# Analyse sémantique des réseaux sociaux

Guillaume Ereteo

# Semantic Social Network Analysis

## Ph.D. thesis

Defended on the 11th of April 2011 by Guillaume Erétéo

Jury:

- President : Fabrice Rossi (Telecom ParisTech)
- Reporters : Marie-Aude Aufaure (Ecole Centrale Paris)
  Pascale Kuntz (University of Nantes)
- Directors : Michel Buffa (I3S, University of Nice - Sophia Antipolis)
  Fabien Gandon (INRIA Sophia Antipolis)
- Invited: Patrick Grohan (Orange Labs - Sophia Antipolis)

Orange Labs
Telecom ParisTech
INRIA Sophia Antipolis – Méditerranée

à Audrey, Mylène, Maman, Papa, Mich et Fab

"I hope that while so many people are out smelling the
flowers, someone is taking the time to plant some."
Herbert Rappaport

## Remerciements/Thanks

- A Audrey, Mylène, Maman, et Papa, pour votre amour qui me permet de rêver, de créer et d'avancer chaque jour.
- A Michel Buffa et Fabien Gandon, mes directeurs de thèse et amis, pour tout ce que vous m'avez appris et d'inoubliables moments de bonne humeur.
- A Olivier Corby pour tes indispensables contributions à mes travaux et tes précieux enseignements.
- A Alain Giboin pour nos fructueuses et intéressantes discussions.
- A toutes les personnes avec qui j'ai collaboré au sein du projet ISICIL et des équipes ACACIA, EDELWEISS, KEWI et PUPE d'Orange Labs.

# Abstract

The outburst of social functionalities in web-based applications has fostered the deployment of a social media landscape where people freely contribute, gather and interact with each other. The integration of various means for publishing and socializing allows us to quickly share, recommend and propagate information to our social network, trigger reactions, and finally enrich it. These shared spaces fostered the creation and development of interest communities that publish, filter and organize directories of references in their domains at an impressive scale with very agile responses to changes.

In order to reproduce the information sharing success story of the web, more and more social platforms are deployed into corporate intranets. However, the benefit of these platforms is often hindered when the social network becomes so large that relevant information is frequently lost in an overwhelming flow of activity notifications. Organizing this huge amount of information is one of the major challenges of Web 2.0 to achieve the full potential of Enterprise 2.0, i.e., the efficient use of Web 2.0 technologies like blogs and wikis within the Intranet.

This thesis proposes to help analyzing the characteristics of the heterogeneous social networks that emerge from the use of web-based social applications, with an original contribution that leverages Social Network Analysis with Semantic Web frameworks. Social Network Analysis (SNA) proposes graph algorithms to characterize the structure of a social network and its strategic positions. Semantic Web frameworks allow representing and exchanging knowledge across web applications with a rich typed graph model (RDF), a query language (SPARQL) and schema definition frameworks (RDFS and OWL). In this thesis, we merge both models in order to go beyond the mining of the flat link structure of social graphs by integrating a semantic processing of the network typing and the emerging knowledge of online activities. In particular we investigate how (1) to bring online social data to ontology-based representations, (2) to conduct a social network analysis that takes advantage of the rich semantics of such representations, and (3) to semantically detect and label communities of online social networks and social tagging activities.

# Résumé

L’explosion des fonctionnalités sociales au sein des applications du Web a favorisé le déploiement d'un panorama de médias sociaux permettant aux utilisateurs de librement contribuer, de se regrouper et d’interagir entre eux. La combinaison de divers moyens de publication et de socialisation permet de rapidement partager, recommander et propager l'information dans son réseau social, ainsi que de solliciter des réactions et de nouvelles contributions. Ces espaces partagés ont favorisé la création et le développement de communautés d'intérêts qui publient, filtrent et organisent de vastes répertoires de références dans leurs domaines, avec une impressionnante réactivité aux changements.

Afin de reproduire les succès du Web dans la gestion d'information, de plus en plus de plates-formes sociales sont déployées dans des intranets d'entreprise. Cependant, l'avantage de ces plates-formes est fortement atténué lorsque le réseau social devient si grand que les informations pertinentes sont noyées dans des flux continus de notifications. Organiser cette énorme quantité d'informations est l'un des défis majeurs du Web 2.0 afin de tirer pleinement partie des bénéfices de l'Entreprise 2.0, à savoir, l'utilisation des technologies du Web 2.0, tel que les blogs et les wikis, dans un intranet.

Cette thèse propose d’améliorer l’analyse des réseaux sociaux multiples et variés émergeant des usages sociaux du Web, au travers d’une contribution originale qui enrichit l’analyse des réseaux sociaux avec les technologies du Web Sémantique. L’analyse des réseaux sociaux propose des algorithmes de graphes pour caractériser la structure d'un réseau social et ses positions stratégiques. Les technologies du Web Sémantique permettent de représenter et d’échanger les connaissances entre des applications distribuées sur le Web avec un modèle de graphes richement typés (RDF), un langage de requête (SPARQL) et des langages de description de modèles (RDFS et OWL). Dans cette thèse, nous fusionnons ces deux modèles afin d'aller au-delà de l’analyse structurelle des graphes sociaux en intégrant un traitement sémantique de leur typage et des connaissances qu’ils contiennent. En particulier nous examinons comment (1) modéliser des données sociales en ligne à base d’ontologies, (2) réaliser une analyse du réseau social qui tire partie de la sémantique de ces représentations, et (3) détecter et étiqueter explicitement des communautés à partir de réseaux sociaux et de folksonomies.

# Sommaire

# 1. Introduction

*« The web is more a social creation than a technical one. » Tim Berners-Lee*

“Social Web” sounds like a pleonasm: user interactions and social networks are among the cornerstones of the Web. Human participation and freeform contributions are at the core of most popular web sites, creating shared spaces where people can freely gather, interact, and explicitly connect. From these usages, online communities of interests spontaneously emerge with roles and life cycles that are inherent in their members’ interactions and involvements. Guided by common interests and goals, these communities publish, filter and organize directories of references in their domains at an impressive scale with very agile responses to changes. Now, we have access to an ever-growing long tail of information and knowledge.

The main problem is no more to collect and publish resources but mainly to structure and mash them in a way that matters to people and to their communities. Consequently, intelligent agents are crawling web resources, mining and indexing them in order to provide added-value services and extended information to web users. Interested by the audience driven through such activities, content providers make explicit and available their public data through the form of API or mark-ups in their pages. The activity of these agents is made easier and easier by the growing adoption of Semantic Web technologies to capture, publish and access data with standard machine-readable formats and protocols. In particular, we are witnessing the outburst of standard semantic mark-ups inside HTML pages, thanks to their consideration by biggest web actors (e.g. Google, Facebook, Yahoo). This exponential growth of readily available semantic data foster the deployment of more and more intelligent software that consume these linked and structured data to personalize, enrich and multiply user experiences (e.g. web augmented reality).

Intranets of organizations are progressively reproducing various web evolutions and web based social applications are progressively deployed inside companies. For instance, Wikis are used to foster collaborative editions and knowledge capture, and social networking services to increase and ease sharing between employees. Intranet users are now able to partially adapt the flow of information inside the company to their daily tasks and evolving needs. However, social web applications inside intranets are more often disconnected, and corporate information is still more structured according to the organization chart rather than to how people use it. Beyond the reluctance related to emerging and auto-organized information, data that are produced by these applications lack the semantics and interoperability to be mashed and integrated in the intranet structure. The adoption of Semantic Web technologies could greatly benefit such social intranet by turning its information into structured data and connect it. Once semantically revealed, structured and connected, social data can in turn be exploited to develop functionalities that will structure information according to the need and the use of intranet users.

In previous researches on semantic wikis [Buffa et al 2008a], we investigated how the integration of Semantic Web technologies in a wiki could enhance the experience of its users and help a community build and structure a shared vocabulary. On one hand, we used Semantic Web technologies to manipulate the inner structure of the wiki by typing its different elements with the concepts from a "wiki ontology" ("document", "page", "tag", "link", "backward link"); thus, we were able to reason on this structure, enrich it, and interoperate with others wikis. On the other hand, SweetWiki enabled its users to annotate pages with their own vocabulary that they can freely modify and restructure, through a user friendly interface (e.g. add/merge/remove concepts or declare hierarchical links). This synergy between automatically generated metadata and human contributions offers a rich structuring and interoperability of the wiki data while answering the specificities and evolving needs of the user community.

Several researches have been conducted to develop this social semantic perspective of web based applications, and we now dispose of standards to capture, to represent and to interlink socially produced and structured metadata. However, this important step toward applications that easily collect, mash and publish data, puts users and companies in front of a huge amount of social signals that need to be filtered and organized to avoid hindering their initial benefits. In particular, socially issued metadata embed an emergent structure that is inherent in user relations, interactions, and affiliations. Revealing this social structure would enable its exploitation to help filtering and organizing this huge amount of data.

This thesis investigates methods for identifying the social structure emerging from the semantic representation of online social activities. Building on top of Semantic Web technologies and classical graph theory, we propose a novel approach to take benefits of both models and conduct a semantic social network analysis. We will see how to semantically represent, link and access online social networks, how to enable classical operators of social network analysis to consider the semantics of these networks, and how these semantics could be exploited to enhance community detection.

These researches were part of the ISICIL[1] project within the PUPE team of Orange[2] Labs, the Edelweiss[3] team of INRIA - Sophia Antipolis[4] and the Kewi team from the I3S Laboratory of the University of Nice. The ISICIL project proposes to study and to experiment the usage of Web 2.0 tools enhanced by Semantic Web technologies to assist corporate intelligence tasks. The PUPE team investigates prospective business services. The research team Edelweiss aims at offering models, methods and techniques for supporting knowledge management and collaboration in virtual communities interacting with information resources through the Web, , and collaborates a lot with the Kewi research team on these thematics.

This thesis is organized as follow:

**Chapter 2 presents the scenario that motivated the realization of this thesis and the ISICIL project.** The deployment of social web applications in corporate intranets promises to conduct innovative intelligence tasks, taking great benefits of a smart exploitation of the social signals emerging from free online contributions. However, in order to deal with the reactivity challenge of business intelligence, the numerous signals produced by Web 2.0 applications have to be structured and filtered.

**Chapter 3 reviews the literature and definitions of the basic notions related to social network analysis and online social networks.** We present the traditional methods used to capture and represent social networks, the different metrics and algorithms of social network analysis, and their application to online social networks.

**Chapter 4 presents how Semantic Web technologies enable us to structure, link and exchange social networking data across web sites**. Semantic web technologies provide a whole stack of languages and protocols to describe resources, to define vocabularies, to query and access such representations. In particular, many vocabularies have been designed to represent persons, relationships and web based activities.

**Chapter 5 presents the conceptual stack we designed to conduct a semantic social network analysis**. We extend social network analysis operators using Semantic Web frameworks to include the semantics used to structure social links, and we propose a model to enrich social data with the results of the analysis.

**Chapter 6 proposes a semantic algorithm, SemTagP, to label and detect communities.** This algorithm not only offers to detect but also to label communities, taking benefits of the tags used by people to classify web resources as well as the

[1] Information Semantic Integration through Communities of Intelligence online, http://isicil.inria.fr

[2] http://www.orange.com

[3] http://www-sop.inria.fr/edelweiss/

[4] http://www.inria.fr/centre-de-recherche-inria/sophia-antipolis-mediterranee

semantic relations that can be inferred between tags. Doing so, we are able to refine the partitioning of the social graph with semantic processing and to label the activity of detected communities.

**Chapter 7 discusses some important issues and perspectives that I would like to address in future works**. In particular, we discuss the importance of considering temporal data in social network analysis, we raise the time and space complexity of our approach for scaling to very large networks, and we propose some elements to turn the result of a semantic analysis into functionalities.

This thesis is organized in order to progress from the initial scenario and problems that motivated these researches to the final technical solutions that advance its resolution. Chapter 2 and 3 define the general context of and the positioning of this thesis in respect with existing literature. Chapter 4 presents and argues the technological choices in which we ground our solution. Chapter 5 and 6 describe the contributions of this thesis and the experiments that were conducted to assess and evaluate the presented solutions. Finally, Chapter 7 presents perspectives that I consider as important evolutions of this thesis and that I would like to address in further researches.

# 2. Motivating Scenario

*« The intranet tends to follow trends from the web and social Networking is no exception »*
*Nielsen Norman Group*

Since web users are core elements of most online applications, emergent contributions and free interactions form the main content of the web. People express themselves online, connect to exchange and to stay in touch, spontaneously gather and interact on similar interests. The outburst of social applications on the web produced a dramatic shift in information sharing and content production. These applications turned the privileged professional activities of producing, publishing and distributing content into massive amateur activities, enabling anybody to contribute. Massive and free contributions on the web have made information produced, shared, and accessible at an impressive scale and speed. The freeform of these applications enabled the development of financially non profitable activities that were forsaken by professionals. For instance, authoring tools like blogs enable the creation of a long tail of numerous precious source of information to the attention of small communities, so small that they were not targeted by professional editors. Moreover, in their online publications, authors refer to other documents (produced either by themselves or by others) by the means of hyperlinks, which implicate them in the evolution of the inner structure and in the organization of the web. Users are even more involved in this organization as most of online applications, like blogs, Flickr and Youtube, introduce explicitly their users in the classification of content with freely chosen labels, named tags. Consequently, the huge amount of online content is now organized and filtered by the mass of people. As a side effect of this collaborative classification, users of these applications are able to spontaneously and massively gather on shared interests at an unexpected large scale just by using the same tags. In addition, users of these applications are provided with advanced functionalities for connecting, interacting, and gathering. Due to the massive adoption of these practices, "online communities of interest have emerged and started to build directories of references in their domains of interest at an impressive speed and with very agile responses to changes in these domains" [Gandon et al 2009]. These communities can freely emerge and evolve to very large scales, for any purposes and in every domain.

Introducing such reactivity in the complex business processes and organizational chart of companies is an appealing opportunity to tackle the growing diversity of market and technological signals produced both internally and externally. Moreover, the generational turn over speaks up for challenging the acceptance of a corporate use of web 2.0 applications. In 2015, the generation Y, used to social medias in their daily personal activities, will represent 15% of the European population and 40% of workers in France[5]. This generation will ease and probably argue for the adoption of social medias in enterprises both for internal collaboration, public relationships and market insight.

In the first section of this chapter, we will discuss the benefits and the issues of introducing Web 2.0 applications inside companies, namely the enterprise 2.0. In the second section we focus on the ISICIL project in which we investigate the application of enterprise 2.0 to business intelligence. Finally, we discuss the need of understanding emergent social structures and the lack of tools to reach such goal.

## 2.1 Enterprise 2.0

More and more social solutions (e.g. Socialtext[6]) are being deployed in corporate intranets to reproduce inside corporations the information sharing success stories from the open web. This new trend is often called Enterprise 2.0, which is defined by [McAfee 2009] as follow:

**Definition 1. Enterprise 2.0:** the use of emergent social software platforms within companies, or between companies and their partners or customers.

**Definition 2. Emergent social software platforms**: digital environments in which contributions and interactions are (1) globally visible and persistent over time, (2) performed with social softwares that enable people to gather, connect or collaborate through computer-mediated communication and to form online communities (3) emergent, freeform, with patterns and structure inherent in people’s interactions.

Introducing such platforms inside an organization can provide different benefits for managing information and enhancing collaboration by enabling employees to:

- easily share and publish the content and knowledge they discover or produce.

[5] http://fr.wikipedia.org/wiki/Génération_Y

[6] http://www.socialtext.com/

- collaboratively filter and organize both internal and external documents and sources.
- spontaneously connect and gather on related working topics and objectives.
- easily search and find information, documents and experts

In order to achieve these benefits, [McAfee 2009] introduces the SLATES acronym that defines the features that should provide emergent social softwares:

- Search for enabling employees to find information.
- Links for strengthening the connectedness of information and fostering the discovery of new sources.
- Authoring for providing employees with an easy and non technical way of publishing information.
- Tags for enabling people to easily organize content with freely chosen labels, and enable a large scale and human classification.
- Extensions for automatically proposing the discovery of new content suggested by pattern matching algorithms.
- Signals for enabling web users to subscribe to targeted sources and topics, and to be automatically notified of new publications.

However, while the freeform of these platforms enables to collectively handle "the diversity and the mass of information sources" [Gandon et al 2009], it is also their main problem for their acceptance in a corporate context. Firstly, companies have been working for decades at limiting the number of collaborators and actions each one of their employees has to handle for optimizing individual performances. Companies are driven by well defined business processes and formal structures while emergent social software platforms are characterized by free activities and freely evolving social structures. It is one of the main reasons of the reluctance of many companies to introduce social solutions in their business practices. Secondly, some companies face strong information security and confidentiality restrictions and cannot even accept unexpected practices and interactions in their processes. Finally, some decision makers simply fear their employees will loose time and efficiency in sharing and consuming non controlled and unsupervised information. Consequently, social web applications cannot just be deployed in companies' intranets without being fully integrated in their formal organizations and their business processes. Many challenges have to be tackled to fully achieve the objectives promised by the advocate of the enterprise 2.0. All the corporate reluctances that are cited above highlight common expectations for using these tools: a better effectiveness, supervision and control of the flow of data and information.

## 2.2 ISICIL: Information Semantic Integration through Communities of Intelligence onLine

In the ISICIL project, we investigate the application of enterprise 2.0 to business intelligence, with a framework that takes advantage of collaborative platforms to allow conducting innovative strategic watch. The goal is to introduce social interactions into every step of the intelligence cycle: searching, monitoring, collecting, handling, disseminating. Information produced by different sources becomes socially connected, can be quickly shared and permanently enriched with comments, tags and new related sources. Figure 1 shows how emergent social softwares are integrated in every steps of the business intelligence cycle. People are not only connecting together, they connect themselves to documents, data and information:

- Searching is no more a lonely task consisting in looking for relevant information and sources; we now collectively search information in document but also through people and expert that become one of the main sources of information. Even more, searching is sometime unnecessary, information is simply propagated to people by people.
- Monitoring is not only about monitoring document sources but about listening to human sensors, namely collaborators and expert activities.
- Collecting consists in selecting and organizing the information, a task supported by a collaborative pre-treatment of the social network which proposes its insight on sources and information, and organizes it, in particular through the means of social tagging.
- Analysing consists in synthesizing the collected information to detect and highlight weak signals, tendencies and prospective deductions. This step is crucial to support decision making and is once again preciously leveraged by the insight of the crowd and the benefit of collaboration.
- Disseminating is greatly favoured by social networks and online social applications: information and documents are better connected and are de facto easier to find while their propagation is empowered by people that are better interconnected, whatever their locations and affiliations are.

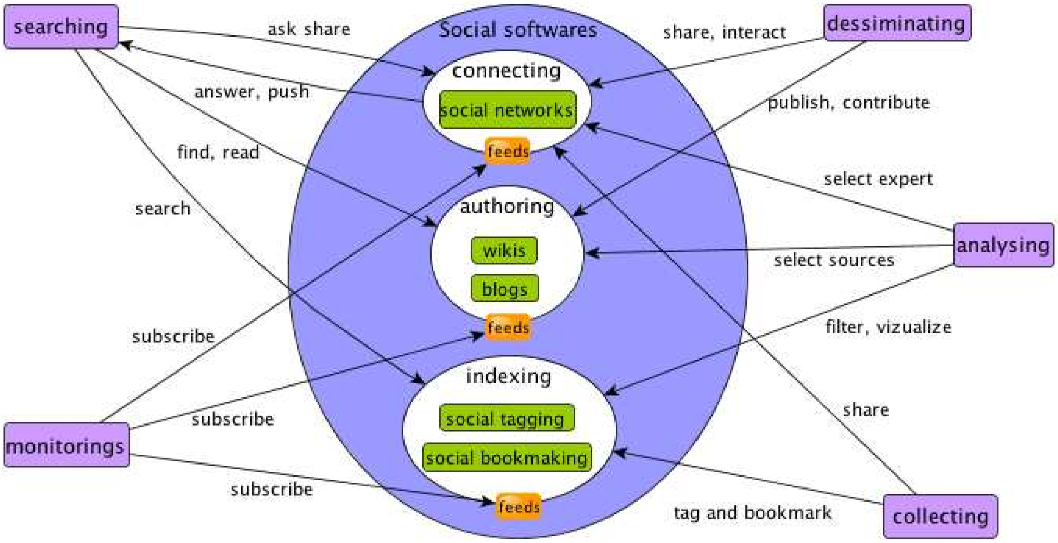


**Figure 1.** Business intelligence 2.0.

While this socialization of business intelligence leverages the information management and increases the cooperation, it also augments the amount of information employees are exposed to. The benefit of collaborative tools is often hindered when the social network becomes so large that relevant information is lost in an overwhelming flow of notifications. Users are facing huge objects, evolving all the time with a growing amount of information that exceeds their attention span, which is unacceptable in organizations. Moreover, it complicates the management of confidentiality and security of strategic information. Organizing this huge quantity of information is necessary for achieving the full potential of Enterprise 2.0. (1) We need to link and organize the huge amount of shared and produced data. (2) We need to reconcile the free activity of web 2.0 applications with formal processes of companies. (3) We need to reconcile spontaneous relationships and community structure that emerge through online collaboration with formal organizational charts.

In the ISICIL project, we propose to tackle these issues with a multidisciplinary approach [Gandon et al 2009] to deal with:

- the sociological and usability challenges for reconciling web 2.0 approaches with organizational charts and processes.
- the technological challenges of capturing, representing and processing the diversity of decentralized data emerging from the use of different online social applications.

We focus here on the technological issues, which are tackled with Semantic Web technologies by offering data interoperability between applications and for leveraging information processing. We need to deal with heterogeneous data (involving actors, content and relationships) that are generated and spread across the internet and intranet networks on different sites. Semantic web technologies answer this problem with standard languages and protocols for

- describing and exchanging resources and data across applications on a network with a uniform structure.
- representing and linking the models and the domain vocabularies used to define the semantics of these descriptions.
- querying and accessing distant data describing both resources and models.

These technological advances enable us to handle the indexing and the processing of the data that are produced by the decentralized and disconnected web applications. First, interoperability between applications and the exchange of data is enhanced by the use of standard languages and protocols for describing, querying and accessing data on networks. Then processing and handling the semantics and the models use to structure exchanged data is consequently eased with the standardization and the linkage of their representation.

[Passant et al 2009] argue that a Semantic Web layer on top of an enterprise 2.0 information system enable to deal with "information fragmentation and heterogeneity of data formats", "knowledge integration and re-use", and "tagging and information retrieval". Once structured and represented in a uniform way, social data can be mined and leveraged to meet enterprise requirements for (1) linking information, (2) detecting and structuring emergent process, and (3) providing insight into and from spontaneous communities and emergent collaborative structures.

