permet ainsi de mettre en exergue l'apport de l'approche réseaux complexes dans les modèles d'organisation de services Web.

Nos travaux de recherche ont porté sur la représentation et la structuration de l'espace des services Web. Ils comprennent trois axes principaux qui sont la conception de modèles de réseaux, une analyse topologique des réseaux de services Web, un mode de classification basé

sur la similitude des services et un mode de classification basé sur leur interaction. Il reste cependant de nombreuses pistes à explorer au-delà de ce travail.

Tout d'abord, on peut envisager un certain nombre d'extensions de différentes natures.

Il est ainsi possible de poursuivre l'analyse topologique des réseaux. Un certain nombre de mesures existent qu'il convient d'examiner. A titre d'exemple, on peut citer les mesures de centralité. Ainsi, dans l'analyse topologique que nous avons conduite, nous avons pu remarquer la présence de nœuds spécifiques dans les réseaux qui sont les hubs et les autorités. Ces nœuds jouent un rôle prépondérant dans les phénomènes dynamiques intervenant dans les réseaux et leur défaillance peut gravement impacter ces phénomènes. D'autres mesures existent afin de quantifier l'importance d'un nœud ou d'un lien dans un réseau. La caractérisation de ces nœuds peut ainsi permettre aux fournisseurs de développer ce genre de service qui occupe un rôle central dans les processus de composition. L'identification de ces nœuds peut aussi permettre de développer des heuristiques pour guider les algorithmes de découverte de composition.
Au niveau de la découverte de communautés dans les réseaux d'interaction, nous nous sommes intéressés aux algorithmes qui effectuent une partition du réseau. Dans la réalité, les frontières entre communautés sont plus lâches. Une entité peut ainsi appartenir à plusieurs communautés. Il convient d'utiliser des algorithmes qui permettent de considérer cette situation.

Nous avons constaté que les caractéristiques de la principale collection sur laquelle sont conduites nos expérimentations, à savoir SAWSDL-TC1, ont pu parfois être à l'origine de comportements biaisés ou ne pas satisfaire pleinement nos attentes. Cependant, il s'agissait de la seule collection publiquement disponible à notre connaissance, à même de répondre au mieux à nos exigences. Il s'avère donc utile de pouvoir disposer d'une collection de services réels et non ré-échantillonnés, comportant à la fois la description syntaxique et sémantique des paramètres. Cette collection doit posséder un nombre suffisant de services aux fonctionnalités similaires et être utilisable dans le cadre de la composition. Dans la mesure du possible elle doit contenir des services classifiés par domaine. Nous proposons pour cela de procéder à l'annotation automatique d'une collection existante de descriptions syntaxiques réelles. C'est un travail qui est en cours et qui a d'ores et déjà fait l'objet d'une publication [144].

Nous envisageons par ailleurs d'utiliser les réseaux de services Web comme support pour la synthèse de compositions. Nous avons mené des travaux préliminaires en ce sens. On peut ainsi définir une architecture à deux couches basée sur les réseaux de similitude associés aux réseaux d'interaction. La couche inférieure est la couche instance. Elle représente l'ensemble des opérations concrètes de services Web publiés organisés sous la forme d'un réseau de similitude. La couche supérieure est formée d'un réseau d'interaction de méta opérations. Chaque méta opération représente une composante du réseau de similitude. Les relations de composition des opérations concrètes sont utilisées pour construire un réseau d'interaction des méta-opérations. Cette couche d'abstraction permet dans un premier temps une recherche de compositions dans un espace restreint de méta-opérations. Les compositions ainsi découvertes

sont dans un second temps instanciées par les opérations concrètes du réseau de similitude avec possibilité de substitution.

En dehors de ces extensions naturelles, l'utilisation des réseaux complexes permet d'envisager un vaste champ d'investigation. On peut ainsi penser à l'étude de la dynamique des réseaux de services Web et à celle des processus de composition.

BIBLIOGRAPHIE

[1] W3C, *Web Services Architecture*. 2003.