The Figure 2 describes how a business intelligence conducted with emergent social software platforms can be enhanced by a semantic layer on top of all these applications:

- Searching is enhanced and better assisted with a global and more relevant search across applications, thanks to the explicit semantic of the nature, the context and the meaning of data.
- Monitoring is enhanced by a semantic based filtering and structuring of data, and augmented with related information by a semantic processing of data links.
- Collecting is assisted by suggestions of concepts for classifying content, based on the semantic inferred from emerging vocabulary. Moreover, when collecting and classifying content, users become implicated (both implicitly and explicitly) in the structuring process of their company's vocabulary.
- Anlysing will benefits of semantic perspectives on collected data and of social signals, which will enable advanced interactions and smart filtering.
- Disseminating will be supported in the targeting of relevant communities for sharing produced content, while the members of these communities will be notified of the creation of content that is relevant to their interests.

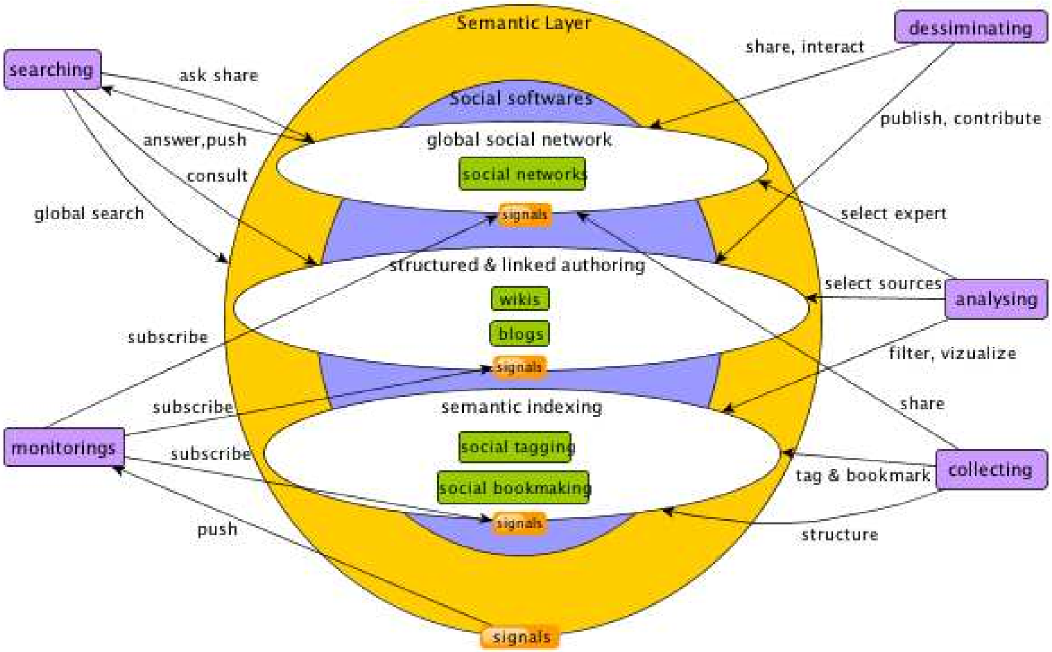


**Figure 2.** Business intelligence 2.0 enhanced by a semantic layer.

In order to achieve such an evolution of the business intelligence cycle we not only need to support web 2.0 approaches with Semantic Web technologies but also to tackle two important scientific challenges:

- Integrating the light classifications of resources performed by web 2.0 users with freely chosen labels and formal ontologies to get the best of both worlds. In other words the goal is to classify resources with an evolving vocabulary that is both structured and representative of users' knowledge perception.
- Reifying and exploiting the dynamic and rich social networks that are embedded in the emerging social data of web 2.0 applications, in order to foster interactions and collaborations, to help user positioning in these networks and filtering overwhelming social notifications.

While the first issue is tackled by Freddy Limpens in its Ph.D. thesis [Limpens 2010], we focus here on the analysis of the semantic representation of web 2.0 social networks.

## 2.3 Semantic Analysis of Social Networks

The social data that emerge in online social applications embed rich social links, between their users, that have to be revealed and reified in order to be mined and exploited. In particular, these applications enable their users to connect, interact and develop interest affiliations between each other, which enable us to build and mine the resulting social networks. However the structures of these social networks are complex to represent, due to the multiplicity of context, roles and identities, and to their distribution across applications. Each user of an application represents a person, in a particular role and a given context that constitute a fragment of its identity. Consequently, a person develops different social links, across several applications, which are contextualized by the different fragments of its identity. An effective mining of the resulting global social network should consider such specificities and require thus an adequate representation.

In the previous section we argued that Semantic Web technologies answer the problematic of exchanging, mashing and querying data across applications. Based on these technologies, we need to reuse existing models and develop new ones, if necessary, to smartly represent people, user profiles and their different social links for revealing the online social networks they form. Once represented in a uniform structure, these social networks can then be mined for extracting the metrics that will be used for managing social data.

Social Network Analysis is particularly well suited for understanding and determining the global structure of a social network, the distribution of actors and activities, and the strategic positions and actors. The result of the network analysis can be exploited for leveraging the social experience of collaborative tools. On one hand, we can better organize and filter social data in every step of the business intelligence process. During the searching, monitoring, collecting and handling steps, the presentation of social data to the users should consider the insight of a network analysis as well for classification purposes as for information quality indicators. During the disseminating step, the network analysis metrics will help propagating the produced information toward the relevant part of the network and connect it to the targeted communities. On the other hand, the analysis can be used for strengthening the network structure in order to increase its efficiency, both locally and globally. At a local scale, the analysis can be used to assist applications' users in maintaining their relationships and developing relevant new ones that could best serve their efficiency. At a global scale the analysis could be used to stimulate new connections that would empower the whole network efficiency, such as bridging disconnected communities that would benefit from collaborating

However, social network analysis' algorithms are only based on the linking structure of the network and do not exploit the semantics that are embedded in such typed representations from which they could highly benefit. The richness and the specificities of online social networks offer many perspectives for conducting more accurate analyses. In particular, the online artifacts that mediate interactions and develop affiliations provide social networks with the purposes of the creation, the maintenance or the disappearance of social links. In addition, the semantic structuring of the vocabulary generated by the users provides social networks with the knowledge that is produced, maintained and shared by their members to support their exchanges. Social network analysis is now provided with these multidimensional representations that include not only the linking structure but also the shared knowledge of the social network.

Our objective is to leverage social network analysis metrics for handling the semantic representations of social networks, which is a necessary step for fully achieving the social evolution of the business intelligence proposed by the ISICIL project.

# 3. State of the Art on Social Network Analysis and Its Application on the Web

*« When we change the way we communicate, we change the society » Clay Shirky*

Social Network Analysis (SNA) provides graph algorithms to characterize the structure of social networks, strategic positions in these networks, specific sub-networks and decompositions of people and activities. This domain has raised lots of interests and the outburst of social data on the web has led to the collection of the biggest social networks ever.

In this chapter, we will review (1) the traditional methods and models that are used to build and represent social networks, (2) the different metrics and algorithms of social network analysis, and (3) the applications of social network analysis to online social networks.

## 3.1 Capture, Detect and Represent Social Networks

A social network is made of actors that are linked by social relations. Social actors can be people, organizations, or groups of actors. A wide range of social relations exists between actors; we can group these relations in three categories:

- *explicit and declared relations* between humans.
- *interactions* between actors.
- *affiliation* between actors.

Explicit relations include all the relations we can define between persons (e.g. parent, sibling, cousin, friendship, love, simple acquaintance, co-worker, etc.), between persons and organizations (e.g. member, employee, etc.), and inside organizations (e.g. owner, manage, etc.).

Interactions represent all the exchanges that could be observed between actors such as a discussion, a collaboration, a meeting or any action that involves at least two actors. Some interactions actively implicate all the concerned actors, like a synchronous discussion. Others are initiated by some actors and target other actors, like a single message that has a sender and a recipient.

Affiliations correspond to any similarity between actors that links them, like, for instance, sharing the same attributes, the same interests, the same activities, the same objects, or the same organizations.

The type of social links is determinant in the construction of the corresponding social networks. While the first social networks have been collected and analyzed by interviewing people or by observing social actors, social networks have also been extracted from many sources in order to achieve different purposes and confirm varied sociological hypothesis.

### 3.1.1 Explicit Relationship Networks

Initially data about social networks were mainly collected with interviews, pen and paper. Social networks were built from experiment with people who declared explicitly their relationships with other persons. Today, online social network services offer huge databases of such declared relationships that are easier to collect on the web; this part will be detailed in chapter 4, for now we present examples of the famous historical social networks.

**One of the most popular social networks is the university-based karate club of Zachary [Zachary 1977]**. Following the appearance of different internal conflicts in the club, the social structure emerging from the evolution of social links, i.e. creation and destruction, highlighted a break-up in the social cohesion. Consequently, this karate club was divided in two separate clubs due to relationships fission. The corresponding dataset was collected with interviews about the relationships among actors with the purpose of understanding how the fission occurred in this network. By observing the evolution of social linkage, the authors have extracted patterns that enable us to predict an upcoming break-up in a social network. The Figure 3 proposes a visualisation of the collected network. Members of the first sub-club are represented by white circles and members of the other one are represented by grey squares. We can clearly witness a difference of social cohesion between the two groups and the whole network.

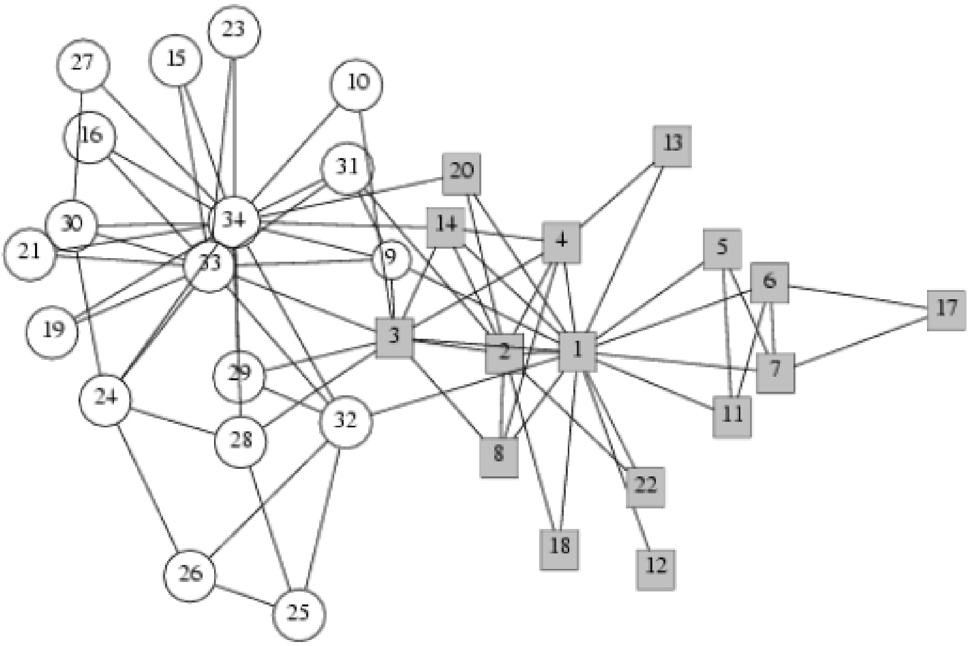


**Figure 3.** The Zachary karate club has been divided in two clubs in 1977. Members of the first club are represented by round white nodes and members of the second one by grey square nodes.

Many social networks were built from asking people to name some of their friends. Such social networks are consequently built from data such as *Peter states Jack is his friend*. Such methods may produce non reciprocal relationships as Peter can name Jack as its friend but not vice-versa. There are three ways to interpret such data. First we can infer that if Peter considers Jack as his friend, Jack also considers Peter as his friend, and we can define a symmetric friendship relation between Peter and Jack. Then we can also define a relationship only between persons that reciprocally declared them as friend. Finally we can simply define directed relationships and consider exploiting this specificity in the analysis. [Moody 2001] applied this interview method to build the friendship network of a U.S. school. This experiment highlighted the tendency of people to bond with people that share some attributes. He observed a clear separation between white and black individuals and between younger and older ones. The Figure 4 proposes a visualisation of this social network. The black people are on the right while the white people are on the left part of the visualisation. Similarly, we observe a horizontal division between younger and older people. The combination of age and colour attributes produces four main dense groups with poor linkage between these groups.

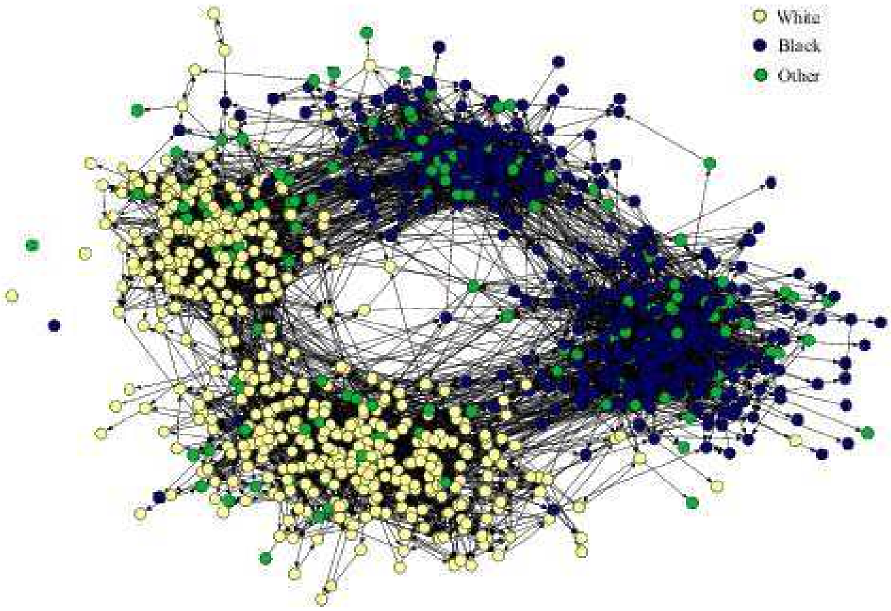


**Figure 4.** Frienship network of a U.S. school, which highlight a high tendency of children to bind with similar others.

### 3.1.2 Interaction Networks

Generally, social networks based on interactions are built from observation of actions, or traces of actions, involving at least two actors. Any action involving an exchange between actors is considered as an interaction and is used to set a relationship between them. The large variety of possible interactions between social actors offers many ways to detect and collect social network data.

Milgram conducted a famous experiment to measure the average distance between people in a very large social network [Milgram 1967]. people from one city of the United States were asked to send a letter to other people in a socially and geographically opposite city of the U.S. To make the letter reach its recipient, each holder of the letter had to write its name on the letter and give the letter to the person he thought was the closest from the final recipient. Milgram exploited the sequence of names on the letter to reconstitute the paths from the initial holders to the final recipients of the letters. Two persons which names are adjacent in the name list on a letter that reached its destination are linked by a “mail interaction”. Doing so, Milgram built a network from the traces on the letters, of the interactions between players of the experiment. He observed that the average paths between any two persons in the United States were of length 6. This result is also known as "six degrees of separation" (however, some criticisms of this experiment were recently published [Kleinfeld 2002]).

The U.S. College Football network is also an interaction network in which actors are football teams, namely organizations, and relations "represent regular-season games between the two teams they connect" [Girvan & Newman 2002]. Consequently, every time a game was played between two teams, a relationship was set, representing the interaction of playing a same game. Teams are grouped into conferences, namely divisions of teams, and more games are played between teams of the same conference than between teams of different conferences. Consequently Newman observed more relationships among members of a same conference than between members of different conferences. Consequently, this network has a strong community structure and has been frequently used by researchers as a dataset for evaluating community partition algorithms.

### 3.1.3 Affiliation Networks

A broad range of social networks have been inferred and collected by analyzing similarities between actors. Similarities between actors are frequently a source of interactions. Moreover people that share characteristics tend to behave similarly. Similarity based relationships are more generally called *affiliations*.

Social network analysis has gained a lot of interest from economical sciences. In particular it is interesting to analyze how organizations and their members interact in order to understand economical mechanisms. In particular, an affiliation social network was built from the membership to the director board of U.S. industry firms and banks [Mariolis 1975]. The actors of this social network are the directors. A link between two directors is set whenever two directors are members of the same directory board. The analysis of this affiliation network highlighted many interest conflicts and showed how banks control and interlock such industry directory boards.

Another interesting affiliation network is the social network populated by the characters of Victor Hugo's novel "Les Miserables" [Knuth 1993]. A relation is set between two characters of the novel when they co-appear in a scene, and thus an affiliation network is built. This social network has been used in many experiments, in particular for evaluating community detection algorithms. The Figure 5 highlights a community partition of this social network.

Many affiliation networks were also extracted from scientific paper databases, in order to build co-authorship networks. The actors are the authors of the papers. A relation is set between two actors who are co-authors of the same paper. These networks have been widely used to test social network analysis algorithms, in particular community detection algorithms and centrality algorithms. The Figure 6 represents the co-authorship network studied in [Newman 2006b].

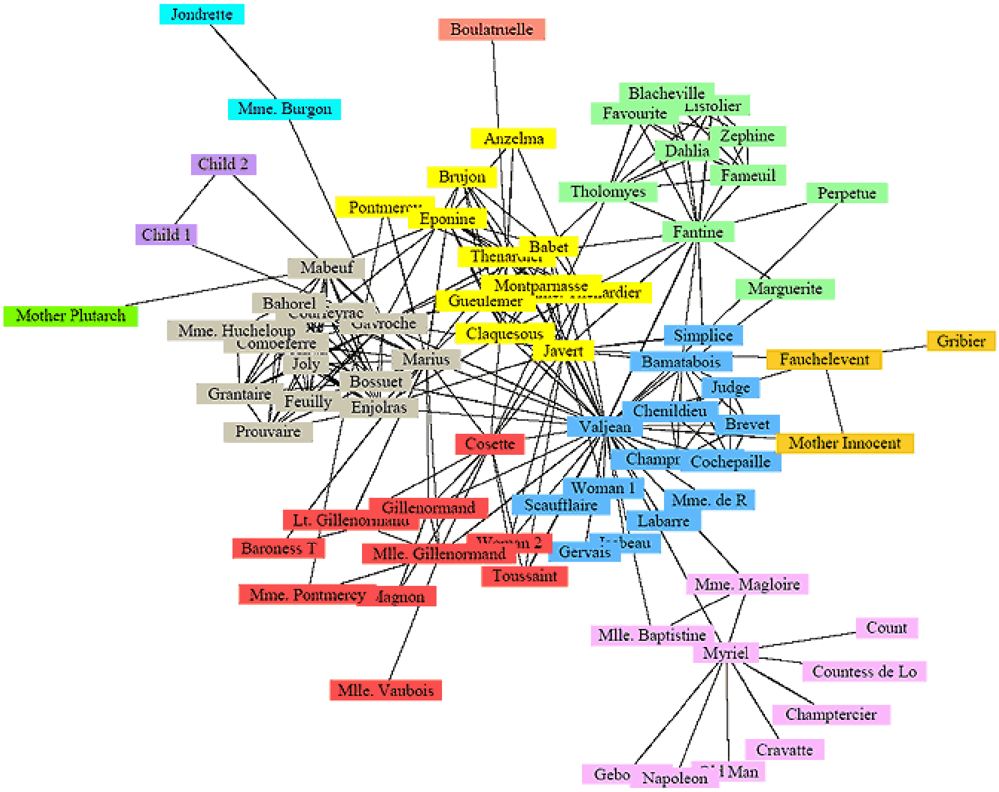


**Figure 5.** Co-appearance social network of the characters of Victor Hugo's novel "Les Misérables".

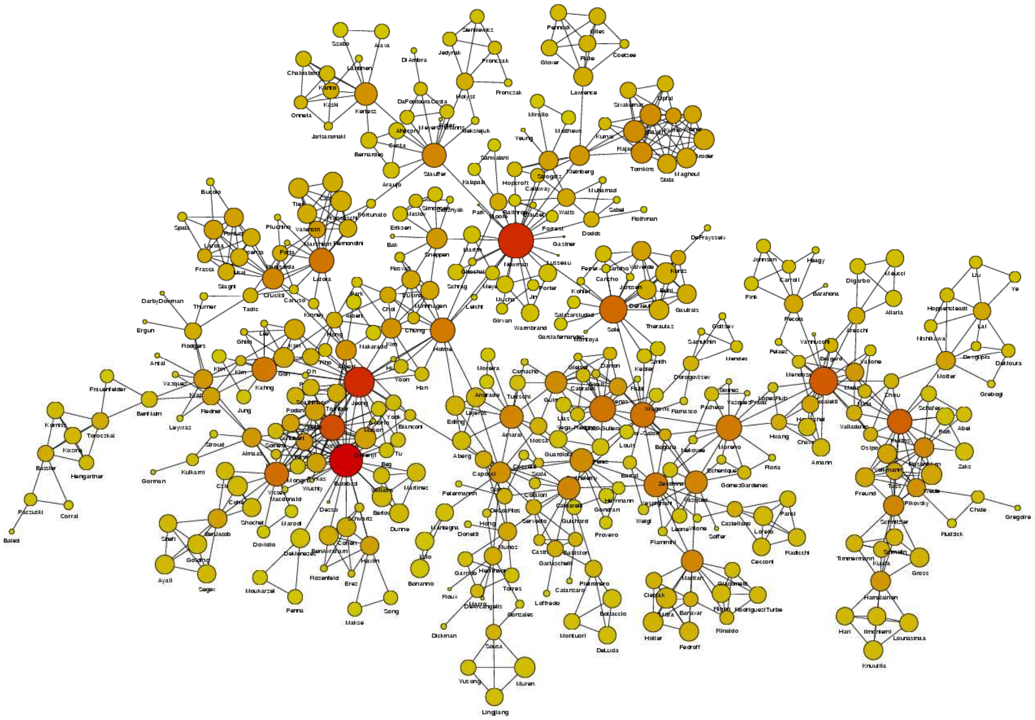


**Figure 6.** Visualisation of the co-authorship network studied in [Newman 2006b].

### 3.1.4 Social Network Representation

In order to graphically visualize social networks, in 1930's, Moreno systematized the first representations of social networks: the sociograms [Moreno 1933]. Sociograms consist in representing people by points and relationships by lines connecting points. These representations were also named 'web' due to their spider web aspect, this is an interesting unintentional coincidence of history. As little innovative as it may appears today, this type of visualisations offered to quickly detect some network features that are highlighted by specific visual patterns. As an example, Moreno introduced the concept of "star" for designing people having the most connections in a social network, due to the star shape formed by a point and its numerous connected lines. Sociograms were the first step for further involvement of mathematicians in social network analysis.

[Harary & Norman 1953] were among the first mathematicians who made the relation between graphs and sociograms and who built mathematical models of social networks based on graph theory. In a graph, the nodes represent the actors and the edges represent relationships. [Scott 2000] proposes an historical overview of the first applications of graph theory to social network analysis in the mid of the 20th century. Today, Graph structure has been adopted as the main mathematical model for social networks in social sciences, computer science or economical sciences. Having a mathematical model enables us to better formalize the analysis of social networks, and propose algorithms to detect and compute graph patterns that characterize social organizations.

#### 3.1.4.1 Definitions

**Definition 3. Node**: basic unit of a network that represents a resource, also called a vertex. In a social network we talk about actors or agents.

**Definition 4. Edge**: a connexion between two nodes. We also use the terms arcs or links.

**Definition 5. Hyperedge**: an edge than connects more than two nodes.

**Definition 6. Directed edge**: an edge used in only one direction, from its source node to its end node. In opposition, an undirected edge can be used in both directions and does not distinguish its extremity nodes.

**Definition 7. Weighted edge**: an edge with an assigned a value, called a weight, to represent the importance of this edge.

**Definition 8. Labelled edge**: an edge with a term used to label the relation.

**Definition 9. Graph**: a graph is defined by a set of nodes and a set of edges.

**Definition 10. Hypergraph**: an hypergraph is defined by a set of nodes and a set of hyperedges [Berge 1985]

**Definition 11. Directed graph**: a directed graph is defined by a set of nodes and and a set of directed edges.

**Definition 12. Weighted graph**: a weighted graph is defined by a set of nodes and a set of weighted edges.

**Definition 13. Labelled graph**: a labelled graph is defined by a set of nodes and a set of labelled edges.

**Definition 14. Multipartite graph**: a multipartite graph is decomposed in *k* set of nodes, each set contains a unique type of node, with edges that connect nodes of different sets.

**Definition 15. Bipartite graph**: a multipartite graph with only 2 types of nodes.

**Definition 16. Tripartite graph**: a multipartite graph with only 3 types of nodes.

**Definition 17. Degree**: the degree of a node is its number of adjacent edges.

**Definition 18. Path**: list of nodes of a graph, each linked to the next by an edge.

**Definition 19. Directed path**: a sequence of directed edges from a source node to an end node.

**Definition 20. Geodesic and shortest path:** shortest sequence of edges between two given nodes.

**Definition 21. Diameter**: the length of longest geodesics of the network.

**Definition 22. Complete graph**: a graph having an edge between any two pair of its nodes.

**Definition 23. Connected graph**: a graph having a path between any two pair of its nodes.

### 3.1.4.2 Notations

- A node is noted *v* and an edge between two nodes $v_i$ and $v_j$ is noted $(v_i, v_j)$.
- A graph is noted *G = (V, E)* with *V* a set of nodes, *E* a set of edges, $n = |V|$ the number of nodes, $m = |E|$ the number of edges and, $v_i$ the $i^{th}$ node.
- A subgraph of *G=(V,E)* is noted *G' = (V', E')* with $V' \subset V$, $E' \subset E$ | $(v_i, v_j) \in E' \Rightarrow v_i \in V' \& v_j \in V'$, $n' = |V'|$ and $m' = |E'|$.
- A bipartite graph is noted *G = (U, V, E)* with *U* and *V* two sets of nodes and *E* a set of edges, $v_i$ the $i^{th}$ node of V and $u_i$ the $i^{th}$ node of *U*.
- A tripartite graph is noted *G = (U, V, W, E)* with *U*, *V*, and *W* three sets of nodes and *E* a set of edges, $u_i$ the $i^{th}$ node of *U*, $v_i$ the $i^{th}$ node of V and, $w_i$ the $i^{th}$ node of W.
- The degree of a node $v_i$ is noted $d(v_i)$.
- $g_{ij}$ represent a geodesic between the nodes $v_i$ et $v_j$, and $|g_{ij}|$ represents the length of a geodesic between $v_i$ et $v_j$.

### 3.1.4.3 Using Different Types of Graphs

The famous social network of the Zachary karate club in Figure 3 is represented by a *simple graph* with undirected an unlabeled edges. More generally, social networks with symmetric relationships (e.g. married) are represented by undirected graphs. Inversely, *directed gr*aphs are well suited to model social networks with directed relationships (e.g. manages). For instance the interactions emerging from the sent letters in the Milgram experiment can be modelled with directed edges. In *weighted graphs*, weights are associated to edges to specify the intensity of the relationships, and in particular for representing the frequency of interactions between people. In the U.S. College Football network, weights on edges can be used to represent the number of games played between teams. *Labelled graphs* are well suited to represent social networks containing different types of relationships (e.g. family, friends, colleague). *Bipartite graphs* are generally used to model affiliation networks using two types of nodes, the actors and the objects of the affiliation, and edges that always link nodes of each type. For instance in a graph representing directory boards of companies, we have 2 types of nodes, the companies and their directors, with edges linking companies to their directors. More complex social network with complex relationships involving more than two types of resources (e.g. an author, a paper, and a keyword) are represented by hyperedges producing *hypergraphs*.

#### 3.1.4.4 Representing Graphs With Matrices

**Definition 24. Matrix**: A matrix is a rectangular table of values, in which each cell is noted $a_{ij}$ with $i$ and $j$ that are the row and the column of the cell.

Matrices are popular mathematical objects for handling graphs. Usually, when modelling a graph with a matrix, the rows and the columns of a matrix represent the nodes of the graph, and the value in the cell $a_{ij}$ in the matrix represents an edge (or an absence of edge) between the corresponding nodes $v_i$ and $v_j$. Generally, two types of matrices are used for representing social networks: (1) adjacency matrices which rows and columns represent *the same sets of nodes*, and (2) incidence matrices which rows and columns represent *different sets of nodes*.

**Definition 25. Adjacency Matrix**: An *adjacency matrix* is a squared matrix in which rows and columns are labelled by the same list of nodes (the $i^{th}$ row and the $i^{th}$ column represent the same node).

**Definition 26. Incidence Matrix**: An *incidence matrix* have two types of nodes (e.g. authors and papers) with rows representing one type and columns the other one.

An *adjacency matrix* is well suited to represent a graph with only one type of nodes (e.g. rows and columns that represent persons). Consequently, a graph $G = (V, E)$ with $n=|V|$ can be represented with a matrix $M$ with $n$ lines and $n$ columns. If there is no relationship between $v_i$ and $v_j$ this value is 0. If a relationship exists between $v_i$ and $v_j$ this value is 1 in the case of a non weighted graph, and the weight of the relationships in the case of a weighted graph. In the case of a directed graph the rows and the columns represent respectively the source nodes and the end nodes of edges. A directed edge from $v_i$ to $v_j$ will be only represented by a positive value in cell $a_{ij}$. It is also possible to use a negative value in the cell $a_{ji}$ to represent a directed edge from $v_i$ to $v_j$. The Figure 7 is an example of a matrix representing a social network built from the interaction (e.g. "collaborates with") of 4 employees. This social network is undirected, so we have $a_{ij}= a_{ji}$, i.e. the matrix is symmetric. The values that are contained in the matrix cells represent the intensity of the corresponding collaborations. The Figure 8 is a sample of the adjacency matrix representing the Zachary karate club of the Figure 3.

| | Employee1 | Employee2 | Employee3 | Employee4 |
|---|---|---|---|---|
| Employee1 | 0 | 1 | 3 | 1 |
| Employee2 | 1 | 0 | 1 | 0 |
| Employee3 | 3 | 1 | 0 | 2 |
| Employee4 | 1 | 0 | 2 | 0 |

**Figure 7.** Adjacency matrix of employees collaborating together, the value in a cell represents the number of shared projects between corresponding employees.

| | $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ | $V_7$ | … |
|---|---|---|---|---|---|---|---|---|
| $V_1$ | - | 1 | 1 | 1 | 1 | 1 | 1 | … |
| $V_2$ | 1 | - | 1 | 1 | 0 | 0 | 0 | … |
| $V_3$ | 1 | 1 | - | 1 | 0 | 0 | 0 | … |
| $V_4$ | 1 | 1 | 1 | - | 0 | 0 | 0 | … |
| $V_5$ | 1 | 0 | 0 | 0 | - | 0 | 1 | … |
| $V_6$ | 1 | 0 | 0 | 0 | 0 | - | 1 | … |
| $V_7$ | 1 | 0 | 0 | 0 | 1 | 1 | - | … |
| … | … | … | … | … | … | … | … | … |

**Figure 8.** Sample of the adjacency matrix of the Zachary karate club.

An *incidence matrix* is well suited to represent affiliation networks with bipartite graphs. Consequently, a bipartite graph $G = (U, V, E)$ with $p=|V|$ and $q=|U|$ can be represented with a matrix $M$ with $p$ lines and $q$ columns. In this case the value of the cell $a_{ij}$ represents the relation between nodes $v_i$ and $u_j$. The principles to define weighted and directed edges are identical to the principles of adjacency matrices, presented above. A $p*q$ incidence matrix can be converted into two squared adjacency matrices of size $p$ and $q$, one for each type of resource. The values of the cells of these adjacency matrices represent the number of shared connections in the incidence matrix. Figure 9 represents an example of incidence matrix built from the directory board of companies; rows and columns respectively represent companies and directors. The Figure 10 is the adjacency matrix of companies deduced from the shared directors of companies in the incidence matrix of Figure 9. For instance, the *company 1* and the *company 2* are both connected to the *director 1* and the *director 2*. Consequently, in the adjacency matrix of companies, the value of the cells $a_{12}$ and $a_{21}$ is 2, and represents the intensity of the interlock between the corresponding companies. Similarly the Figure 11 is the adjacency matrix of directors deduced from the companies shared by directors in the incidence matrix of Figure 9. The values of the cell of this matrix represent the intensity of collaboration of the corresponding directors.

| | Director 1 | Director 2 | Director 3 | Director 4 | Director 5 |
|---|---|---|---|---|---|
| Company 1 | 1 | 1 | 1 | 0 | 0 |
| Company 2 | 1 | 1 | 0 | 1 | 1 |
| Company 3 | 0 | 1 | 1 | 0 | 0 |
| Company 4 | 0 | 0 | 1 | 1 | 1 |

**Figure 9.** Example of incidence matrix of the social network of directory boards.

| | Company 1 | Company 2 | Company 3 | Company 4 |
|---|---|---|---|---|
| Company 1 | - | 2 | 2 | 1 |
| Company 2 | 2 | - | 1 | 2 |
| Company 3 | 2 | 1 | - | 1 |
| Company 4 | 1 | 2 | 1 | - |

**Figure 10.** Adjacency matrix of companies deduced from the Figure 9, each cell represents the number of directors shared by the corresponding companies.

| | Director 1 | Director 2 | Director 3 | Director 4 | Director 5 |
|---|---|---|---|---|---|
| Director 1 | - | 2 | 1 | 1 | 1 |
| Director 2 | 2 | - | 2 | 1 | 1 |
| Director 3 | 1 | 2 | - | 1 | 1 |
| Director 4 | 1 | 1 | 1 | - | 2 |
| Director 5 | 1 | 1 | 1 | 2 | - |

**Figure 11.** Adjacency matrix of directors deduced from the Figure 9, each cell represents the number of companies shared by the corresponding directors.

Finally particular matrices are frequently used to encode the degree of the nodes of the graph that they represent. Such matrices are particularly well suited to compute structural properties of the graphs.

**Definition 27. Degree Matrix**: The degree matrix of a graph is a diagonal matrix with the degree of the graph's nodes on the diagonals

**Definition 28. Laplacian Matrix**: The Laplacian Matrix (or Kirchhoff matrix) of a graph is the difference between its degree matrix and its adjacency matrix. The value of the cells of a Laplacian Matrix is defined as follow:

$$a_{ij} = \begin{cases} d(v_i) \quad if \quad i = j \\ -1 \ if \ i \neq j \ a\text{nd} \ (v_i, v_j) \in E \\ 0 \quad otherwise \end{cases}$$

## 3.2 Social Network Analysis

SNA tries to understand and exploit the key features of social networks in order to manage their life cycle and predict their evolution. Much research has been conducted on SNA using graph theory [Scott 2000] [Wasserman & Faust 1994]. Among important results is the identification of sociometric features *that characterize a network*. SNA metrics can be decomposed into two categories; (1) some provide information about the position of actors and how they communicate and (2) others give information about the global structure of the social network.

### 3.2.1 Strategic Position and Important Actors

**The *Centrality* highlights the most important actors and the strategic positions of the network.** The main question of centrality is to define what makes an actor more central than another one. Different criteria have been considered to define the centrality, and the chosen criteria enable to obtain different information about the position of actors. The three main definitions of centrality are resumed by [Freeman 1979]: the degree centrality, the betweenness centrality and the closeness centrality.

**Definition 29. Degree Centrality**: The *Degree centrality* considers nodes with the highest degrees (number of adjacent edges) as the most central.

[Shaw 1954] introduced the idea to measure point centrality with its degree. It highlights the local popularity of the network, actors that influence their neighbourhood and ones who are highly visible in their community. In directed graphs the *out-degree* and *in-degree* are alternative definitions that take into account the direction of edges, representing respectively the influence and the support of the actor [Nieminem 1973]. In a directed graph, the outgoing and ingoing edges of a node respectively have this node as source and end. Consequently the out-degree and the in-degree of a node are respectively the number of its outgoing and ingoing edges. The *n-degree* is an alternative definition that widens the neighbourhood considered to a distance of *n* or less [Garrison 1960] [Pitts 1965]. The distance between two actors is the minimum number of relationships that link them. In respect with this definition the *n*-degree of an actor is the number of others actors he is linked to by a sequence of *n* relationships or less. However the *n*-degree is rarely used with *n* higher than 2 as it has been shown that we cannot see, nor influence more distant network [Burt 1992]. The *reach* is equivalent to a 2-degree (3-degree in rare cases) and represents the network that an actor can see and/or influence. For instance, in the social network of Zachary karate club, the actors 1, 33 and 34 have degrees well above the rest of the network and are the most central both in terms of degree (Figure 12).

**Definition 30. Betweenness Centrality**: The *betweenness centrality* considers nodes that are more often on shortest path between other nodes as the most central.

[Bavelas 1948] introduced the idea that actors are more central when they are located on communication paths of the social network. The *betweenness centrality* focuses on the ability of an actor to be an intermediary between any two other actors in the network. An actor located on a geodesic path has a strategic position in the cohesion of a network and the flow of information, especially if this path is unique. For instance, an agent located on the only road connecting two groups of actors has a strong control on the communication between these groups. Consequently, a network is highly dependent on actors with high betweenness centrality and these actors have a strategic advantage due to their position as intermediaries and brokers [Shimbel 1953] [Cohn & Marriott 1958] [Burt 1992] [Holme et al 2002] [Burt 2004]. These actors have the power to choose to leverage or to lower the communication between groups and have a privileged access to information of each group [Burt 2004]. *The more intermediate a node is, the more strategic its position is in the network.* In a directed graph, the interpretation of the betweenness centrality is still the same but we only considered path without any change of direction. For instance, in the social network of Zachary karate club, the actors 3, 9, 14, 20, 31 and 32 are the most central in terms of betweenness, their absence or the failure of their links with one of the clubs would weaken the cohesion of the network (see Figure 12).

**Definition 31. Closeness Centrality**: The *closeness centrality* considers as most central the nodes that have the smallest average length of the roads (sequence of relationships) linking an actor to others.

[Leavitt 1951] considers the centrality as a measure of closeness in the social graph. The *closeness centrality* reveals the ability of a node to quickly connect with all the other actors of the network. In directed graphs, the interpretation of the closeness centrality is modified when we consider the direction of edges. The closeness centrality computed on outgoing and ingoing edges respectively represents the capacity of an actor to reach or to be reached in the whole network. For instance, in the social network of Zachary karate club, the actors 1, 33 and 34 are closer to other nodes in the network due to their high degrees and are the most central both in terms of degree and closeness.

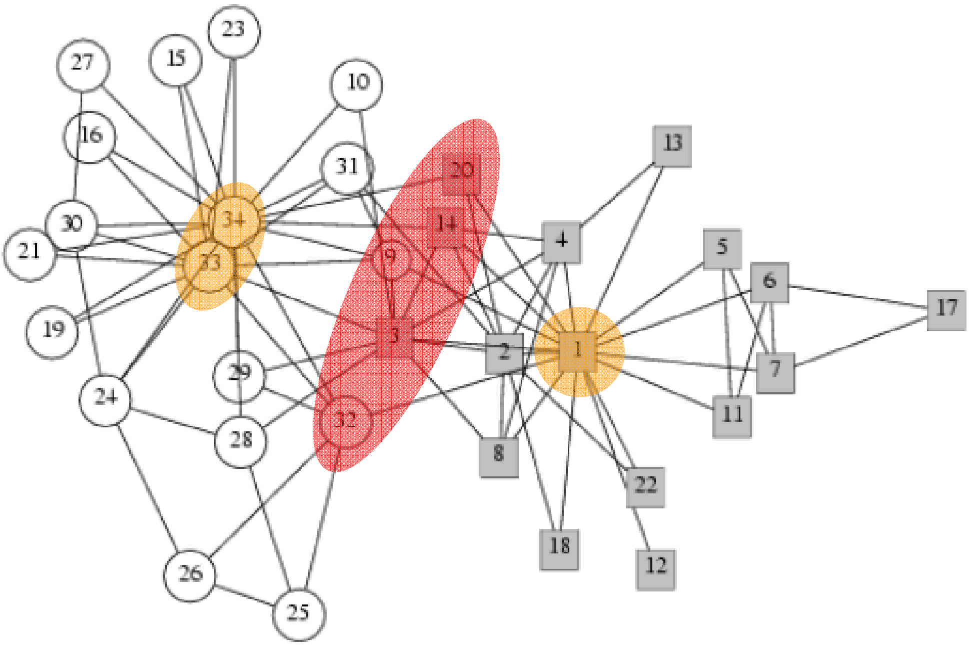


**Figure 12.** Important actors in the Zachary karate club. Nodes 1, 33 and 34 have the highest degree and closeness centralities and nodes 3, 9 14, 20 and 32 have the highest betweenness centralities.

The three measures of centrality discussed above are nuanced if one takes into account the orientation of the arcs. In all these definitions of centrality, we nuanced their interpretation when considering the direction of relationships. This highlights the semantic embedded only in the direction of relationships when building the network. For example, in order to analyze the propagation of information in a network, the direction of relationships is essential to convey information from Peter to Jack, only paths from Peter to Jack with no change of direction have to be considered. Generally, in directed graphs, an incoming arc is considered as a support for the target node while an outgoing arc represents an influence from that node.

In a social network, the direction of relationships contains a lot of semantics. Taking into account the direction of relations leads to the notion of *prestige* that differentiate incoming and outgoing edges to characterize the position of actors in respect with their neighbours. An incoming edge is considered as a support for the target node while an outgoing edge represents an influence from that node. The three measures of centrality discussed above are qualified if one takes into account the direction of the edges.

[Scott 2000] discusses an interesting approach, arguing that *the centrality measure of a node should take into account the centrality of its adjacent nodes*. Indeed, a node close to another node having a higher centrality enjoys some of the advantages offered by this position. Thus, the centrality of a node should consider the centralities of each of its neighbours. For instance, the degree centrality should be the sum of its degree and a fraction if its neighbours degree.

Other approaches have focused on the *egocentric centrality*:

**Definition 32. Egocentric Centrality:** centrality of a node on the sub network of its neighbourhood

This type of centrality measures the influence and the position of an actor in its adjacent social network. This approach is considered by [Everett & Borgatti 2005] demonstrating a correlation between the centrality and the ego-centrality of a node.

In relation with the betweenness centrality, [Burt 1992] introduced the concept of *structural hole* that defines a separation or a weak linkage between two groups of redundant contacts. Contacts are redundant when they belong to the same sub-group of dense contacts and when they share several connections. The structural holes offer two majors advantages to those controlling them. Firstly, the people who control the structural holes have a strategic networking advantage and can make the most of their position as intermediary in the communication and the information flows. Then structural holes provide an informational benefit, allowing quick access to non redundant information. The closest contacts of structural holes are better informed and faster. In [Burt 2004], the author shows that people close to the structural holes are more likely to have "good ideas", with the informational benefits brought by structural holes. *In a redundant group of contacts, information is usually shared with the majority, and new information in a coherent group generally comes from outside. Structural holes are the communication channels for this information.*

### 3.2.2 Global Metrics and Network Structure

Metrics help understanding the global structure of the network which enables us to:

- <u>evaluate</u> the effectiveness of the network at communicating.
- <u>estimate</u> the resilience of the network to connection failures.
- <u>anticipate</u> its evolutions.
- <u>understand</u> repartition of people and activities.

**The density indicates the cohesion of the network.**

**Definition 33. Density**: number of edges of the network expressed as a proportion of the maximum possible number of edges: *n*(n-1)*, with *n* the number of nodes.

According to [Scott 2000], this measure can be used in the context of (1) an egocentric or (2) a socio-centric analysis.

- *An ego-centric analysis* measures the density of links around a node. Such an analysis shows the influence of the analyzed node on the density subgraph it belongs to with its neighbours.
- *A socio-centric analysis* computes the density on the whole graph and measure the constraint on the network on its members.

The calculation of density is relative to the maximum number of edges a graph can contain. However, this maximum number is itself a function of the size of graph, and any comparison between densities of graph of different sizes does not provide any significant result. [Scott 2000] proposes an interesting approach in calculating the maximum number of connections in a social network. Indeed, the management of social relationships is time-consuming, and time limits the number of contacts a person can maintain, implying that the bigger is a social network the lower is the density. The Dunbar limit argues on the cognitive cost inherent in the maintenance of social relationships and proposes 150 as the average number of relationships one can maintain [Dunbar, 1998]. The density also varies depending on the types of relationships in a social network. A love network is much less dense than a professional network due the characteristics of ties (e.g. exclusive relationship, more or less time consuming maintenance, etc.). So the type of relationships in a social network would parameterize the density (e.g. considering the sub graph of an exclusive relationships, the density could be maximum when every nodes has a connection).

**The global centralization of the network enables us to estimate the dependency of a network structure on its members.** Freeman explains how to assess the centralized nature of the structure of a social network [Freeman 1979]. The overall centrality, or centralization, of a social network is calculated from local centralities of nodes. The calculation of centralization depends on the definition of local centrality that we consider, whether one considers the centrality as control, independence or activity. If we consider local degree centralities, the overall centrality highlights the existence of points of high interest within a social network, namely an activity concentrated around certain actors. A measure of the global centralization based on local betweenness centralities, provides an indication of the dependence of the network connectivity and efficiency over its actors. Finally a global centrality based on local proximity centralities measures the performance of the communication network, including traffic information.