[2] OASIS, *Reference Model for Service-Oriented Architecture 1.0*. 2006.

[3] A. Gustavo, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architecture and Applications*. Springer Verlag, 2004.

[4] J. Yang and M. P. Papazoglou, "Service Components for Managing the Life-Cycle of Service Compositions," *Information Systems*, vol. 29, no. 2, 2004.

[5] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana, *Web Services Description Language (WSDL) Version 2.0*, no. W3C Recommendation 26 June 2007.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *In Scientific American, May 2001*, 2001.

[7] D. Martin et al., *OWL-S: Semantic Markup for Web Services*, no. W3C Member Submission 22 November 2004.

[8] R. Akkiraju et al., *Web Service Semantics - WSDL-S*, no. 7 November 2005.

[9] J. Farrell and H. Lausen, *Semantic Annotations for WSDL and XML Schema*, no. W3C Recommendation 28 August 2007.

[10] M. Dumas, L. García-Bañuelos, and R. Dijkman, "Similarity Search of Business Process Models," *IEEE Data Eng. Bull*, 2009.

[11] Ginsberg A., "An unified approach to Automatic Indexing and Information Retrieval," *IEEE Expert*, vol. 8, no. 5, pp. 46-56, 1993.

[12] Voorhees E., "Using WordNet for text retrieval," in *WordNet: An Electronic Lexical Database*, Cambridge: C. Fellbaum MIT Press, 1998.

[13] Bishr Y., "Semantic aspects of interoperable," GIS Wageningen Agricultural University and ITC, 1997.

[14] A. Bouguettaya, B. Benatallah, and A. Elmagarmid, "Interconnecting heterogeneous information systems," in *Advances in Database Systems*, A. Elmagarmid, 1998.

[15] M. Levit, E. Nth, and A. Gorin, "Using EM-trained stringedit distances for approximate matching of acoustic morphemes," in *Proc. of International Conference on Spoken Language Processing*, 2002, pp. 1157-1160.

[16] E. Hovy, "Combining and standardizing large-scale practical ontologies for machine translation and other uses," in *Proc. of the First Int. Conf. on Language Resources and Evaluation*, 1998, pp. 535-542.

[17] R. Rada, H. Mili, E. Bicknell, and M. Blettner, “Development and application of a metric on semantic nets ,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, 1989.

[18] M. Papazoglu and W. J. van den Heuvel., “Service-Oriented Computing : State-of-the-Art and Open Research Issues,” *ICSOC*. 2003.

[19] Y. Ganjisaffar, H. Abolhassani, M. Neshati, and M. Jamali, “A Similarity Measure for OWL-S Annotated Web Services,” in *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI’06)*, 2006, pp. 621-624.

[20] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg, “Adaptive name matching in information integration,” *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16-23, Sep. 2003.

[21] W. Cohen, P. Ravikumar, and S. Fienberg, “A comparison of string metrics for matching names and records,” in *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*, 2003, pp. 73-78.

[22] M. Sanderson and B. Croft, “Deriving concept hierarchies from text,” in *SIGIR Conference’99*, 1999, pp. 206-213.

[23] N. Seco, T. Veale, and J. Hayes, “An intrinsic information content metric for semantic similarity in WordNet,” in *Proc. of ECAI-04*, 2004, pp. 1089–1090.

[24] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, “Semantic Matching of Web Services Capabilities,” in *International Semantic Web Conference*, 2002, pp. 333-347.

[25] Oh S.-C., “Effective Web Services Composition in diverse and large-scale services networks,” Pennsylvania State University, 2006.

[26] J. Ma, Y. Zhang, and J. He, “Web Services Discovery Based on Latent Semantic Approach,” in *2008 IEEE International Conference on Web Services*, 2008, pp. 740-747.

[27] C. Wu, V. Potdar, and E. Chang, “Latent semantic analysis - The dynamics of semantics web services discovery,” in *Advances in Web Semantics I*, vol. 4891, T. S. Dillon, E. Chang, R. Meersman, and K. Sycara, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 346-373.