**The network resilience to the withdrawal of nodes or edges enables us to evaluate the strength of the network connectivity in case of failure**, such an actor leaving the network or the brokerage of relationships. [Newman 2003a] gives us an overview on network resilience. We have seen that the extent of centralization of a network shows the dependence of a network over its vertices. This dependence can also be measured by the impact on connectivity of the network of withdrawing nodes or edges. Indeed, the removal of a strategic node or edge, such as nodes with high betweenness centrality, may increase the length of the shortest path between many other nodes or even split the network into unconnected sub networks. We can test the resilience of a network with two types of removals: random or targeted. In general, the structures of social networks are quite resistant to random withdrawals of vertices or edges, while targeted withdrawals can seriously affect these structures. For example, the removal of a bridge between two groups of strongly connected vertices reduces or cuts off communication between these two groups. [Holme et al 2002] points to possible strategies for attacking networks focused on strategic nodes and the authors extend these strategies to attacks based on the edges. The main targeted strategies consist in iteratively withdrawing the most central nodes and edges for a given definition of the centrality.

**Community detection helps understanding the distribution of actors and activities** in the network by detecting groups of densely connected actors [Scott 2000]. The *community structure influences the way information is shared and the way actors behave* [Burt 1992] [Burt 2001] [Burt 2004] [Coleman 1988]. The concept of *network closure* argues that in a community the density of connections enable its members to be aware and have access to the information held by each [Coleman 1988]. Moreover, the redundancy of contacts enables information to quickly spread and facilitates both the penalty and confidence in a community [Burt 2001]. Penalties may include isolation in the network and loss of confidence, easing the decision of trusting someone and increasing the information quality. In an educational or business context, social network analysis helps forming productive working groups and helps improving the efficiency of communication channels.

**[Scott 2000] proposes three main graph patterns to detect cohesive subgroups of actors** that play an important role in community detection:

**Definition 34. Component**: A *component* is an isolated connected sub graph.

**Definition 35. Clique**: A *Clique* is a complete sub graph.

**Definition 36. Cycle**: A *Cycle* is a path returning to its point of departure.

Alternative definitions extend these initial concepts that are too restrictive for social networks or do not handle important network characteristics.

**Definition 37. Strong Component**: A *strong component* is a component whose paths that connect nodes do not contain change of direction while a weak component ignores direction changes.

**Definition 38. Strong cycle**: A *strong cycle* is a path that does not contain any change of direction while a *weak cycle* ignores direction changes.

**Definition 39. k-plex**: A *k-plex* is a component in which every node is connected to all the other nodes of the *k-plex* except a maximum number of *k* nodes.

**Definition 40. n-clique**: An *n-clique* is a component in which every node has a maximum distance of *n* to any other members of the *n-clique*.

The paths connecting the vertices of an *n*-clique may contain vertices excluded from this clique. For instance, in Figure 13 Gérard helps connecting the 2-clique, which is composed of the other nodes, but is not contained in it because he has a distance of 3 to Pierre. This last case is excluded by the *n-clan* that restricts the definition of *n*-clique as follow:

**Definition 41. n-clan**: An *n-clan* it is a set of nodes all connected by paths of maximum length *n* that forming a sub graph with a diameter less than or equal to *n*.

**Definition 42. LS-SET**: An *LS-SET* is subset of vertices S such that any proper subset of S (subset of S different from S) has more links to its complement in S than to the outside of S.

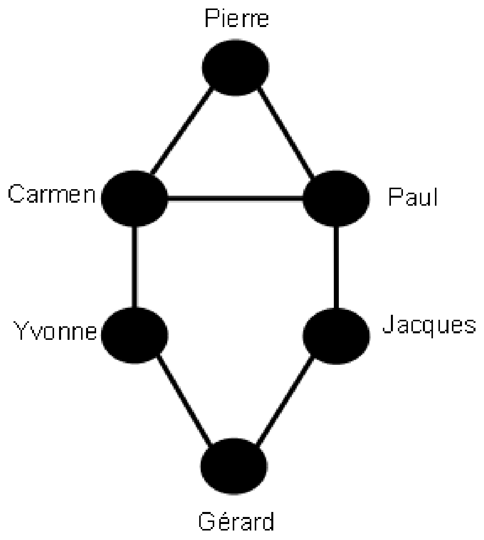


**Figure 13.** Pierre, Paul, Jacques, Carmen and Yvonne form a 2-clique but a 3-clan as the only geodesic between Yvonne and Jacques has a length of 2 but go through Gérard.

**Definition 43. Triad**: A triad is a cycle of length 3, so named as it connects three different nodes.

Triads are frequent patterns in social networks that have a strong tendency to clustering, namely two nodes connected to one same node have a high probability of being linked. This tendency to clustering is measured by the clustering coefficient that is the ratio of triads on the maximum number of possible triads for in a given network.

**Definition 44. Clustering Coefficient**: Let |*TRIAD*| the number of triads, and |*2_PATH*| the number of paths of length 2 in the network, the clustering coefficient is:

$$\frac{3 \times |TRIAD|}{|2\_PATH|}$$

**Definition 45. Node Clustering Coefficient**: Let |*TRIADi*| the number of triads, and |*2_PATHi*| the number of paths of length 2 having the node $v_i$, the clustering coefficient of $v_i$ is:

$$CCi = \frac{|TRIADi|}{|2\_PATHi|}$$

We can alternatively compute the clustering coefficient of the network with the local values of each node:

$$\frac{1}{n}\sum_i CCi$$

**Social networks generally highlight structural properties that are present in most complex networks [Newman 2003a].** The most popular one is the *small world property*, which was first highlighted by the famous experiment of Milgram in 1967

[Milgram 1967]. Every actor in a social network is connected to any other actor by a short path of lengths, initially observed, as an average of 6. In fact the shortest path between two vertices in a social network of size n tends to be of an order of *log(n)*. Thus, when the network size increases, the length of the shortest paths also increases but slightly. Moreover members of social networks have the ability to easily find these shortest paths [Newman 2003a]. Another characteristic is derived from the human tendency to socialize in a group that provides social networking with a *strong tendency to clustering and community structure*, due to an important transitivity of social relationships. If Peter and Jack are both connected to Paul, then Peter and Jack have a high probability to be connected too. Such transitivity produces a community structure, namely densely connected groups of nodes that are connected by bridges. Then this connection in the network frequently follows a *tendency to affiliation* between nodes that have similar properties (e.g. in [Moody 2001] people tend to connect with other people of the same colour and same age); this property of social networks is named *assortative mixing* [Newman 2003a]. Finally *the degree distribution follows the classical power law*, namely that few nodes have the highest degrees. Figure 14 shows the degree distribution of the social network of the Zachary karate club of Figure 3 and Figure 12.

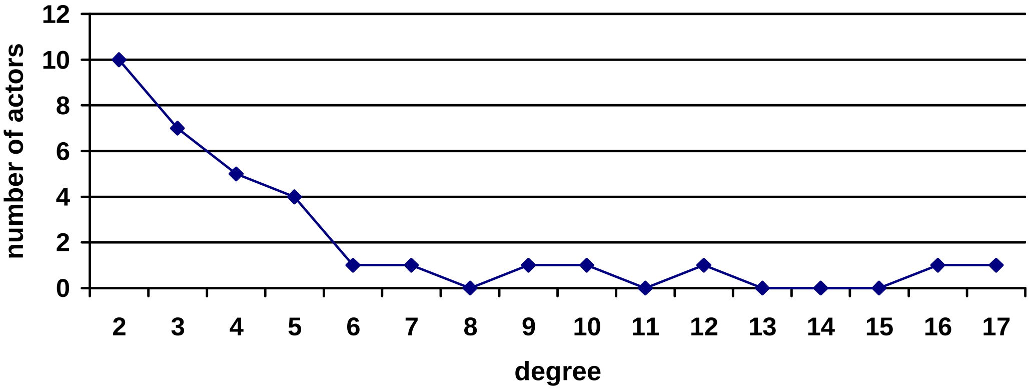


**Figure 14.** Degree distribution of the Zachary karate club social network.

The definitions presented in this part to define community structure are still too theoretical and restrictive, and do not reflect the communities' characteristics of real social networks. Moreover, algorithms to detect some patterns are exponential problems, such as detecting all cliques in a graph [Bron & Kerbosch 1973]. For example, in the social network of Zachary karate club, we can clearly distinguish two groups in a visual way, and none has strictly the properties listed above. Consequently, broader notions have been taken into account to detect communities in social networks. We will now discuss these concepts in the next part that presents different community detection algorithms.

### 3.2.3 Community Detection Algorithms

Community detection algorithms give an overview of a social network and global view of the distribution of actors and activities. We can classify these algorithms in two categories: hierarchical algorithms and heuristic based.

#### 3.2.3.1 Hierarchical Algorithms

First, there are hierarchical algorithms that build a hierarchical tree of communities, called a dendrogram, namely a tree of denser and denser communities from top to bottom. The Figure 15 proposes an example of dendrogram. These algorithms start by assigning weights to each pair of nodes or edges. These weights represent the connectivity of the corresponding pairs of nodes in the network. Then they build a tree whose nodes are groups of vertices more or less similar. The deepest communities of the tree represent more densely connected groups of nodes. Thus, the more you go up in the tree the larger the communities are, with the root representing the whole network. Two main strategies for building these trees are used, grouping hierarchical algorithms into two categories: (1) agglomerative algorithms and (2) divisive algorithms. They differ in the construction of the tree and in the logic of allocating importance to the edges. Agglomerative algorithms start from the leaves of the tree, and group nodes in larger and larger communities, while divisive algorithms start from the root of the tree, and group nodes in denser and denser communities.

- <u>Agglomerative Algorithms</u>

In these algorithms, there are three main criteria for allocating weights to pairs of nodes. The first criterion is the number of paths that go through these nodes. The other two criteria are variants of the first criterion; one restrains the number of paths to paths having no nodes in common and the other to paths having no edges in common. Once these weights are assigned, they iteratively include nodes in the tree by considering the weights in descending order, until you have considered all the weights. The main drawback of these algorithms is that they exclude in most cases the peripheral members, which are more isolated from their community.

[Donetti & Munoz 2004] use the eigenvectors of the Laplace matrix of the graph to measure similarities between nodes, their algorithm runs in time *O(n3)*.

The Netwalker algorithm of [Zhou & Lipowsky 2004] is "based on the average time for reaching a summit by random walks" to measure the similarity between nodes. Its time complexity is $O(n^3)$.

The SCAN algorithm [Xu et al 2007] offers to find overlapping communities. It is based on the basic idea that the community structure of a node is defined by its neighbours; this algorithm forms communities by establishing a minimum score of structural similarity between a node and its neighbours. The structural similarity between two nodes is based on the number of neighbours they share.

Newman offers an fast algorithm [Newman 2004a] for detecting communities in large networks with a complexity of $O(n.log^2(n))$. This algorithm provides a cut of the graph by optimizing a modularity function:

$$Q = \sum_j (e_{ij} - a_i^2)$$

with $e_{ij}$ the fraction of edges of the network that link nodes belonging to same communities in the network, $a_i$ is defined by:

$$a_i = \sum_j e_{ij}$$

In other words, the modularity measures the fraction of edges within communities in the network minus the expected value of the same quantity in a network with the same community partition but with random connections between nodes (the randomization of connections preserves the degree of the nodes).

**Definition 46. Modularity**: let $m$ be the number of edges of the network, $d_{<i>}$ the degree of vertex $i$, $A_{ij}$ the number of edges between $i$ and $j$, $c_i$ the community of $i$, δ(ci,cj) = 1 if $c_i = c_j$, 0 otherwise, the modularity is:

$$Q = \frac{1}{m} \sum_{i,j \in V} [A_{ij} - \frac{d_{<i>} d_{<j>}}{m}] \delta(c_i, c_j)$$

The higher the modularity is, the better the community partition is. When a partitioned network has a high modularity, it means that there are more connections between nodes within each community than between nodes from different communities. Many variants were proposed to measure the modularity and to take into account different graphs characteristics. The modularity is generalized for directed graphs in [ Leicht & Newman 2008], proposing an alternative approach to this community detection algorithm:

**Definition 47. Directed Modularity**: let $m$ be the number of edges of the network, $d^{in}_{<i>}$ and $d^{out}_{<i>}$ the in-degree and out-degree of vertex $i$, $A_{ij}$ the number of edges between $i$ and $j$, $c_i$ the community of $i$, δ(ci,cj) = 1 if $c_i = c_j$, 0 otherwise, the directed modularity is:

$$Q = \frac{1}{m} \sum_{i,j \in V} [A_{ij} - \frac{d^{out}_{<i>} d^{in}_{<j>}}{m}] \delta(c_i, c_j)$$

[Djidev 2008] reduces the problem of modularity optimization to the weighted min-cut problem and proposes an algorithm that runs in time $O(n.log(n) + m)$ for weighted graphs. [Barber 2007] proposes a definition of modularity for bipartite graphs. Finally [Chen et al 2009] proposed a variant that maximizes the edges within community and minimizes the outgoing edges of communities.

- Divisive Algorithms

These algorithms build the tree in reverse. They assign a weight to each edge to set a divisive criterion between nodes. The tree is built from the whole graph, iteratively removing edges by descending weight.

The most popular of these algorithms is that of [Girvan & Newman 2002] which sets the weights of the edges according to their capacity to be on geodesic paths between any two nodes of the network. The "most intermediary" nodes are removed first and so on. This technique provides very good cuts of a network and is adapted to the community structure of social networks. However, this algorithm requires the calculation of betweenness centralities and has consequently a high time complexity *$O(m^2.n)$*. It is therefore usable only on small networks. The Figure 15 presents the dendrogram of the Zachary karate club network computed with this algorithm. The Figure 16 proposes a visualization of the community partition obtained by applying this algorithm on the collaboration network of scientists in a university. Different variants of this algorithm have been proposed. In particular, [Fortunato et al 2004] optimized the quality of the resulting partition but with a higher time complexity, *$O(m^3.n)$*, and [Bothorel & Bouklit 2008] adapted this algorithm for hypergraphs.

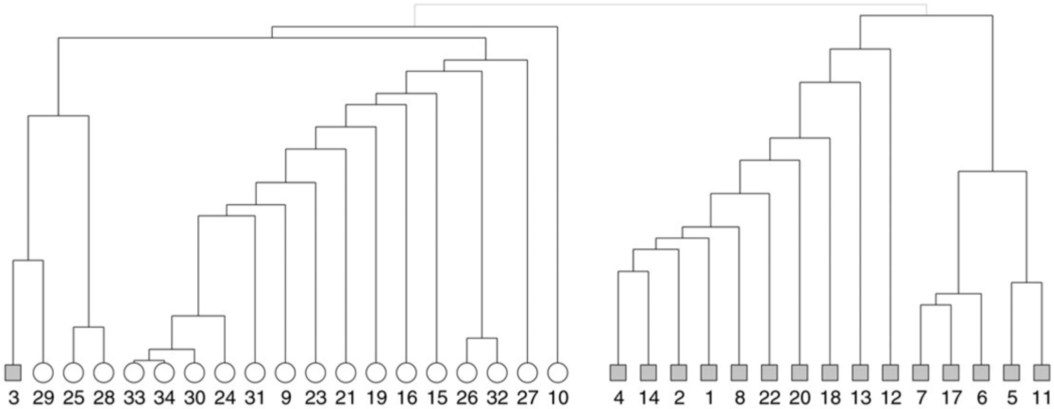


**Figure 15.** Hierarchical tree of the Zachary karate club network computed with the community detection algorithm of [Girvan & Newman 2002]. "The initial split of the network into two groups is in agreement with the actual factions observed by Zachary, with the exception that node 3 is misclassified."

[Radicchi et al 2004] extend the concept of clustering coefficient to edges and propose an algorithm that iteratively removes the edges with the lowest coefficient. The clustering coefficient of an edge is the fraction of the number of cycles of a given length that go through this edge divided by the number of possible cycles of the same length that could have gone through it, depending on the degree of the extremities.

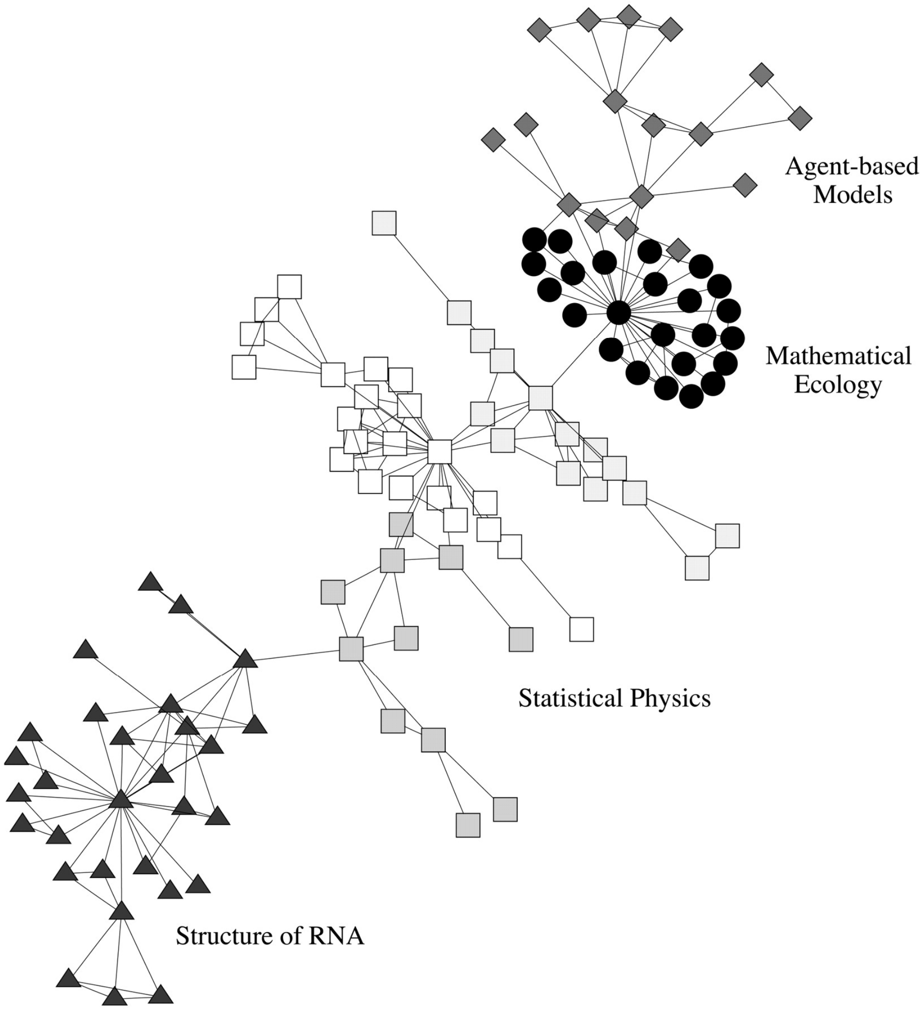


**Figure 16.** Community partition of the "the largest component of the Santa Fe Institute collaboration network" computed with the algorithm of [Girvan & Newman 2002]. Each shape represents a community.

#### 3.2.3.2 Heuristic Based Algorithms

Several non-hierarchical algorithms have also been proposed, they are based on heuristics related to the community structure of networks and to community characteristics.

[Wu 2004] focus on similarities between a social network and an electrical network and provides an algorithm based on the simulation of electricity circulation. This method provides a result in linear time in practice but imposes a major constraint: fixing the number of resulting communities in advance, which is not possible in most cases.

Several other algorithms are based on random walks in a graph. These algorithms are based on the assumption that a random walk in a graph tends to end up "trapped" in parts of the graph corresponding to highly connected communities. One of the most popular among these algorithms is the Markov Cluster Algorithm that simulates flows in a graph [Dongen 2000] with successive matrix operations. The Figure 17 presents the different groups of nodes that are detected and isolated along the different loops of the matrix operation process. Among community detection algorithms that are based on random walk, the algorithm of [Pons et al 2005] has the most efficient time complexity in practice, *$O(n^2.log(n))$*, but it is the most expensive in space *$O(n^2)$*. An overview of random walk based algorithm is proposed in [Pons et al 2005].

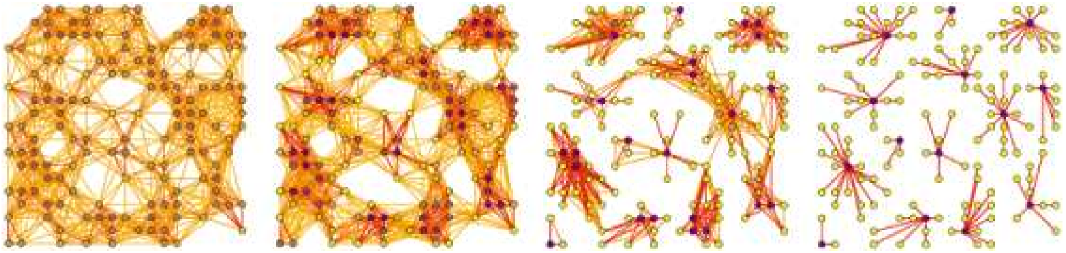


**Figure 17.** Detection and isolation of densely connected groups of nodes by random walks with the Markov Cluster Algorithm [Dongen 2000].

The label propagation algorithm of [Raghavan et al 2007] is the most efficient algorithm in practice, but its ending is not deterministic (however in practice it always ends). Every node is given an initial random unique label representing the community to which it belongs. At each step each node changes its label by taking the most used one in its neighbourhood. This iterative process leads in practice to a consensus with a unique label for each community. [Gregory 2009] proposes a variant of this algorithm to detect overlapping communities.

The Figure 18 summarizes the performances and the characteristics of the algorithms mentioned above. Other algorithms are also described in [Danon 2005] [Newman 2004b] [Girvan & Newman 2004] and [Leskovec et al 2010].

| Algorithm category | Reference | Time complexity | Graph size | Graph type |
|---|---|---|---|---|
| Hierarchical Agglomerative | [Donetti & Munoz 2004] | $O(n^3)$ | $10^3$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Zhou & Lipowsky 2004] | $O(n^3)$ | $10^4$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Newman 2004a] | $O(n.log^2(n))$ | $10^5$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Newman 2008] | $O(n.log^2(n))$ | $10^5$ nodes | Unlabelled<br>Directed<br>Not weighted |
| Hierarchical Divisive | [Girvan et Newman 2002] | $O(m^2.n)$ for not weighted graph<br>$O(m^2.n.log(n))$ for weighted graphs | $10^4$ nodes | Unlabelled<br>Undirected<br>Weighted |
| | [Radicchi et al 2004] | $O(n^2)$ | $10^4$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| Heuristic based | [Djidev 2008] | $O(n.log(n)+m)$ | $10^5$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Wu 2004] | $O(n+m)$ | $10^5$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Pons et al 2005] | $O(m.n^2)$ theorical<br>$O(n^2.log(n))$ in practice | $10^4$ nodes | Unlabelled<br>Undirected<br>Not weighted |
| | [Raghavan et al 2007] | $O(n)$ *non determinitic* | $10^6$ nodes | Unlabelled<br>Undirected<br>Not weighted |

**Figure 18.** Categories and performances of community detection algorithms.

#### 3.2.3.3 Evaluating a Community Partition

[Bolshakova & Azuaje 2003] propose three indices to evaluate the quality of a graph community partition. The Silhouette index compares the silhouette of the obtained clusters with the whole network silhouette, namely it compares characteristics such as the diameter to evaluate the heterogeneity of the obtained communities. The Dunn index and the Davies-Bouldin index, compare the intra-cluster distance and the inter-cluster distance to determine the quality of obtained community partition.

In [Girvan & Newman 2004], an alternative approach is proposed: the calculation of modularity. Its value ranges between 0 and 1. The closer the value is to 1, the better the partition. The modularity is currently the baseline measurement for assessing the quality of a cut into communities. In [Gustafsson et al 2006], a comparison is made between the modularity and the Silhouette index and modularity is found to be more relevant.

[Rattigan 2007] proposes two complementary indices to measure the quality of a community partition. These two indices are (1) the proportion of inter-edges and (2) the proportion of intra-community edges. Values are both measured between 0 and 1. A good community partition has a low rate of intercommunity edges and a high rate of intra-community edges.

Finally [Leskovec et al 2010] compares the different properties of the output communities computed by community detection algorithms, when applied to different networks. Each of these algorithms approximates different objective functions for finding groups of nodes with more and/or better interactions amongst their members than between their members and the rest of the network. Thus, the output communities highlight different characteristics of each algorithm, depending on the input network. Consequently, when selecting a community detection algorithm, one should take care of the characteristics of both the analyzed network and the algorithms' output communities in respect with its objectives and performance constraints.

#### 3.2.3.4 Partial Conclusion

Most of the community detection algorithms, consider only unlabeled and undirected graphs, while they rarely provide non-overlapping clusters. None of these algorithms works on directed, labelled graph and proposes overlapping communities. Ignoring the orientation of edges leads to some loss of properties of community formation, in the same way that the interpretation of centralities is different when considering edge direction. Typing, or labelling, of links in a social network also brings a lot of semantics, as well as the typing of nodes that helps us describe a social network with different types of actors. In addition, when one actor may belong to different communities with different membership involvement, most of these algorithms will only make that actor a member of a single community.

### 3.2.4 Algorithms for Computing Centralities

The centrality measure is useful for detecting strategic positions in a social network. We saw in section 3.2.1 that there are different kinds of centralities (degree, betweenness, closeness). In this section we will review existing methods. We will also focus on algorithms that compute the betweenness centrality, which is the trickiest to compute efficiently.

#### 3.2.4.1 Formulas and Principles

[Freeman 1979] suggests two methods of calculation for each of the three measures of local centrality (degree, betweenness, closeness). He presents a measure that is function of the network size and another one that does not take into account these parameters. The former is used to measure the influence of the activity of a node in a network, while the latter is used to compare centralities between different networks.

**Definition 48. Degree Centrality:** The degree of centrality of a node is simply its degree [Nieminem 1974].

The method for calculating the betweenness centrality of a node consists in adding the values of partial betweenness of that node for each pair of other nodes in the network. The partial betweenness of a node *b* for a couple of node *x* and *y* is the ratio of the number of geodesic paths between *x* and *y* containing *b* divided by the total number of geodesic paths between *x* and *y*, namely the probability of *b* to be on a shortest path between *x* and *y* [Freeman 1977].

**Definition 49. Partial Betweenness:** Let $nb^g(b,x,y)$ the number of geodesics between *x* and *y* going through *b* and $nb^g(x,y)$ the total number of geodesics between *x* and *y*, then the partial betweeness of *b* for *x* and *y* is:

$$B(b,x,y)=\frac{nb^g(b,x,y)}{nb^g(x,y)}$$

**Definition 50. Betweenness Centrality:** Let B(b, x, y) the partial betweenness of a node *b* for a couple of node *x* and *y*, then the betweenness centrality of a node *b* is:

$$C^B(b)=\sum_{x,y\in E_G} B(b,x,y)$$

For instance, in the Figure 19, the node *B* is located on one of the two shortest paths going from *A* to *D* and has consequently a probability of 0.5 to act as an intermediary between these two nodes. Thus the partial betweenness of *B* for the nodes *A* and *D* is 0.5. However, *B* is on the only shortest paths going from *A* to *E*, so it has a partial betweenness of 1 for this pair of nodes. *B* is not located on any others shortest paths between other pairs of nodes so its betweenness centrality in this network is 1.5.

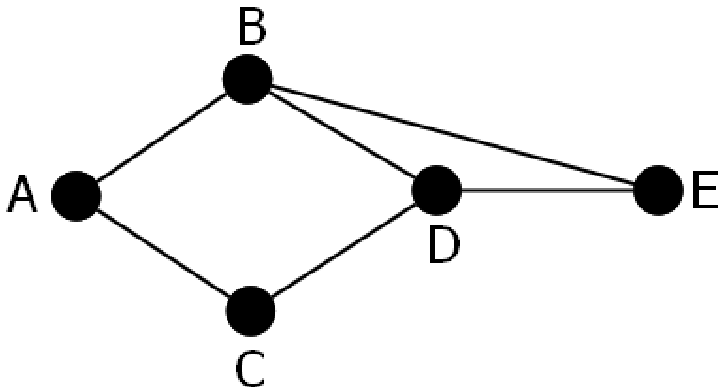


**Figure 19.** In this toy example, *B* has the highest betweenness centrality with a value of 1.5.

The closeness centrality of a node is measured as the inverse of the sum of distances from this node to all the other nodes [Freeman 1979].

**Definition 51. Closeness Centrality:** Let *g(k,x)* be a shortest path between *k* and *x*, and *length(g(k,x))* the length of such a shortest path, the closeness centrality is:

$$_{\mathrm{S}}C^{c}(k)=\left[\sum_{x\in E_G} length(g(k,x))\right]^{-1}$$

To make these measurements independent from the network size, Freeman suggested in both cases to divide the result by the maximum possible value of the centrality in a network of the same size. The maximum value is always reached by the most central node in a star network, *i.e.* a network with one node connected to all the other nodes. So for a network of size *n*, the maximum value of the degree is *n-1*, the maximum value of betweenness is *$(n^2$ - 3n + 2) / 2*, and the maximum value of the closeness centrality is *1/(n-1)* (the minimum sum of distances is *n-1*, for a point that is adjacent to everyone).

Finally Freeman provides a formula for calculating the global centralization of a network for each of the 3 measures of local centralities. The principle is to measure the difference between the value of the highest centrality and the centralities of the other nodes in the graph.

While the calculation of the degree centrality is obviously trivial, computing the betweenness and closeness centralities requires computing all the geodesic paths, which has an important time complexity of *O(m.n)*. However the small world property of social networks enables to consider the degree as a good estimator of the closeness centrality of a node; nodes with higher degrees are closer to other nodes in the network, thanks to their good connectivity.

In social network analysis the betweenness centrality is one of the most significant measures as it highlights highly strategic positions, both for the information channels and the network resilience.

We will now review the different algorithms that were proposed to compute the betweenness centralities.

### 3.2.4.2 Algorithms for Computing the Betweenness Centrality

Different types of algorithms have been proposed to compute the betweenness centrality:

- exact algorithms compute the formal definition of the betweenness centrality
- sampling algorithms propose to estimate the betweenness centrality
- parallel algorithms have been proposed to deal with very large graphs

**Exact algorithms**

Several algorithms that compute the exact betweenness centrality have been proposed. They are applicable on small networks, with an order of $10^5$ nodes for the most efficient ones. These algorithms usually offer a version for weighted and not weighted graphs. The main ones only consider geodesic paths between nodes [Brandes 2001] [Newman 2001]. The others are based on an optimal distribution of information flow in the network between the different possible paths [Freeman & Borgatti 1991], or combine both approaches [Latora & Marchiori 2004]. The most efficient exact algorithm is described in [Brandes 2001], it provides a result with a space complexity of $O(n+m)$ and a time complexity of $O(n.m)$ and $O(n.m+log^2(n))$, respectively for not weighted and weighted graphs. This algorithm is based on a set of lemmas that consider only the necessary computations and reduce the complexity of the optimal methods that computes the geodesics. For example, if $v_s$ is on a geodesic from $v_r$ to $v_t$, then $dist_{rt} <= dist_{rs} = dist_{st}$. [White & Borgatti 1994] takes also into account the orientation of the arcs.

[Brandes 2008], performs an overview of the alternatives proposed for the computation of the betweenness centrality. These variants include the importance of the position of nodes on geodesics, the importance of the path lengths, edge betweenness (namely edges located on geodesics), betweenness for group of nodes and betweenness in multipartite networks. He adapted the algorithm of [Brandes 2001] for each of these alternatives.

The computation of betweenness centrality in multipartite network is poorly treated in the literature. We note especially [Flom et al 2004] and [Brandes 2008], who treat betweenness centrality in networks with nodes of two different types, namely bipartite graphs. [Everett and Borgati 1999] adapt the core concepts of node betweenness centrality to group of nodes. Nodes that share a given attribute, such as sex or age, are considered as members of the same group. However, the type of the nodes could be considered as an attribute, offering the possibility to use their approach as a solution for computing the betweenness centrality in multipartite graph in general.

[Everett & Borgatti 2005] provides a method for calculating the betweenness centrality in an egocentric network, namely betweenness of a given node over the network formed by its adjacent nodes. This measure is used to extract the most influent actors in the neighbourhood of a given node.

Finally [Bothorel & Bouklit 2008] propose an algorithm that adapts the definition of betweenness centrality to hypergraphs.

**Sampling and Heuristic based Algorithms**

Several other algorithms provide estimates of the betweenness centrality [Radicchi et al 2004] [Newman 2003b] [Brandes & Pitch 2007] [Bader et al 2007] [Geisberg et al 2008]. These algorithms give slightly less accurate results but with much better performance, making them usable for networks of around $10^6$ nodes. [Brandes & Pitch 2007] [Bader et al 2007] [Geisberg et al 2008] propose algorithms that compute the betweenness centrality from a sample of randomly distributed nodes in the network. The quality of these algorithms depends on the sampling technique; the more representative of the whole network is the sample the more accurate is the result. [Radicchi et al 2004] and [Newman 2003b] are based on heuristics that exploit the small world property of social networks.

**Parallel Algorithms**

Finally [Bader & Madduri 2006] and [Santos et al 2006] provided major contributions in terms of performance with two parallel algorithms for treating social networks of the order of $10^6$ nodes with exact results for the first one and with an estimation for the other. The algorithm of [Santos et al 2006] is also particularly interesting as it proposes an incremental approach, where the accuracy of the results improves as new data are processed. This algorithm is well suited for huge and for evolving graphs. The algorithm of [Bader & Madduri 2006] provides an accurate result by parallelizing the algorithm of [Brandes 2001].

The Table of the Figure 20 summarizes the performances and the characteristics of the algorithms mentioned above.

| Reference | Complexity | Graph size | characteristics | Graph type |
|---|---|---|---|---|
| [Newman 2001] | $O(n.m)$ and $O(n.m.log(n)$ respectively for not weighted and weighted graphs | $10^5$ nodes | Exact | Unlabelled<br>Undirected<br>weighted |
| [Brandes 2001] | $O(n.m)$ and $O(n.m + n^2.log(n))$ respectively for not weighted and weighted graphs | $10^5$ nodes | Exact | Unlabelled<br>Undirected<br>weighted |
| [Geisberger et al 2008] [Brandes et Pich 2007] | Like [Brandes 2001] but estimated with *k* nodes, *k* < *n*. | $10^6$ nodes | Incremental | Unlabelled<br>Undirected<br>weighted |
| [Bader & Madduri 2006] | $O(n.m)$ and $O(n.m + n^2.log(n))$ respectively for weighted and not weighted graphs | $10^6$ nodes | Parallel<br>Exact | Unlabelled<br>Undirected<br>weighted |
| [Santos et al 2006] | Not estimated | $10^5$ nodes | Incremental<br>Parallel | Unlabelled<br>Undirected<br>Weighted |

**Figure 20.** Types and performances of betweenness centrality algorithms.

### 3.2.5 Datasets

The quality and performance of the algorithms have been evaluated on several datasets. These datasets are artificially generated or based on real networks. Regarding the generation of networks, three main methods are used, (1) the random graphs of [Gilbert 1959], (2) the "preferential attachment" algorithm of [Barabasi & Albert 1999] and (3) the "small world" model of [Watts & Strogatz 1998]. The random graph generation produces networks without taking into account the properties of social networks. The "preferential attachment" model of [Barabasi & Albert 1999] proposes to generate a random graph with a power law distribution of the degrees. The model of [Watts and Strogatz 1998] reproduces the small world property of social networks. However these networks are automatically generated and are mostly used as witness dataset for comparing different community detection algorithms.

Several real datasets are regularly used to assess the effectiveness and quality of an algorithm for analyzing social networks. The earliest studied networks were built from manual surveys, for example by asking people to cite friends. The social network of the Zachary karate club has only thirty members but is often used as proof of proper functioning of a clustering algorithm. However, testing the time and space performances and limits of algorithms requires networks of large sizes to assess their performance, to judge their quality and observe their limits. Samples of the web graph can be easily extracted from the hyperlinks structure of web pages and offered the opportunity to get very large networks. The co-author and citation networks formed by scientific papers are also extensively used. CiteSeer is one the main sources for extracting such networks (http://citeseer.ist.psu.edu/).

However we will see in the section 3.3 that the web provides numerous social data that are now used as sources for all the domains that have interests in social network analysis.

### 3.2.6 Partial Conclusion

We discussed here the main methods and algorithms for analyzing social networks; In particular, we reviewed algorithms for computing community partition and betweenness centrality. Many community partition algorithms have been proposed, with different time and space performances and with different partition quality. On one hand algorithms with the best time complexity produce a low community partition quality or are just applicable in some cases. On the other hand algorithms producing good community partition are time and/or space consuming and can't be applicable on very large graphs (millions of nodes). However, while modularity based algorithms like [Newman 2004a] are favoured for not too large networks, more and more heuristic based algorithms are proposed and exploit smart heuristics to handle community properties with a lower cost [Raghavan et al 2007]. [Radicchi et al 2004] has opened the door for detecting communities with heuristic based betweenness centralities.

Approximating social network analysis measures, and graph sampling are probably the solution for dealing with very large networks. We started witnessing several advances in graph sampling for estimating SNA measures. [Rattigan et al 2006] proposes a method to index the graph structure and use this to highly improve computation of different measures such as shortest paths or betweenness centrality. These indexes are also used to optimize algorithms, including [Girvan & Newman, 2002]. Graph sampling for betweenness centralities have been deeply investigated by [Brandes and Pitch 2007] [Bader et al 2007] [Geisberg et al 2008], leading to new ways for estimating the betweenness centrality with an efficient computational cost on large graphs. Such an estimation of the betweenness centrality could be an interesting approach for reducing the computation time of community detection algorithms based on this centrality measure such as the one by [Girvan & Newman 2002]. [Maiya et al 2010] have proposed an innovative and efficient approach "to produce sub graph samples representative of the community structure". The authors show that community detection algorithms on such samples produce representative community partition of the original network and propose a method for affiliating the nodes of the whole network to the detected communities.

Finally some of the mentioned algorithms can be adapted to take into account the orientation, the weights and the labelling of edges, and also the attributes of nodes. [Brandes 2008] propose different version of the algorithm of [Brandes 2001] to separately take into account these graph characteristics. However none of the algorithms we presented considers all these graph properties as this would increases the complexity of the network analysis. Consequently the problem of analyzing a rich social network represented with directed typed weighted graph and handling very large graphs is still an open problem

We shall now see how the outburst of Web 2.0 and the emergence of Semantic Web technologies bring new perspectives, and open new problems for social network analysis, in particular in the detection and the construction of rich semantic social graphs.

## 3.3 Social Data on the Web

[Buffa, 2008] recalls the history of collaborative tools of the era preceding the arrival of the web as we know it today. The "liberalization" of the Internet in the late 80s has been quickly followed "by the creation of the web by Tim Berners Lee" in the early 90s. The synchronous and asynchronous means of communication offered by these technologies have been widely adopted, first by individuals and then by business organizations. Sociologists have soon highlighted a great interest in the quickly emerging social networks through these new means of communication, bigger and easier to reconstruct than by forms and questionnaires. The outburst of knowledge on the web motivated researches in web mining, a discipline aimed at exploring and extracting knowledge from web resources, including mining social networks. Internet has offered to breakdown geographical barriers and was quickly perceived as a boon for easing and stimulating collaboration. Since the mid-90s and the emergence of the first wiki[7] (see [Buffa 2008] for an history of the wikis), created by Ward Cunningham, social applications have proliferated on the Web and provide users with the opportunity to greatly improve their visibility and become important players in the landscape of the Web and its content.

The communication tools of the Web became progressively necessary as major modes of interaction in our society. Computer-mediated interactions between people on the internet and especially later on the World Wide Web reveal social network structures with properties that are close to those observed in the physical world [Wellman 2001]. Consequently, the interactions of internet users are sources of choice for extracting and analyzing social networks of very large sizes. Researchers have extracted social

[7] http://www.wiki.org,

networks from synchronous and asynchronous discussions (e.g., emails, mailing-list archives, IRC, instant messaging), the hyperlink structure of homepage citations, co-occurrence of names in web pages, and from the digital data created by Web 2.0 application usages. Turning the read web into a read/write web, the emergence of the Web 2.0, firstly mentioned by Tim O'Reilly in 2005 [O'Reilly 2005], has led to dramatic growth in the different possibilities for interacting and connecting, which started producing a huge amount of heterogeneous social data. Today the massive adoption of Web 2.0 collaborative tools gives the opportunity to study new networks with actors who always provide information not only about themselves but also about those with whom they interact.

In addition to [Wellman 2001], many results argue that online relationships form virtual social networks that are representative of real social networks. In fact, these virtual networks are created from interactions initiated by individuals. This argument is confirmed by [Mika 2007], but Mika stressed the incomplete nature of these social networks because of the online absence of some components of the offline world. However, [Hendler & Golbeck 2008] shows that the Web 2.0 and the Semantic Web amplify user connectivity and help online networks reproducing offline networks. Today, in 2010, more and more people are represented online and the sizes of many online social networks are now close to those of the populations of big countries. The growing mobile access to social networks on the web is also amplifying the convergence of offline networks and online networks.

This section deals initially with the application of social network analysis techniques on online social networks, then, it details the need for richer graph representations and for the exploitation of semantics in the analysis of a social network.

## 3.3.1 Social Network Analysis and Online Social Data

[Mika 2007] distinguishes three categories of social network sources on the web:

- Implicit social networks inferred with web mining techniques: links between personal homepages and co-occurrence of names in web pages.
- Online discussions: mails, chat, forum.
- The social applications of Web 2.0: publishing tools (wiki, blog, news), social networking and sharing sites (content, products, events, etc.) and collaborative games.

### 3.3.1.1 Web mining

[Adamic & Adar 2003] extracted the friendships networks of Stanford University and MIT, from the *hyperlink structure* of students' personal homepages. Students from these universities used to add, on their homepage, hyperlinks to the personal homepage of their friends. The authors showed that the graph formed by the hyperlink structure of these homepages highlights the same properties than offline social networks: "small world" graphs, power law distribution of the degrees in the graph, and a high clustering rate. Then, an index of similarity between the personal homepages is defined from the co-occurrence of text elements and the presence of hyperlinks between pages.

[Kautz et al 1997] [Mika 2005b] [Matsuo et al 2006] and [Jin et al 2007] extracted and analyzed social networks based on *co-occurrences of names on Web pages*. The principle of these methods is to measure the strength of a relationship between two persons based on the co-occurrences of their name in web pages. [Kautz et al 1997] and [Mika 2005b] use the Jaccard coefficient that is defined, for a pair of names *X* and *Y*, by ($n_{X\cap Y} / min(n_X,n_Y)$), with $n_X$ and $n_Y$ the number of pages containing the names *X* and $n_{X\cap Y}$ the number of pages containing both *X* and *Y*. [Matsuo et al 2006] and [Jin et al 2007] using the recovery coefficient which is defined with the same notation, as $n_{X\cap Y}/min(n_X,n_Y)$. The number of pages containing a name or a co-occurrence of names is obtained by querying a search engine, Altavista for [Kautz et al 1997] and Google for the others. These four approaches provide pretty similar methods for extracting social networks but for different purposes. [Kautz et al 1997] provide an exploration tool of an extracted social network for finding experts. [Mika 2005 bis] and [Matsuo et al 2006] apply the co-occurrence between names and terms in order to extract affiliation networks. [Mika 2005 bis] has used this method to build an affiliate network between the concepts manipulated by members of the Semantic Web community. [Matsuo et al 2006] provide a coordination tool for communities of researchers, POLYPHONET, which extracts and exploit affiliation networks. [Jin et al 2007] reapplies the techniques of [Matsuo et al 2006] for extracting online affiliation networks of artists and major Japanese firms.

#### 3.3.1.2 Synchronous and Asynchronous Discussions

[Tyler et al 2003] constructed an interaction of a company's employees from the analysis of email headers containing the sender and the recipient. Having demonstrated that this network has the social network characteristics, the authors identified sixty six communities of practice with the clustering algorithm of [Wilkinson and Huberman 2002]. The authors validate the obtained community partition with interviews of the members of seven communities, randomly chosen among the sixty six found communities.

[Leskovec & Horvitz 2008] analyzed a huge dataset of instant messaging communications extracted from the Microsoft Messenger system. They extracted, from this dataset, a social network of 180 million people, making it one of the largest ever analyzed. At the time of their experimentation, they observed that this social network has similar characteristics than offline social networks. First, they found an average distance of 6.6 between Microsoft Messenger users, which is close to the observed value in offline social networks. Then, the corresponding social graph is "well-connected and robust to node removal". Finally, they observed the classical tendency of people to bind with similar others, with more communication between people of similar age, language, and location.