[28] U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel, “Automatic Location of Web Services,” *ESWC*. 2005.

[29] U. Küster and B. König-Ries, “Evaluating semantic web service matchmaking effec tiveness based on graded relevance ,” in *Proc. of the 2nd International Workshop SMR 2 on Service Matchmaking and Resource Retrieval in the Semantic Web at ISWC* , 2008.

[30] C. Preist, “A Conceptual Architecture for Semantic Web Services,” *In Proceedings of the 3rd International Semantic Web Conference*. pp. 395-409, 2004.

[31] Lécué F., “Web Service composition: Semantic Links based approach,” Ecole des Mines de Saint-Etienne, Saint-Etienne, France, 2008.

[32] I. Katakis, G. Meditskos, G. Tsoumakas, N. Bassiliades, and I. Vlahavas, “On the Combination of Textual and Semantic Descriptions for Automated Semantic Web Service Classification,” vol. 296, Boston, MA: Springer US, 2009, pp. 95-104.

[33] B. Medjahed and A. Bouguettaya, “A Dynamic Foundational Architecture for Semantic Web Services,” *Distributed and Parallel Databases*, vol. 17, no. 2, pp. 179-206, Mar. 2005.

[34] I. Arpinar, B. Aleman-Meza, and R. Zhang, “Ontology-driven web services composition platform,” *Inf. Syst. E-Business Management*, vol. 3, 2005.

[35] B. Benatallah, M. Dumas, and Q. Z. Sheng, “Facilitating the Rapid Development and Scalable Orchestration of Composite Web Services,” *Distributed and Parallel Databases*, vol. 17, no. 1, pp. 5-37, Jan. 2005.

[36] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z. Maamar, “Towards an Approach for Web Services Substitution,” in *10th International Database Engineering and Applications Symposium (IDEAS’06)*, 2006, pp. 166-173.

[37] M. Bruno, G. Canfora, M. Di Penta, and R. Scognamiglio, “An Approach to support Web Service Classification and Annotation,” in *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2005, pp. 138-143.

[38] N. Oldham, C. Thomas, A. Sheth, and K. Verma, “METEOR-S Web Service Annotation Framework with Machine Learning Classification,” 2004.

[39] InfoEther and B. B. N. Technologies, “SemWebCentral,” 2004. [Online]. Available: http://wwwprojects.semwebcentral.org/.

[40] A. Konduri and C. C. Chan, “Clustering of Web Services Based on WordNet Semantic Similarity.” University of Akron, USA, 2008.

[41] J. Wu and Z. Wu, “Similarity-based Web Service Matchmaking,” *IEEE International Conference on Semantic Computing*. pp. 287-294 , 2005.

[42] R. Nayak and B. Lee, “Web Service Discovery with additional Semantics and Clustering,” in *IEEE/WIC/ACM International Conference on Web Intelligence (WI’07)*, 2007, pp. 555-558.

[43] Z. Azmeh, M. Huchard, C. Tibermacine, C. Urtado, and S. Vauttier, “WSPAB: A Tool for Automatic Classification & Selection of Web Services Using Formal Concept Analysis,” in *2008 Sixth European Conference on Web Services*, 2008, pp. 31-40.

[44] S. V. Hashemian and F. Mavaddat, “A Graph-Based Approach to Web Services Composition,” in *The 2005 Symposium on Applications and the Internet*, 2005, pp. 183-189.

[45] H. Nacer Talantikite, D. Aissani, and N. Boudjlida, “Semantic annotations for web services discovery and composition,” *Computer Standards Interfaces*, vol. 31, no. 6, pp. 1108-1117, 2009.

[46] J. Liu and L. Chao, “Design and Implementation of an Extended UDDI Registration Center for Web Service Graph,” in *IEEE International Conference on Web Services (ICWS 2007)*, 2007.