### 3.3.1.3 Web 2.0

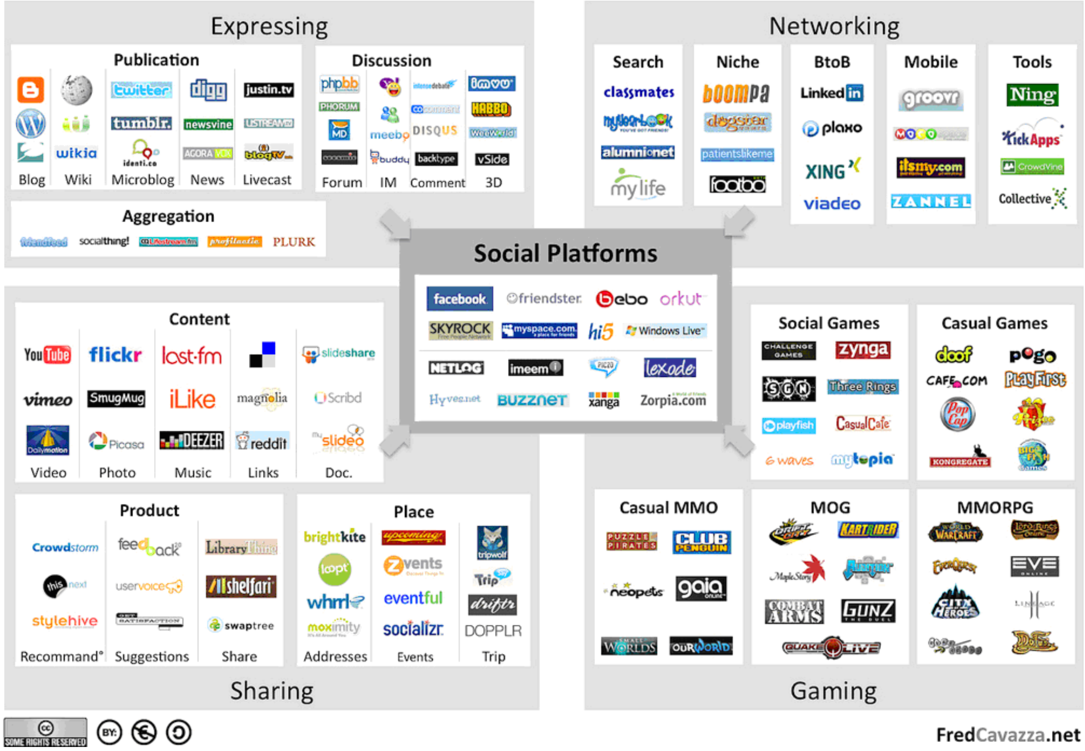


**Figure 21.** Social media Landscape proposed by Fred Cavazza in a blog post[8].

The Figure 21 summarizes the social media landscape proposed by Fred Cavazza in a blog post[8]. He breaks down these social tools into 4 main categories, "expressing tools allow users to express themselves, discuss and aggregate their social life", "sharing tools allow users to publish and share content", "networking tools allow users to search, connect and interact with each other" and "playing services that now integrate strong social features". Social platforms, like Facebook, Orkut, Hi5, etc., are at the centre of this landscape as they enable us to host and aggregate these different social applications. For instance, you can publish and share your del.icio.us bookmarks, your RSS streams or your microblog posts via the Facebook news feed, thanks to dedicated Facebook applications. This integration of various means for publishing and socializing enables us to quickly share, recommend and propagate information to our social network, trigger reactions, and finally enrich it. Sociologists now have access to a valuable source of social data that capture characteristics of our societies with permanently evolving web usages and web technologies. The need for some appropriate representation to exploit them has consequently emerged.

Historical graphs representations are applied to model and manipulate these online social networks in the same manner as for offline social network. Social networks with symmetric relationships like Facebook are represented by undirected graphs similar to the one in Figure 23. Inversely, directed graphs are well suited to model social networks

[8] http://www.fredcavazza.net/2009/04/10/social-media-landscape-redux/

with non-symmetric relations like the "follows" relationships of Twitter. In weighted graphs, weights are useful for representing the frequency of interactions between people through messages or comments. Social networks like Facebook propose adding labels (e.g. family, friends, favourite) on relationships to represent the type of social links they represent and help user filter their contacts. Finally, sharing sites (e.g., Flickr, Youtube, Delicious) allow interaction on shared content (e.g. photos, videos, bookmarks), connecting them through virtual artefacts. Such social networks are represented with bipartite graphs, with two types of nodes and edges that link nodes of each type. These sites sometime produce complex relationships involving more than two types of resources (e.g. a user, a document and a tag) that are represented by hyperedges producing hypergraphs.

The social tagging activity, which consists in collaboratively classifying resources by annotating them with tags, emerged with the advent of Web 2.0, and became the dominant tool for classification of online shared resources (Flickr, del.icio.us). A set of tags built from usage of such applications forms a folksonomy that can be seen as a shared vocabulary that is both originated by, and familiar to, its primary users. [Limpens 2010] defines a folksonomy as follow:

**Definition 52. Folksonomy**: A folksonomy is defined as a collection of taggings. In formal term, a folksonomy is defined by [Hotho et al 2006] as a tuple $F := (U, T, R, Y)$ where U, T, and R are finite sets, whose elements are called users, tags, and tagged resources, respectively. Y is the set of tagging instances such that $Y \subseteq U \times T \times R$. [Mika 2005a] also proposed a graph definition where a folksonomy can be seen as tripartite hypergraph $H(F) = (V, E)$ where the vertices are given by $V = U \cup T \cup R$ and the edges by $E = u,t,r \mid (u,t,r) \in F$.

This collaborative classification of web resources can be further analyzed in order to decipher implicit links between users who use similar vocabularies or tag the same content, highlighting the existence of common interests. As more people use these social applications they expose more and more of their lives and of their social networks. [Mika 2005a] models the social tagging with a tripartite graph (see Figure 22), where the nodes are users, tags and annotated resources while the edges of this graph represent the ternary association of a tag to a resource by an actor. Then Mika extracts and focuses more closely at two bipartite subgraphs. The first one connects actors to the tags they use to annotate resources. This graph allows inferring an affiliate social network; edges connect actors who used that shared tags with weights representing the number of shared tags. Mika deduces similarly a network of tags, where an edge between two tags is weighted by the number of users using both these tags. The second extracted bipartite graph connects the tags to the resources and provides an additional network of tags; a link between two tags is weighted by the number of instances annotated with these tags. Mika builds such graphs from a social tagging dataset extracted from the social bookmarking service delicious.com. He ran different metrics of social network analysis to infer lightweight semantics between tags. In particular he focused on the tags networks. He observed that tags with higher local clustering coefficients represent more specialized concepts. Inversely, tags with lower clustering coefficients and a high betweenness centrality represent broader concepts. Finally, running a community detection algorithm offers to detect interest communities of interests on the affiliate network of users sharing same tags.

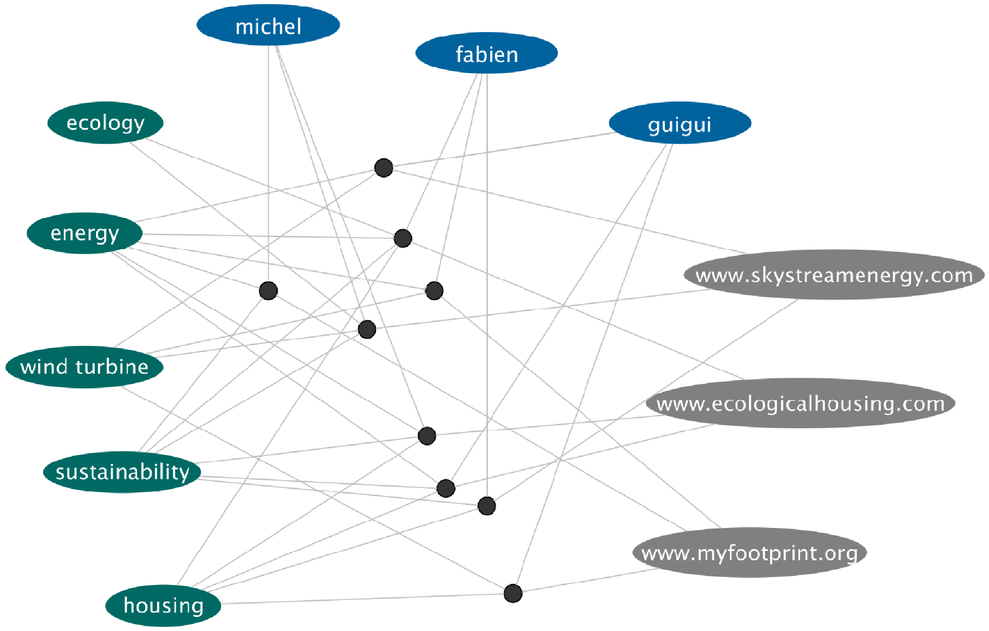


**Figure 22.** Tripartite graph of a folksonomy. Each edge links a person (in blue), a tag (in green) and a resource (in grey) [Limpens 2010].

[Bothorel & Bouklit 2008] model a folksonomy extracted from Flickr with a hypergraph. They generalized the community detection algorithm of [Girvan & Newman 2002] to tripartite graphs in order to generate thematic tag clouds and detect consensus or conflicts in the use of tags among the communities.

[Brandes et al 2009] propose an interesting approach to analyse interactions in Wikipedia and to get insight into the communities emerging from collaborative writing of knowledge writing. They define links between the different authors of versions of a same page, considering co-editing and revising a page as mediated interactions. Moreover, they propose to label these interactions by defining and detecting three possible actions on the content of a wiki page version: augment its content, partial delete or total removal by restoring a previous version. Moreover, as authors tend to contribute many times on one page and several times on domain related page, same authors interact regularly on different versions and pages. Consequently, they add weights on the interactions between authors to highlight the intensity of their interactions. With the combination of the weights and the different types of interactions between authors it is possible to infer agreements, disagreements and even conflicts. With this approach, the authors obtained three weighted graphs, one for each type of interaction, on which they performed a network analysis. In particular they detected communities and highly intermediary people, highlighting conflicting communities and topics.

The online social networking sites have becomed key applications of Web 2.0 and experience the highest audience. Among the former, we find Friendster[9] and Orkut[10], but the best known and most visited today are Facebook[11] and Twitter[12]. These sites allow their users (1) to maintain their offline relationships and (2) to develop online affiliations. The huge audience of these sites (over 500 million for Facebook [Facebook statistics 2010] and more than 100 millions for Twitter in 2010) and the accessibility of their social data by REST[13] APIs constitute valuable sources to analyze social networks of very large sizes. Indeed, as users explicitly state their relationships and interactions, it is no more necessary to infer social networks with heuristics or interviews as the very nature of these relationships is provided. However, most of the time inferences are still necessary to merge identities and relationships across social network services which run on their own data silos and do not offer interoperability. It is true that some aggregating services propose to centralize content from multiple social networks. But aggregation of a new service requires learning a new API. Some initiatives like Google Open Social[14] tried to overcome such issues by proposing the use of a single common API but are still not adopted by most of the major social services. Figure 23 shows a part of the social network of Guillaume Erétéo on Facebook built with the TouchGraph[15] application that exploits the Facebook API.

[Bonneau et al 2009] sampled and analyzed the student networks of Stanford and Harvard on Facebook, using only the eight friends displayed on the public profiles of students on Facebook. To obtain this sample they simply crawled the public Facebook profile of students that randomly highlights eight of their friend. By performing a social network analysis of this social network sample, they have shown that a small subset of the network is enough to obtain relevant information about the network itself. In particular they accurately estimated the global structure of the social network such as the community structure, the diameter and the density. They also obtained a great insight into strategic positions with most central people according to their degrees and their betweenness centralities.

[9] http://www.friendster.com

[10] http://www.orkut.com

[11] http://www.facebook.com

[12] http://www.twitter.com

[13] Representational State Transfer http://fr.wikipedia.org/wiki/Representational_State_Transfer

[14] Google Open Social http://code.google.com/intl/fr/apis/opensocial/

[15] http://www.touchgraph.com

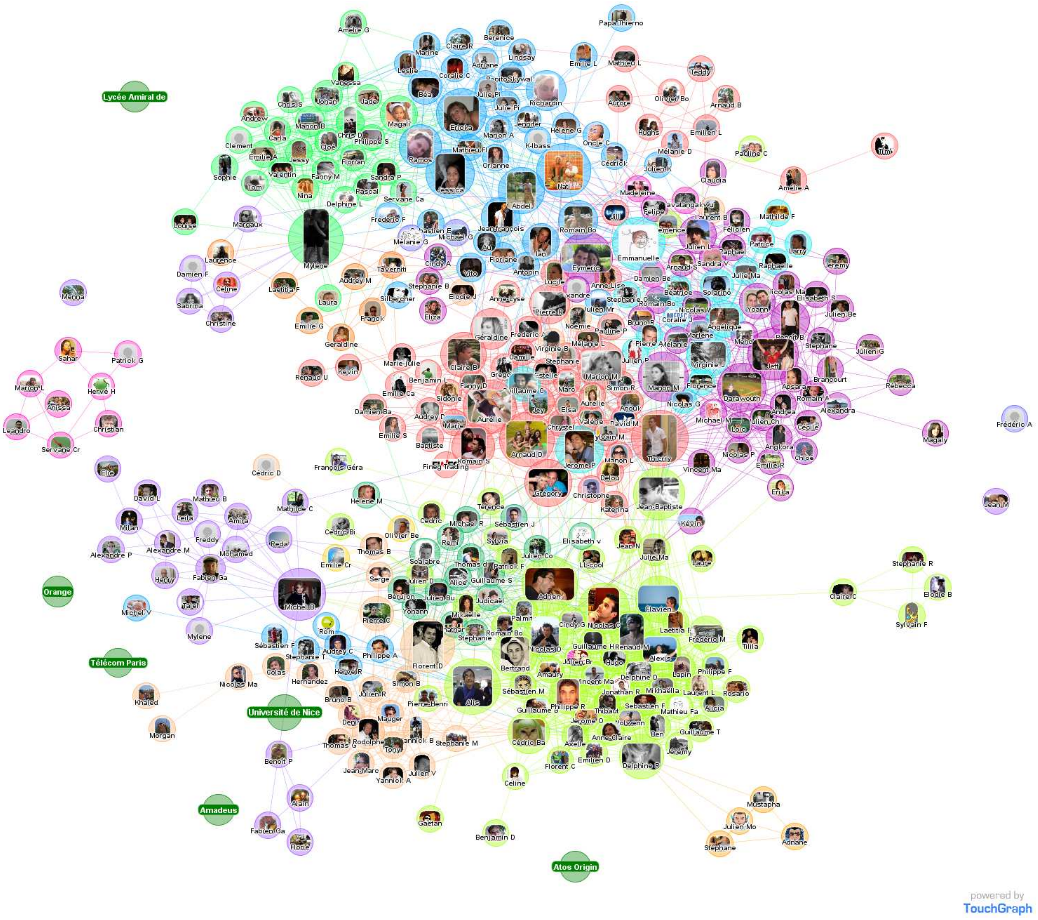


**Figure 23.** Visualisation of the social network of Guillaume in September 2010.

Twitter is another one of the largest social networks. The principles of Twitter consist in publicly publishing short posts (also called statuses), no longer than 140 characters, directly visible to users following the author's posts (and accessible through search otherwise), and to follow other users to watch their own posts. This “follow principle” works in a directed way: Peter can follow Paul's posts while Paul is not necessarily following Peter. Based on this simplicity of use, conventions in post contents enable users to interact in a richer way. One can mention a user by prefixing his username with the character '@'. Then, one can share the post of a user he follows with a "retweet", by mentioning the initial emitter and adding the string 'RT' (or 'via' in some cases) at the beginning of the tweets he is going to forward. Finally, people gather in a public conversation, accessible through the search functionality, by negotiating a common tag they add in their posts. These tags are keywords prefixed with the character '#', they are called hashtags. Twitter is currently considered as one of the most efficient way to share information and raises lots of interests in the social network analysis community. In particular, [Kwak et al 2010] obtained interesting and questioning results by analyzing a huge datasets extracted from Twitter containing both posts and connections between people. Despite the directed nature of the “follow relations” of Twitter, which could have made longer connections between users, they found an average distance of 4 between any Twitter users. This average distance is considerably shorter than the one historically observed in offline social networks. In fact it is a direct consequence of the absence of reciprocity in Twitter way of linking. Without this constraint, creating links becomes so free that the density of links get very high and some people reach huge in and out degrees (e.g. some stars have millions of followers). However, the authors conclude that this non usual network structure probably results in the fact that Twitter is now more a new media than a real social network. But, despite of its simplicity, Twitter offers many ways of interaction, namely social links, between its users, which still are social actors. Consequently even if some users do not act socially, i.e. with broadcasting purposes in mind, Twitter is still a real social network according to the social links it offers and the social nature of the majority of its members. In my opinion, the real question is: how can we analyze relevant relationships to obtain relevant insights on social networks? We previously described how [Brandes et al 2009] started answering it by handling different types of interactions in wikis to obtain a better interpretation of the community structure of Wikipedia. [Kwak et al 2010] also started answering this question by comparing different users' ranking methods, the "follow link" structure (degrees and page rank) or post "retweet" by other users, which result in a completely different classification of "most influential users".

### 3.3.2 Social Network Analysis: the Need for Semantics

You have probably ever been surprised to discover a relationship between two of your acquaintances and concluded: "This is definitively a small world". Effectively, we have previously seen in this chapter that the "small world" property it is one of the most famous characteristics of social networks highlighted by the historical experiment of [Milgram 1967] described in section 3.1.2. In this section, we will underline that the conclusion of this experiment has to be contextualized with the semantics of the relationships and the mode of communication used to transmit the letters. In order to reach the final recipient the current possessor of the letter has to *mail* it to a person he *knew* and that he thinks would be the most efficient one to help sending the letter to its destination. So the type of communication is the *mail* and the type of relationship is *knows*. Let us decompose and analyze these two constraints. However, the *know* relationship is very broad and include *friends, family, works* or even simple *acquaintance*. Do we only consider reciprocal or also consider directed relationships? How many steps would have been observed if we asked to transmit the letter through *family* relationships? In addition, the mail was probably one of the most used *one to one* means of communication in 1967 but it's clearly no more the case. How many steps would have been observed if we had asked to transmit the letter hand to hand? How many steps would have been observed if we had asked to transmit the message of the letter with a *one to many* communication mean, such as a high audience radio or TV channel? All these questions highlight sub cases of the guidelines proposed by Milgram that could impact the conclusion of the experiment.

Today, nearly 2 billion people have an internet access. More and more communicating tools and social applications have been built on top of internet. In particular conversation solutions have been developed, such as IRC, mail and instant messaging.

Consequently, these communication solutions have rained interests and social networks have been extracted from email and instant messaging communication [Tyler et al 2003] [Leskovec & Horvitz 2008]. How many steps are necessary to reach anyone by instant messages? On the Microsoft instant messaging network, [Leskovec & Horvitz 2008] observed "that there are about '7 degrees of separation' among people". In addition, since its creation in 1992, the World Wide Web is connecting more and more users, all around the world, a phenomenon that the so-called web 2.0 [O'Reilly 2005] has amplified. In particular social networking sites, such as Facebook and Twitter, have gather hundred of millions of persons. How many steps are necessary for news to propagate on such online social networks? [Kwak et al 2010] answered that despite the directed nature of the *follow* links of Twitter, every 2 users of this network are linked by an average of 4 relations.

These few examples clearly highlight "the diversity of degrees of separation" when we focus on the diversity of types of relationships and interactions between people. In fact the evolution of communication technologies has soon produced dramatic changes to come in our society in particular by leveraging the density of the human connections. After having overcome space limits by transporting people in most points of the world, the human tackled the time limit of communication. The first major change was the deployment of telegraph network in the first part of 19th century, which offered near instant communication and launched the area of telecommunication. The birth of this technology has been quickly understood as a major evolution in our society, as underlined by the famous quote of Hawthorne: "Is it a fact - or have I dreamt it - that, by means of electricity, the world of matter has become a great nerve, vibrating thousands of miles in a breathless point of time?" [Hawthorne 1851]. Then, the time and geographical limits of communication have been continuously tackled with notably the first transatlantic cable in 1858 and the invention of the phone in 1867 by Graham Bell.

While the 19th century revolutionized the time dimension of the communication, the 20th dramatically changed the mode of communication. In the beginning of the 20th century, the creation of the radio and the TV promoted a new mode of communication: broadcasting. The telegraph and the phone enable a real time bidirectional communication, while broadcasting is a one-way instant communication: from one emitter to many receivers. Consequently, some important actors (e.g. brands, politics, celebrities, etc.) gained a privileged access to the starting point of broadcasting channels to diffuse their messages toward a mass of people. The early hours of the Web revolutionized this approach and democratized broadcasting communication, by enabling anyone to publish a web page on top of Internet. However, this possibility initially required some technical knowledge prerequisites, which still restricted the access to this broadcasting communication. The evolution of web technologies has quickly overcome these technical limits. Web sites were turned into web applications that made web resources writable through a web browser. This led to the Web 2.0 that promoted not only new tools but new means of communication: social medias. The beginning of the 21th century witnessed the outburst of a *many to many* communication.

## Toward a "smaller world"?

While the range of online social tools is still growing, the easy and massive adoption of these tools by nearly 2 billion of web users started producing a huge social impact: "When we change the way we communicate, we change the society" [Shirky 2008]. One of the earliest striking examples is the history of the Stolen Sidekick phone in 2006 in New York. In order to retrieve a stolen smart phone in New York, a man managed to gather attention of millions of people on this thief by smartly exploiting online communication. As a consequence, this stolen phone, among millions of others ones, got back to its proprietary in only 3 weeks while any other one would have definitively vanished in one of the biggest cities. This early example of social impacts, introduced by online collaboration, illustrates how millions of people became "closer" with the World Wide Web, and how easily they connect and gather. By putting web users at the core of web applications, we introduced many new types of links between persons.

First, online social network services, such as Facebook and Twitter, have turned the theoretical cognitive limit of 150 stable relationships that one can maintain, the Dunbar number [Dunbar 1998], into much higher values. On one hand, these services assist the relationship maintenance in diffusing information to its social network and getting reactions about it. In the other hand, these services also assist relationship maintenance for getting information from its social network and pushing reactions. In 2010, having reached more than half a billion users, if Facebook was a country it would be the third in terms of size. Facebook's statistics state that the average number of contacts of this service's users is 130 [Facebook press 2010], and it is common to find users with a number of contacts ranging between 200 and 500.

Then content sharing sites and publishing platforms link people through online artefacts that affiliate web users and foster interactions. For instance, readers of a blog can engage in the same conversation through comment, social bookmarking users can share and bookmark the same pages, Facebook users can declare a similar interest about online resources using functionalities of the Facebook's Open Graph Protocol, and people become implicitly affiliated by the specific vocabulary of their community through search engine and tagging systems.

Finally, data and knowledge representations produce a growing number of links and affiliations between web users. Data connect distributed identities and activities. The striking example is the classical "friend finder" options of many online applications that offers to connect users that have already communicated by mail or by other Web 2.0 services. Some data enable to disambiguate identities and activities even if they are located on different web sites. Then, people can be affiliated by the metadata used to describe their resources or elements of their profiles, which can support future interactions. For instance if Peter annotates a resource with the tag "photovoltaic" and jack another resource with the tag "renewable energy", they can be affiliated by the fact "photovoltaic is related to renewable energy".

Until which point can we consider a link, subjectively considered as social or not, between two web users? What is the real nature of these links and which interpretation can we deduce from an analysis of the corresponding networks and sub networks? With the very large range of types of links offered by means of a simple URL, are social web users getting closer to "one degree of separation" than "six degree of separation"? *This is a whole matter of semantics*. The interpretation of the analysis of a network is highly related to the meaning, and the interpretation, given to the links used to analyze the social network.

**How to analyze this whole set of heterogeneous social links?**

Network characteristics are modified by considering different semantics and types of relationships in the social network. While the advances of telecommunication produce more and more interaction, and affiliation means, the resulting social network highlights different dimensions and perspectives. Each of these perspectives represents a sub network with its own characteristics in terms of structure, distribution of activities, and strategic positions. More and more social software highlights the need to better handle the characteristics of social networks, both for providing services to the actors of these social networks and for interacting with them. In particular, our scenario of social business intelligence (described in chapter 2) highlight a great interest in efficiently mining social data, for detecting online strategic positions and gaining insight from emerging communities. We have seen that social network analysis offers effective algorithms to investigate the linking structure of social network. However, the consideration of the diversity of relationships that can be mined argues for a rich representation of social networks and a social network analysis that exploits this richness. This is where Semantic Web frameworks can help by offering a richly typed graph structure and an efficient querying language to mine such graphs.

**On one side social network analysis proposes metrics and graph algorithms to characterize the structure of a social network, understand the distribution of actors and activities, and identify important actors and strategic positions.** While these algorithms mine efficiently the flat graph structure of the network to extract structural characteristics, they do not leverage, for instance, the types of the links, the different profiles or roles of the actors. However, these may be extremely important when looking at the social activity from different perspectives for instance in the context of your professional, family, or friend activity.

**On the other side, Semantic Web frameworks enable us to represent distributed identities, activities and relationships in a uniform directed typed graph structure while being located on different sites.** Both nodes and relationships of social graphs can be richly typed with concepts from specific vocabularies. It offers a semantic dimension to social graphs that semantic engines can mine and enrich with inferences like transitive closure for subsumption (e.g. if someone is a man the system knows he is also a person) and automatically handled for querying and transforming such data. However, while this whole stack of semantic technologies is efficient for representing, exchanging and extracting rich social data, it still lacks graph operators for meeting SNA requirement.

**There is clearly a need to combine both graph models and merge their capabilities**. In particular we need to investigate (1) how to query and transform semantic social graphs to perform a social network analysis that takes advantage of the semantic dimension of social networks and (2) how to augment social data with the output of semantic social network analysis in order to organize and structure them.

# 4. Semantic Social Network Representation

While human interactions in web 2.0 sites produce a huge amount of social data, capturing more and more aspects of physical social networks, this decentralized process suffers from interoperability and linking between diffused data. In fact, such rich and spread-out data cannot be represented using only the models of graph theory outlined in the precedent chapter without some loss of information. These representations lack of typing and are not necessarily adapted for exchanging and mining social data across applications. Semantic web technologies tackle these requirements and are now used to represent online social networks and exchange social data across applications.

## 4.1 From Data Silos to a Global Semantic Social Graph

Average web users have accounts on different social networks (Facebook, LinkedIn, Twitter, etc.), chat services (Microsoft Messenger, AIM, etc.), sharing platforms (Youtube, Flickr, delicious, etc.), and many of them manage one or more blogs. Some use even a wider set of social sites ranging from main opinion and recommendation sharing sites (about products, places, events, etc.) to niche networks and alternative, underground sites that appeared with the web 2.0 sites. In addition, most of classical personal applications such as mails and bookmarks that are available in the form of web applications now dispose of strong social features. All these applications are used for different purposes, in different context and collect different characteristics of their users. The freeform of online contributions make unpredictable the way people will use them. Consequently, users define freely and unpredictably their personal goals, their own context of use and the details of the information they share. This behaviour results in a sparse distribution of identities, roles and activities across web applications. People share different parts of their identities and play different roles depending of the social service they use. Moreover they sometime use avatars (e.g. in forums) or simply express themselves anonymously (e.g. in blog comments). Along these different uses they build different types of relationships, which are characterized by the context and the roles used when they are initiated.

Most social services propose a set of REST web services accessible through some kind of APIs, to create and to modify the social data of their users from external applications or browser extensions. For instance, applications aggregating social services (e.g. FriendFeed[16], Seesmic[17], Tweedeck[18]) exploit these APIs to offer a unique access point to all these distributed activities. However, when third applications propose to match social data coming from different sites, they suffer the lack of interoperability between data. Despite some initiatives that tried to propose consensus for designing social API, most of the social services still have (1) their own way of accessing their data, and (2) their own format to represent these data. First, even if most of the APIs of social sites are accessible through REST web services, these APIs differ in the signature of their web services and in their architecture even if they offer similar functionalities and expose equivalent data. Then, these APIs also differ in the schemas of the data they return. They all expose data in structured format, XML and JSon in most of the cases, but with different data models and without any semantics while they represent similar concepts.

Users access several social web sites and produce data that form explicitly or implicitly social graphs. But since the data of these graphs are trapped in the data silos of the

[16] http://friendfeed.com/

[17] http://seesmic.com/

[18] http://www.tweetdeck.com

different social services, these graphs remain disconnected. In order to consolidate and interconnect these social graphs we need standards to describe data, to connect their semantics, and a common protocol to access them. Semantic web frameworks answer the problem of representing and exchanging data on the web with a rich typed graph model (RDF[19]), schema definition frameworks (RDFS[19] and OWL[19]), and a protocol with a query language for accessing data (SPARQL[19]).

### 4.1.1 RDF: a Standard Resource Description Framework

RDF enables us to make assertions and describe resources with triples (subject, predicate, object) that can be viewed as "the subject, verb and object of an elementary sentence", "a natural way to describe the vast majority of the data processed by machines" [Berners-Lee 2001]:

- The subject represents the described resource.
- The predicate represents the property used to describe the resource.
- The object represents the value of the property for the described resource.

RDF provides the basis to make such triple description on top of the linking structure of the web:

- URIs are used to identify the subject and the predicate of a triple which enable the description to reference, without any ambiguity, the resource that is described and the property that is used to make this description. The object can be as well a URI to link the subject to another resource or a literal to provide a value. Blank nodes may also be used to introduce anonymous subjects or objects.
- The predicate `rdf:type` is the basis property to type resources.
- The type `rdf:Property` is the base type to define properties that are with URIs the atomic elements of a triple.

With this basis, “anyone can make any statements about any resource”[20]. Any property can be declared to define an attribute, which can be used to describe any resource, since it is identified by a URI. For instance to describe the paper "the semantic web" by declaring its attribute *creator* with the resource Tim Berners Lee, namely "*the semantic web" has for creator Tim Bernes Lee*, we produce the following triple:

```
(<http://www.scientificamerican.com/article.cfm?id=the-
semantic-web>, <http://purl.org/dc/elements/1.1/creator>,
<http://www.w3.org/People/Berners-Lee/card#i>)
```

With respectively:

[19] Semantic Web, W3C, http://www.w3.org/2001/sw/

[20] Resource Description Framework (RDF):Concepts and Abstract Syntax http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-anyone

- `http://www.scientificamerican.com/article.cfm?id=the-semantic-web` the URI that identifies the article "the semantic web".
- `http://purl.org/dc/elements/1.1/creator` the URI that identifies the property *creator*.
- `http://www.w3.org/People/Berners-Lee/card#i` the URI that identifies Tim Berners-Lee

Similarly, we can state that *Tim-Berners Lee's first name is "Tim"* with the following triple:

```
(<http://www.w3.org/People/Berners-Lee/card#i>,
<http://xmlns.com/foaf/0.1/firstName>, "Tim")
```

With respectively:

- `http://www.w3.org/People/Berners-Lee/card#i` the URI that identifies Tim Berners-Lee
- `http://xmlns.com/foaf/0.1/firstName` the URI that identifies the property *first name*.
- `Tim` the string value representing the first name of Tim Berners Lee

By means of URIs, descriptions share uniform identifiers that link them between each other. These triples can be seen as arcs (properties) between nodes (URIs, literals) providing RDF with a directed labelled graph structure that is well suited to represent and link data and metadata about heterogeneous content on different web sites. They allow data to be spread across the internet and intranet networks, representing actors, content and relationships, while being represented and linked with a uniform RDF graph structure. The URIs are used to identify resources and properties, link distributed identities and activities. Same URIs identify the same resources and two URIs describing the same resource can be unified with a single description stating so. For instance, the Figure 24 represents the graph that corresponds to the two precedent descriptions about Tim Berners-Lee. This graph can be freely augmented by any description. For instance, the graph of the Figure 25 augments the graph of the Figure 24 with two triples that propose the descriptions stating that *Tim Berners-Lee holds the online account http://twitter.com/timberners_lee* and that the account *http://twitter.com/ereteog follows http://twitter.com/timberners_lee*.

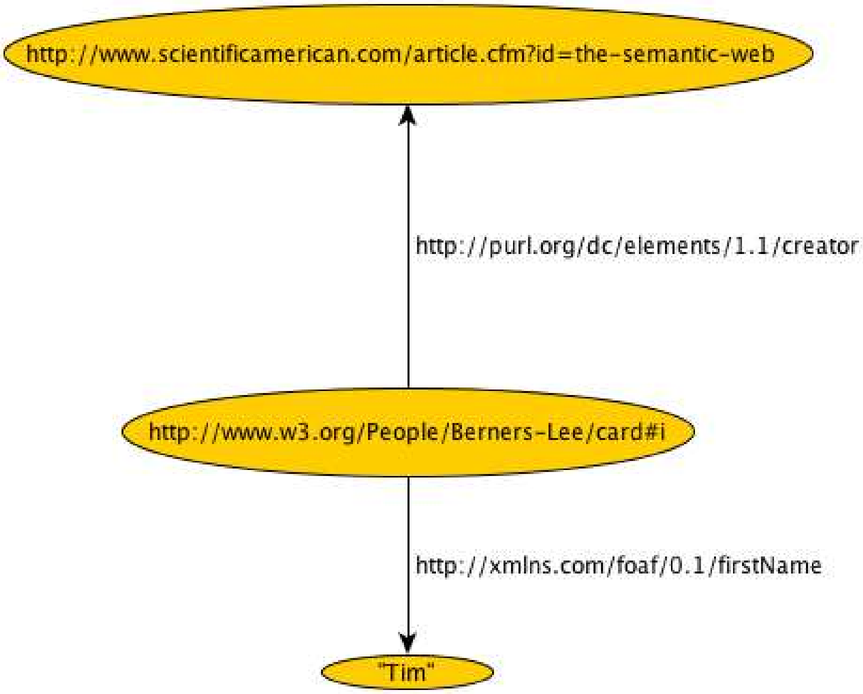

http://www.scientificamerican.com/article.cfm?id=the-semantic-web
http://purl.org/dc/elements/1.1/creator
http://www.w3.org/People/Berners-Lee/card#i
http://xmlns.com/foaf/0.1/firstName
"Tim"

**Figure 24.** Graph representation of the triples that describe the sentences "*the semantic web" has for creator Tim Bernes Lee* and *Tim-Berners Lee's first name is "Tim"*.

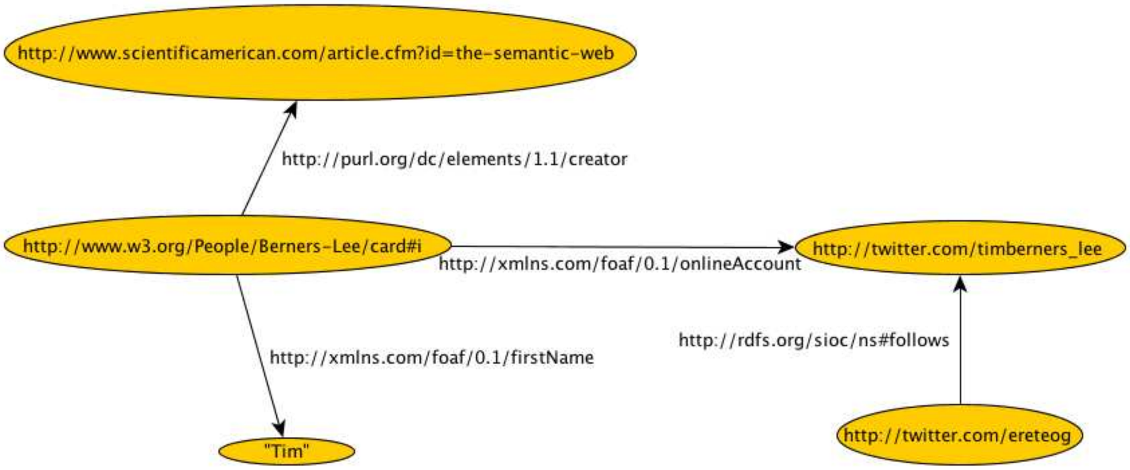

http://www.scientificamerican.com/article.cfm?id=the-semantic-web
http://purl.org/dc/elements/1.1/creator
http://www.w3.org/People/Berners-Lee/card#i
http://xmlns.com/foaf/0.1/onlineAccount
http://twitter.com/timberners_lee
http://rdfs.org/sioc/ns#follows
http://xmlns.com/foaf/0.1/firstName
"Tim"
http://twitter.com/ereteog

**Figure 25.** Graph representation of the triples that describe the sentences (1)"*the semantic web" has for creator Tim Berners Lee* and (2) *Tim-Berners Lee's first name is "Tim"* augmented with the triples describing the sentences (3) *Tim Berners-Lee holds the online account http://twitter.com/timberners_lee* and (4) *http://twitter.com/ereteog follows http://twitter.com/timberners_lee*

Formally speaking the triples of an RDF description can be seen as the labelled arcs of an Entity-Relation graph [Baget et al 2008], ERGraph, defined as follow:

**Definition 53. ERGraph**: An ERGraph relative to a set of labels L is a 4-tuple G=(EG, RG, nG, lG) where :

- $E_G$ and $R_G$ are two disjoint finite sets respectively, of nodes called entities and of hyperedges called relations.
- $n_G : R_G \rightarrow E_G^*$ associates to each relation a finite tuple of entities called the arguments of the relation. If $n_G(r)=(e_1,\ldots,e_k)$ we note $n_G^i(r)=e_i$ the i$^{th}$ argument of $r$.
- $l_G : E_G \cup R_G \rightarrow L$ is a labelling function of entities and relations.

### 4.1.2 Ontologies and Resource Description Framework Schema

An ontology is "a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application" [Gruber 2009].

RDF enables us to define properties and descriptions about concepts in order to structure them and describe domain knowledge. However RDF does not provide the basis primitives to define any logical application of properties, which is necessary for defining an ontology. RDF schema (RDFS) defines a vocabulary, on top of RDF, that provides, in combination with RDF, the basis primitives to define ontologies:

- The resource `rdfs:Class` is a primitive to define the classes of a domain knowledge
- The properties `rdfs:domain` and `rdfs:range` enable us to positively constrain the use of properties and define "the classes of resource to which they apply"[21]. A domain of a property defines a class typing for the resources that are the subject of this property. A range of a property defines a class typing for the resources that are the object of this property.
- The properties `rdfs:subPropertyof` and `rdfs:subClassOf` enable us to structure the property and the classes of an ontology to define taxonomical relations. The inheritance properties are frequently used between classes and properties to define taxonomies (e.g., a *Post* is a sub class of *Document* and *brother* is a sub property of *sibling*). `rdfs:subPropertyof` and `rdfs:subClassOf` are transitive.

These three basic elements, offer to richly type the resource and the properties of an RDF graph with structured classes and properties, while providing a minimum of logical constraint to reason on such graphs. In addition we have a set of type inference rules that enable us to enrich RDF triples. In particular, the positive constraint on the domain and the range offer to enrich the types of its resources when they are involved in a property. We talk about positive constraints because a resource that has (or resp. is the value of) a property will be typed with all the classes that are declared as domain (resp.

[21] RDF Schema, http://www.w3.org/TR/rdf-schema/

range) of this property. Other type inference rules are available[22] such as the transitive closure.

The Figure 26 represents a sample of the linked schemas FOAF [Brickley & Miller 2004] and SIOC (that are detailed in section 4.2). This piece of schemas represents the link between a person and his online accounts, and the *follows* relation that is a common relation between users of social applications. The property `foaf:account` defines an online account of a resource and implies that its subject is a `foaf:Agent` and its value is a `foaf:OnlineAccount`. The property `sioc:follows` implies that both its subject and object are typed with the class `sioc:UserAccount`, which is a subclass of `foaf:OnlineAccount`.

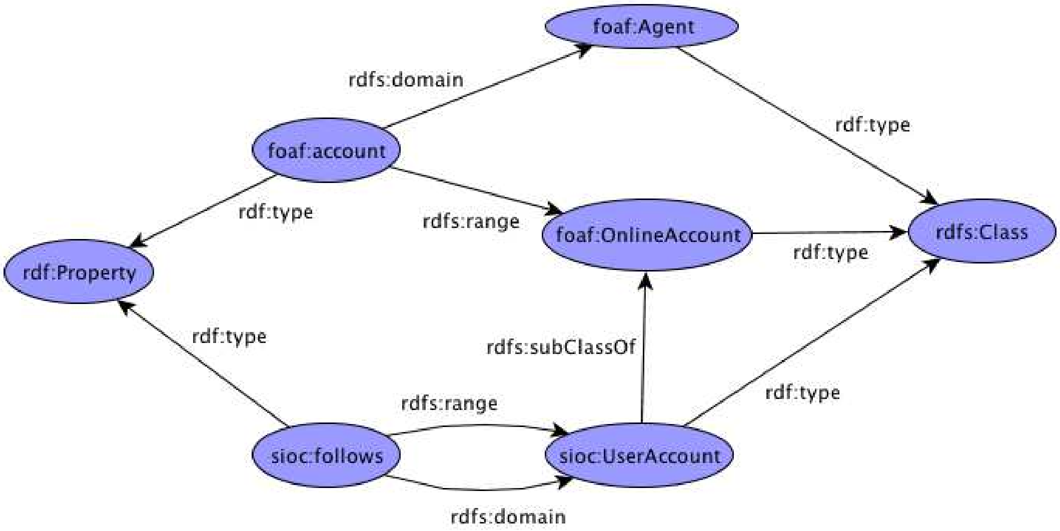


**Figure 26.** Sample of the linked schemas FOAF [Brickley & Miller 2004] and SIOC [Breslin et al 2005] (that are detailed in section 4.2). This piece of schemas represents the link between a person and its online accounts, and the frequently used *follows* relation between users of social applications.

This schema enables us to enrich the triple represented by the graph of the Figure 25. In particular, it enables us to complete the typing of its resources with type inference from the positive constraints that are defined on properties. For instance the resource `<http://www.w3.org/People/Berners-Lee/card#i>` is the subject of the property `foaf:account` which has for domain `foaf:Agent` so we type this resource with the class `foaf:Agent`. Then, the range of this property is `foaf:OnlineAccount`, so this class is inferred as the type of the resource `<http://twitter.com/timberners_lee>`. Similarly, the property `sioc:follow` has for domain and range `sioc:UserAccount`, so the two resources `<http://twitter.com/timberners_lee>` and `<http://twitter.com/ereteog>` are inferred as typed with the class `sioc:UserAccount`. Since, `sioc:UserAccount` is a subclass of `foaf:OnlineAccount`, we also type with this last class the resources

[22] http://www.w3.org/TR/rdf-mt/

`<http://twitter.com/timberners_lee>` and `<http://twitter.com/ereteog>`. The Figure 27 represents the graph that results from the type inference from the schema of the Figure 26 on the triples of the Figure 25.

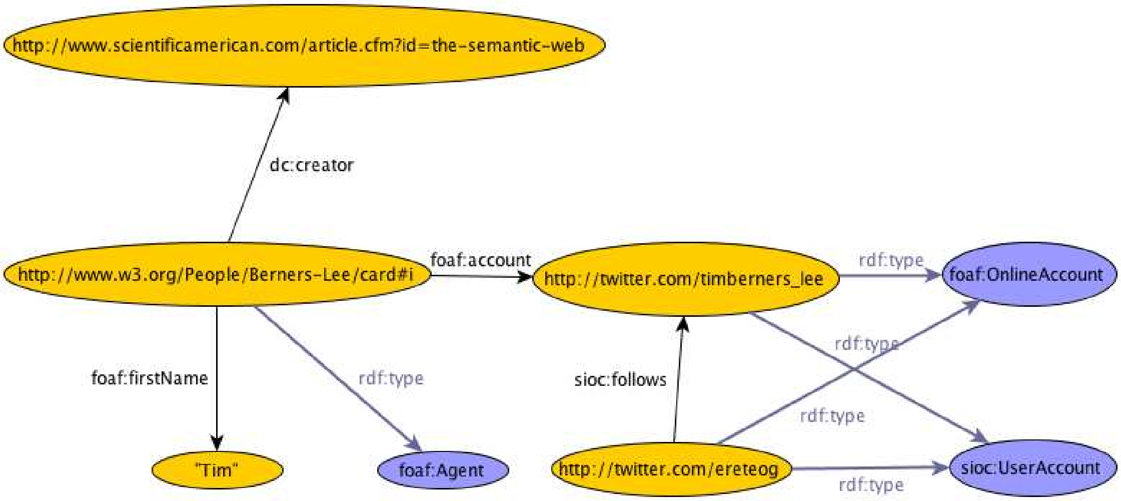

http://www.scientificamerican.com/article.cfm?id=the-semantic-web
dc:creator
http://www.w3.org/People/Berners-Lee/card#i
foaf:account
http://twitter.com/timberners_lee
rdf:type
foaf:OnlineAccount
foaf:firstName
rdf:type
sioc:follows
rdf:type
rdf:type
"Tim"
foaf:Agent
http://twitter.com/ereteog
sioc:UserAccount
rdf:type

**Figure 27.** The graph that results from the type inference from the schema of the Figure 26 on the triples of the Figure 25.

### 4.1.3 An Ontology Web Language for Richer Reasoning on Data

While RDF Schema offers core primitives to structure classes, properties and their applications, OWL (Ontology Web Language) defines additional logical properties that can be applied to classes, properties and individuals. These properties provide a powerful mechanism for enhanced reasoning. The Figure 28 summarizes the characteristics that can be defined on classes, properties or individuals with OWL.