[47] J. Kwon, K. Park, D. Lee, and S. Lee, “PSR : Pre-computing Solutions in RDBMS for Fast Web Services Composition Search,” in *International Conference on Web Services (ICWS)*, 2007.

[48] L. Dekar and H. Kheddouci, “A Graph b-Coloring Based Method for Composition-Oriented Web Services Classification,” vol. 4994, A. An, S. Matwin, Z. W. Raś, and D. Ślęzak, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 599-604.

[49] J. Gekas and M. Fasli, “Employing Graph Network Analysis for Web Service Composition,” in *Agent Technologies and Web Engineering*, D. C. (eds) Alkhatib G. I. and Rine, Ed. IGI Global, 2008.

[50] M. E. J. Newman, “Power laws, Pareto distributions and Zipf’s law,” *Contemporary Physics*, vol. 46, no. 5, pp. 323-351, Sep. 2005.

[51] S. C. Oh and D. Lee, “WSBen: A Web Services Discovery and Composition Benchmark Toolkit,” *International Journal of Web Services Research (JWSR)*, vol. 6, pp. 1-19, 2009.

[52] S.-C. Oh et al., “WSPR*: Web-Service Planner Augmented with A* Algorithm,” in *2009 IEEE Conference on Commerce and Enterprise Computing*, 2009, pp. 515-518.

[53] P. Shvaiko, “A Survey of Schema-Based Matching Approaches,” *J. Data Semantics IV*, pp. 146-171, 2005.

[54] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47-97, Jan. 2002.

[55] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks : From Biological Nets to the Internet and WWW*. Oxford: Oxford University Press, 2003.

[56] M. E. J. Newman, “The Structure and Function of Complex Networks,” *SIAM Review*, vol. 45, no. 2, p. 167, 2003.

[57] S. H Strogatz, “Exploring complex networks.,” *Nature*, vol. 410, no. 6825, pp. 268-276, 2001.

[58] Watts Duncan J., *Six Degrees: The Science of a Connected Age*, 1st ed. W. W. Norton & Company, 2003.

[59] U. Brandes and T. Erlebach, "Network Analysis: Methodological Foundations," Springer, 2005.

[60] F. Zaidi, "Analyse, Structure et Organisation des Réseaux Complexes," 2010.

[61] M. E. J. Newman, "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.," *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 64, no. 1 Pt 2, 2001.

[62] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167-242, Jan. 2007.

[63] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175-308, Feb. 2006.

[64] M. N. K. Boulos, "The use of interactive graphical maps for browsing medical/health Internet information resources," *International Journal of Health Geographics*, vol. 2, no. 1, 2003.

[65] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, pp. 61-67, 1967.

[66] R. Albert, H. Jeong, and A.-L. Barabasi, "The diameter of the world wide web," *Nature*, vol. 401, no. September, pp. 130-131, 1999.

[67] L. A. Adamic, "The Small World Web," in *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, 1999, pp. 443-452.

[68] P. Sen, S. Dasgupta, A. Chatterjee, P. Sreeram, G. Mukherjee, and S. Manna, "Small-world properties of the Indian railway network," *Physical Review E*, vol. 67, no. 3, Mar. 2003.

[69] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks.," *Nature*, vol. 393, no. 6684, pp. 440-2, Jun. 1998.

[70] C. Moore and M. Newman, "Epidemics and percolation in small-world networks," *Physical Review E*, vol. 61, no. 5, pp. 5678-5682, May. 2000.

[71] E. Yoneki, P. Hui, and J. Crowcroft, "Wireless Epidemic Spread in Dynamic Human Networks," in *Bio-Inspired Computing and Communication*, vol. 5151, P. Liò, E. Yoneki, J. Crowcroft, and D. C. Verma, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 116-132.

[72] S. A. Delre, W. Jager, and M. A. Janssen, "Diffusion dynamics in small-world networks with heterogeneous consumers," *Computational and Mathematical Organization Theory*, vol. 13, no. 2, pp. 185-202, Sep. 2006.

[73] H. Zhang, A. Goel, and R. Govindan, "Using the small-world model to improve Freenet performance," *Computer Networks*, vol. 46, no. 4, pp. 555-574, Nov. 2004.

[74] D. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, pp. 510-515, 1965.

[75] D. de Solla Price, "A general theory of bibliometric and other cumulative advantage processes," *Journal of the American Society for Information Science*, pp. 292-306, 1976.

[76] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509-512, 1999.

[77] E. A. Bender and E. R. Canfield, "The asymptotic number of labeled graphs with given degree sequences," *Journal of Combinatorial Theory*, vol. Series A, pp. 296-307, 1978.

[78] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Advances in Physics*, vol. 51, pp. 1079-1187, 2002.

[79] A. Broder et al., "Graph structure in the web," *Computer Networks*, vol. 33, pp. 309-320, 2000.

[80] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, "Breakdown of the Internet under intentional attack ," *Phys. Rev. Lett.*, vol. 86, pp. 3682-3685, 2001.

[81] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Giant strongly connected component of directed networks," *Phys. Rev. E*, vol. 64, no. 25101, 2001.

[82] "The National Laboratory for Applied Network Research (NLANR), National Science Foundation." [Online]. Available: http://moat.nlanr.net/. [Accessed: Nov-2011].

[83] A. Oram, Ed., *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, CA: O'Reilly & Associates, Inc., 2001.

[84] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, pp. 41-42, 2001.

[85] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651-654, 2000.

[86] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, "Classes of small-world networks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 21, pp. 11149-52, Oct. 2000.

[87] ICEBE'05, "IEEE International Conference on e-Business Engineering (ICEBE)." 2005.

[88] A. Hess, E. Johnston, and N. Kushmerick, "ASSAM: A tool for semi-automatically annotating semantic web services," in *Proc of the 3rd International Semantic Web Conference*, 2004.

[89] U. Küster, B. König-Ries, and A. Krug, “OPOSSum - An Online Portal to Collect and Share SWS Descriptions,” in *International Conference on Semantic Computing*, 2008, pp. 480-481.

[90] J. Fan and S. Kambhampati, “A snapshot of public web services,” *ACM SIGMOD Record*, vol. 34, no. 1, p. 24, Mar. 2005.

[91] “Salcentral website.” [Online]. Available: http://www.salcentral.com.

[92] “XMethods website.” [Online]. Available: http://www.xmethods.net/ve2/index.po.

[93] “seekda! website.” [Online]. Available: http://webservices.seekda.com/.

[94] “Web Service List website.” [Online]. Available: http://www.webservicelist.com.

[95] “programmableweb website.” [Online]. Available: http://www.programmableweb.com.

[96] “GeoNames website.” [Online]. Available: http://www.geonames.org/export/ws-overview.html.

[97] “WebserviceX.NET website.” [Online]. Available: http://www.webservicex.com/ws/default.aspx.

[98] “Bindingpoint website .” [Online]. Available: http://www.bindingpoint.com/.

[99] V. Batagelj, A. Mrvar, and M. Zaveršnik, “Pajek - Program for Large Network Analysis.” [Online]. Available: http://pajek.imfm.si/doku.php?id=pajek. [Accessed: Nov-2011].

[100] “The R Project.” [Online]. Available: http://www.r-project.org/.

[101] A. Clauset, C. R. Shalizi, and M. E. J. Newman, “Power-Law Distributions in Empirical Data,” *SIAM Review*, vol. 51, pp. 661-703, 2009.

[102] C. Cherifi, V. Labatut, and J. F. Santucci, “On Flexible Web Services Composition Networks,” in *Digital Information and Communication Technology and Its Applications*, 2011, vol. 166, pp. 45-59.

[103] S. Fortunato, “Community detection in graphs,” *Physics Reports 486*, pp. 75-174, 2010.

[104] M. A. Porter, J.-P. Onnela, and P. J. Mucha, “Communities in Networks,” pp. 1082-1097, Feb. 2009.

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

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

[107] 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.

[108] 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.

[109] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, “Self-similar community structure in a network of human interactions,” *Physical Review E*, vol. 68, no. 6, Dec. 2003.

[110] 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.

[111] 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.

[112] 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.

[113] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Physical Review E*, vol. 69, no. 6, p. 5, 2003.

[114] 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.

[115] 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.

[116] 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.

[117] 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.

[118] 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.

[119] 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.

[120] 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.

[121] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Physical Review E*, vol. 74, no. 1, p. 16110, 2006.

[122] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 76, no. 3 Pt 2, p. 036106, 2007.

[123] M. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E*, vol. 69, no. 2, Feb. 2004.

[124] M. E. J. Newman, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577-8582, 2006.

[125] S. Fortunato and M. Barthélemy, “Resolution limit in community detection,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36-41, Jan. 2007.

[126] W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods,” *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846-850, 1971.

[127] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2, no. 1, pp. 193-218, Dec. 1985.

[128] A. Clauset, “Finding local community structure in networks,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 72, no. 2 Pt 2, p. 7, 2005.

[129] L. Danon, A. Diaz-Guilera, and A. Arenas, “Effect of size heterogeneity on community identification in complex networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 11, p. 6, 2006.

[130] P. Pons, “Détection de communautés dans les grands graphes de terrain,” Paris 7, 2007.

[131] L. Donetti and M. A. Munoz, “Improved spectral algorithm for the detection of network communities,” *Aip Conference Proceedings*, p. 4, 2005.

[132] H. Zhou and R. Lipowsky, “Network Brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities,” *Computational ScienceICCS 2004*, pp. 1062-1069, 2004.

[133] J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization.,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, vol. 72, no. 2 Pt 2, p. 027104, 2005.

[134] T. Bennouas, M. Bouklit, and F. de Montgoler, “Un modèle gravitationnel du web,” in *5ème Rencontres Francophones sur les aspects Algorithmiques des Télécommunications (Algotel)*, 2003.

[135] K. Steinhaeuser and N. V. Chawla, “Identifying and evaluating community structure in complex networks,” *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413-421, Apr. 2010.

[136] E. Navarro and R. Cazabet, “Détection de communautés, étude comparative sur graphes réels ,” in *MARAMI*, 2010.

[137] G. K. Orman, V. Labatut, and H. Cherifi, “Qualitative Comparison of Community Detection Algorithms,” in *The International Conference on Digital Information and Communication DICTAP 2011*, 2011, vol. 167, pp. 265-279.

[138] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814-818, 2005.

[139] C. Cherifi, V. Labatut, and J. F. Santucci, “Web Services Dependency Networks Analysis,” in *International Conference of New Media and Interactivity (NMI 2010)*, 2010, pp. 115-120.

[140] C. Cherifi, V. Labatut, and J. F. Santucci, “Topological Properties of Web Services Similarity Networks,” in *Strategic Advantage of Computing Information Systems in Enterprise Management*, M. Sarrafzadeh and P. Petratos, Eds. Athens, Greece: ATINER, 2010, pp. 105-117.

[141] C. Cherifi, V. Labatut, and J. F. Santucci, “Benefits of Semantics on Web Service Composition from a Complex Network Perspective,” in *International Conference on Networked Digital Technologies (NDT 2010)* , 2010, vol. 88, pp. 80-90.

[142] C. Cherifi, Y. Rivierre, and J.-F. Santucci, “WS-NEXT, a Web Services Network Extractor Toolkit,” in *International Conference on Information Technology (ICIT’11)*, 2011.

[143] C. Cherifi and J.-F. Santucci, “Analyzing Web Services Networks: A WS-NEXT Application,” *Ubiquitous Computing and Communication Journal*, 2011.

[144] C. Aksoy, V. Labatut, C. Cherifi, and J. F. Santucci, “MATAWS: A Multimodal Approach for Automatic WS Semantic Annotation ,” in *International Conference on Networked Digital Technologies, NDT 2011*, 2011.