Individuals are mainly provided with relations that enable us to differentiate or join identities. The property `owl:sameAs` enable us to state that two URIs identify the same individual. Inversely, the property `owl:differentFrom` enables us to disambiguate identities and state that two URIs identify different individuals.

Similarly classes are provided with properties about the typing of their individuals. These properties are mainly about the set of individuals typed with classes like equivalence, disjunction or intersection. The property `owl:equivalentClass` enables us to declare that two classes are equivalent and have the same individuals. For instance, "Car can be stated to be equivalentClass to Automobile"[23] and individuals of the type Car are also of the type Automobile, and vice-versa. Inversely, the property `owl:disjointWith` enables us to declare that the set of individuals of two types are disjoint. The property `owl:intersectionOf` enables us to declare that the set of individuals of a class is the intersection of the sets of individuals of others classes. For instance, the class Man is the intersection of the classes Male and Human.

[23] http://www.w3.org/TR/2004/REC-owl-features-20040210/#equivalentClass

Properties are provided a broad set of primitives that address the context of their applications. Among other characteristics and relations OWL offers to define restriction on the cardinality and the values of properties, equality and disjunction between properties and rich typing of properties (e.g. symmetry, transitivity). These properties are used to automatically infer new triples and detect inconsistencies among data. For example, a property *ancestorOf* can be defined as transitive and the inverse of *descendantOf*. Consequently, the triples *Paul ancestorOf Jack* and *Jack ancestorOf Peter* is augmented with the triples *Jack descendantOf Paul*, *Peter descendantOf Jack*, *Paul ancestorOf Peter* and *Peter ancestorOf Paul*.

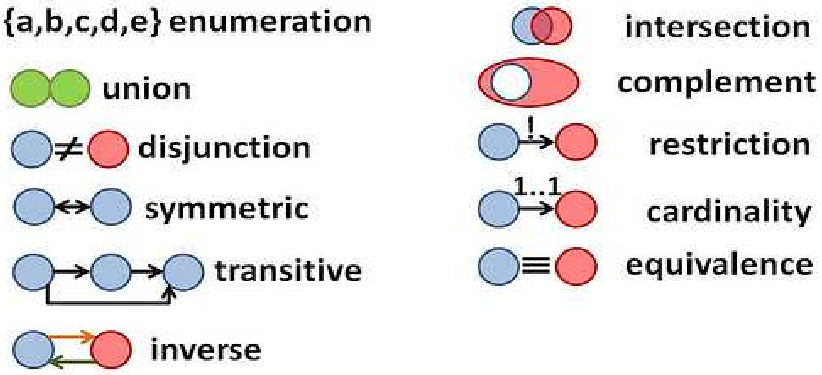


**Figure 28.** "Owl in One" by Fabien Gandon[24].

With the combination of RDF, RDF schema and OWL we dispose both a resource description framework and schema expressivity to build richly typed social graphs. Both nodes and relationships can be richly typed with classes and properties of ontologies with powerful reasoning mechanisms to reason on typing and relations. This provides social graphs with a semantic dimension. Since October 2009, OWL 2[25] is a recommendation that refines and extends OWL with additional constructs and fragments.

### 4.1.4 SPARQL: Protocol and RDF Query Language for Querying and Accessing Data

Once provided with a semantic dimension, we can query this graph with SPARQL[26] (SPARQL Protocol and RDF Query Language). SPARQL is an RDF query language and data access protocol. It defines a query language to query triples, different protocols to send queries and their results across the web, and a result format to exchange these results. The queries are composed of four blocks:

- PREFIX "to declare the schemas used in the query"[27], this clause is optional.

[24] http://www.flickr.com/photos/55704792@N00/3567718085/

[25] http://www.w3.org/TR/owl2-overview/

[26] SPARQL Protocol and RDF Query Language

[27] http://www.slideshare.net/fabien_gandon/sparql-in-a-nutshell

- A clause to determine the type of query and identify the values to be returned. We have three types of clauses:
    - SELECT: "returns all, or a subset of, the variables bound in a query pattern match".
    - CONSTRUCT: "returns an RDF graph constructed by substituting variables in a set of triple templates".
    - ASK "returns a boolean indicating whether a query pattern matches or not".
    - DESCRIBE: "returns an RDF graph that describes the resources found".
- FROM clause "to identify the data source to query"[27]. This clause is optional and the default graph is queried when it is not used.
- WHERE clause, "a conjunction of triples" that defines "the triple/graph pattern to be matched against the triples/graphs of RDF"[27].

The following example proposes to return all the agents and their first names (and only the persons that have a first name):

```
PREFIX foaf: < http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE {
 ?person rdf:type foaf:Agent
 ?person foaf:firstName ?name
}
```

The conjunction of triples of the WHERE clause forms an ERGraph with existing resources and variables. This graph is mapped on the queried RDF graph in order to match same patterns and determine values for the variables of the query. The Figure 29 represents a mapping of the graph of a query on an RDF graph.

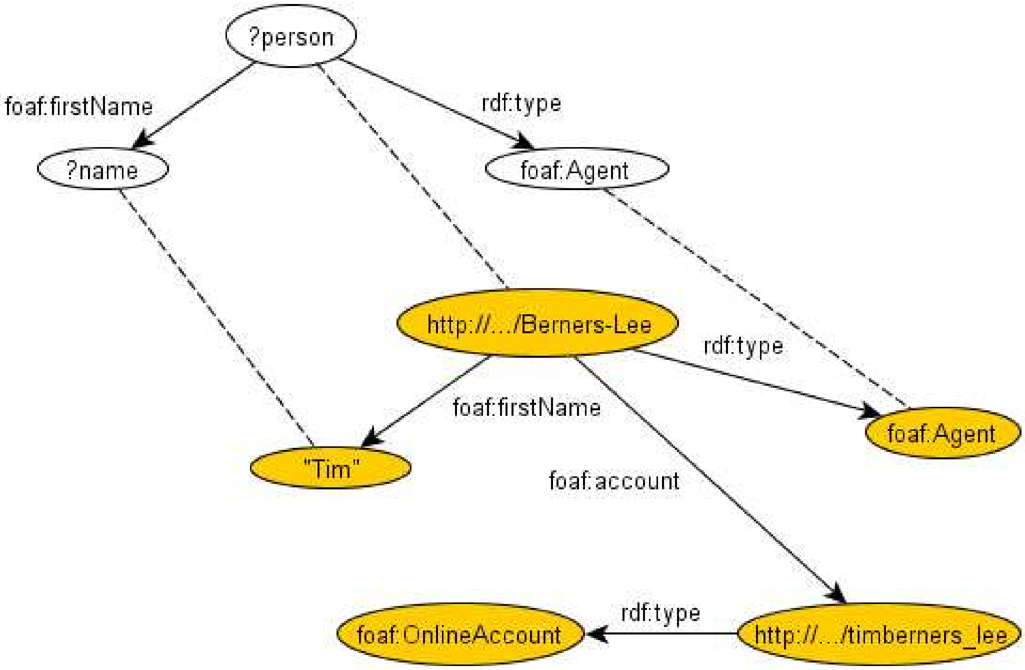


**Figure 29.** Mapping of a query graph on an RDF graph.

Intuitively, a mapping associates entities of a query ERGraph to entities of an ERGraph in a knowledge base of ERGraphs. Mapping entities of graphs is a fundamental operation for comparing and reasoning with graphs [Baget et al 2008]:

**Definition 54. EMapping:** Let *G* and *H* be two ERGraphs, an EMapping from *H* to *G* is a partial function *M* from $E_H$ to $E_G$ *i.e.* a binary relation that associates each element of $E_H$ with at most one element of $E_G$; not every element of $E_H$ has to be associated with an element of $E_G$ unless the mapping is total.

In addition to the mapping of entities linked by the graph, we may want the mapping to enforce that the relations of the graph being mapped are also preserved [Baget et al 2008]:

**Definition 55. ERMapping:** Let *G* and *H* be two ERGraphs, an ERMapping from *H* to *G* is an EMapping *M* from *H* to *G* such that: Let *H'* be the SubERGraph of *H* induced by $M^{-1}(E_G)$, $\forall r' \in R_{H'}$ $\exists r \in R_G$ such that $card(n_{H'}(r')) = card(n_G(r))$ and $\forall 1 \leq i \leq card(n_G(r))$, $M(n_{H'}{}^i(r')) = n_G{}^i(r)$. We call *r* a support of *r'* in *M* and note $r \in M(r')$

Finally, the mapping of the labels of the entities or relations may be done according to an external schema of types of classes and properties. For instance, in the Figure 30 the type of the `<http://www.w3.org/People/Berners-Lee/card#i>` is `foaf:Person`, but we use the schema of FOAF that state that the class `foaf:Person` is a subclass of `foaf:Agent` to perform the mapping.

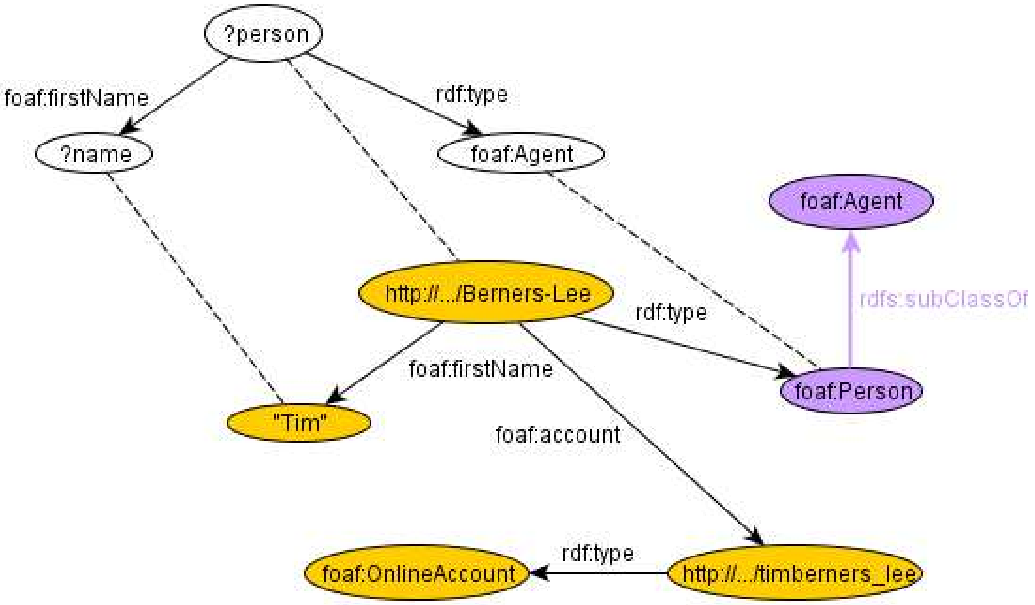


**Figure 30.** Mapping of a query graph on an RDF graph.

A mapping that takes into account an ontology and in particular the pre-order relation defined by its taxonomical skeleton is defined by [Baget et al 2008] as follow:

**Definition 56. Definition of an EMapping$_{<X>}$:** Let $G$ and $H$ be two ERGraphs, and $X$ be a binary relation over $L \times L$. An EMapping$_{<X>}$ from $H$ to $G$ is an EMapping $M$ from H to G such that $\forall e \in M^{-1}(E_G)$, $(l_G(M(e)), l_H(e)) \in X$.

By combining structural constraints and constraints on labelling, we now define the notion of ERMapping$_{<X>}$. In the special (but usual) case where $X$ is a preorder over $L$, the mapping defines the well-known notion of projection [Baget et al 2008]:

**Definition 57. ERMapping$_{<X>}$:** Let $G$ and $H$ be two ERGraphs, and $X$ be a binary relation over $L \times L$. An ERMapping$_{<X>}$ $M$ from $H$ to $G$ is both an EMapping$_{<X>}$ from $H$ to $G$ and an ERMapping from $H$ to $G$ such that:

- Let $H'$ be the SubERGraph of $H$ induced by $M^{-1}(E_G)$
- $\forall r' \in R_{H'}$ $\exists r \in M(r')$ such that $(l_G(r), l_H(r')) \in X$. We call $r$ a support$_{<X>}$ of $r'$ in $M$ and note $r \in M_{<X>}(r')$

**Definition 58. Homomorphism$_{<X>}$:** Let $G$ and $H$ be two ERGraphs, a Homomorphism$_{<x>}$ from $H$ to $G$ is a total ERMapping$_{<X>}$ from $H$ to $G$ where $X$ is a preorder over $L$.

A Homomorphism$_{<x>}$ is also called a Projection. Mapping, especially total mapping, is a basic operation used in many more complex operations e.g. rule application, or query resolution. The semantic engines used in this thesis, CORESE [Corby et al 2004] and KGRAM [Corby & Faron-Zucker 2010], implement the Homomorphism$_{<X>}$ to operationnalize the SPARQL query language for RDF.

### 4.1.5 Linked Data on the Web

Having a standard language for describing resources, for designing schemas that enable us to perform rich reasoning, a standard data access protocol, and a query language, we now need to use connected ontologies in order to achieve the goal of designing a global semantic social graph. The goal of the linked data initiative is to achieve such an objective with any web data.

"Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as 'a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF."[28]

The best practices for linking data are summarized by the five following principles:

1. Use RDF to publish your data
2. Use URIs to identify the things you describe.
3. Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.
4. Provide useful information about the thing when its URI is dereferenced, using standard formats such as HTML when a person looks up or RDF/XML when an agent looks up.
5. Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.

This initiative has led to the creation of connected vocabularies to link data from different domain knowledge. These connected vocabularies form a cloud of data typed with connected ontologies, which is now named *the linked data could*. The Figure 31 is a representation of this cloud, with, in particular, a set of connected social data.

[28] Linked Data http://linkeddata.org

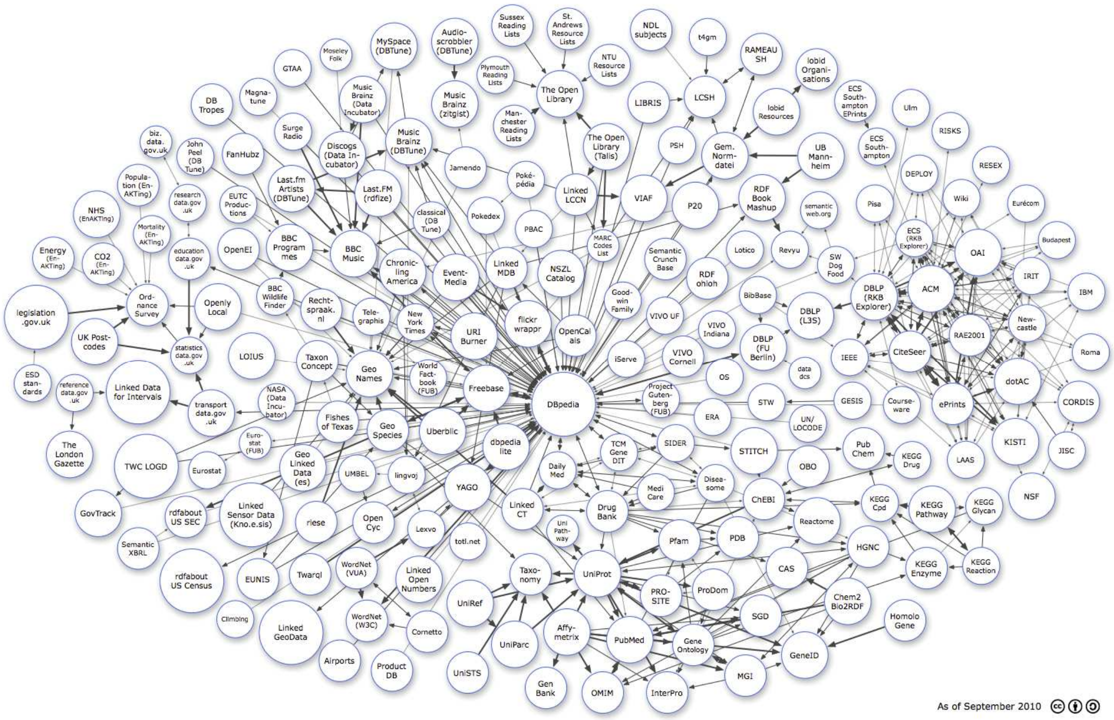


**Figure 31.** The Linked Data Cloud by Richard Cyganiak (DERI, NUI Galway) and Anja Jentzsch (Freie Universität Berlin).

The next section presents how to build and query a global semantic social graph by using, in particular, some of the ontologies of the linked data cloud. We present how to represent and query (1) profiles of peoples and agents on the web, (2) social networks with richly type relationships between persons, (3) published content and mediated interactions, and (4) the vocabulary and affiliations that emerge from social tagging.

## 4.2 Linking and Enriching Social Data with Semantics

Online social data can be seen as a threefold structure:

- Data describing agent profiles. Agent could be persons, groups or organizations.
- Data describing the social network structure.
- Data describing the activities of users and the content they share.

Several ontologies already exist to represent online social networks, and we use them as a basis for our model. All the ontologies we describe are included in the linked data cloud in which the profiles of agents and their networks are described with the FOAF [Brickley & Miller 2004] ontology and user accounts with the SIOC ontology [Breslin et al 2005]. The most popular ontology for describing people, their relationships and their web-based activities is FOAF. The SIOC schema is linked to FOAF, and provides primitives to describe precisely the activities of web people and the content they share. These two ontologies in combination with other more specific ontologies provide the basis to build semantic social networks.

I present here how to represent and query a social network represented with these ontologies. In particular, I show how to query the data built with this models in order to propose the same functionalities than classical online social networks. Web based social services offer different social functionalities:

- to consult the profile of a person.
- to connect with other agents and manage its contacts.
- to generate content and consult the activities of a user or a given network.

### 4.2.1 Representing and Querying Profiles

The FOAF [Brickley & Miller 2004] ontology provides a set of primitives to define the typical attributes of an agent that describe its identity, its online coordinates and its basic web based activities. These attributes match the typical fields that are present in the profile of a person or a group in an online social service.

FOAF propose different properties for describing in particular:

- The identity: `foaf:title`, `foaf:name`, `foaf:firstName`, `foaf:familyName`, `foaf:nick`, `foaf:birthday`, `foaf:depiction`, etc.
- The coordinates: `foaf:mbox`, `foaf:homepage`, `foaf:jabbered`, etc.
- The web based activities: `foaf:account`, `foaf:publication`, `foaf:interest`, etc.

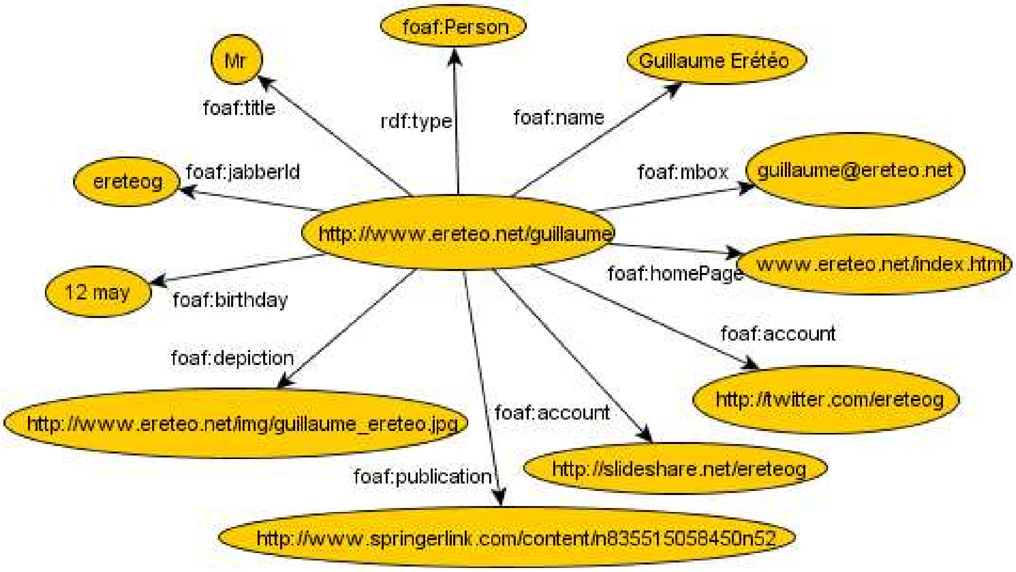

foaf:Person
Mr
Guillaume Erétéo
foaf:title
rdf:type
foaf:name
ereteog
foaf:jabberId
foaf:mbox
guillaume@ereteo.net
http://www.ereteo.net/guillaume
www.ereteo.net/index.html
foaf:homePage
12 may
foaf:birthday
foaf:account
foaf:depiction
http://twitter.com/ereteog
http://www.ereteo.net/img/guillaume_ereteo.jpg
foaf:account
http://slideshare.net/ereteog
foaf:publication
http://www.springerlink.com/content/n83551505845On52

**Figure 32.** Sample of the FOAF profile of Guillaume Erétéo.

Having the URL of the FOAF profile of a person, we can for example ask for his name and his age with the following query with this URL in parameter (an extension of the CORESE SPARQL engine):

```
select ?age ?name where{
 param[url] foaf:name ?name
 param[url] foaf:age ?age
}
```

The conjunction of triples presented in the precedent query can be extended and modified to retrieve any other attributes of an agent.

In some cases we do not dispose of the URL of the FOAF profile to query but only some values of its attributes. These values can be used as well to perform a search or use a unique combination of attributes for identifying an agent. For instance, the following query retrieves the name and the age of the person that holds the online account, which URI is `<http://twitter.com/ereteog>`:

```
select ?age ?name where{
 ?person foaf:account <http://twitter.com/ereteog>
 ?person foaf:name ?name
 ?person foaf:age ?age
}
```

All the combinations of triples that are possible, offer different possibilities to search for agents or retrieve the elements of a profile with its URI or exploiting the values of its different attributes.

### 4.2.2 Representing Social Links and Networks

FOAF also provides the basis to define relationships between persons and groups of agents. The property `foaf:knows` enables us to declare a directed relationship between two persons and state that the subject person knows the object person. The class Group enables us to type a resource as a group and the member agents of a group are related by the property `foaf:member`.

On top of these bases, the RELATIONSHIP ontology provides a whole set of properties for a richer typing of relationships between persons. Most of these properties are sub properties of foaf:knows for describing

- Family relationships: `rel:siblingOf`, `rel:childOf`, `rel:parentOf` or `rel:spouseOf`.
- Love, friend and current life relationships: `rel:friendOf`, `rel:lifePartnerOf`, `rel:neighborOf`, `rel:hasMet` or `rel:ennemyOf`.
- Professional relationships such as `rel:worksWith`, `rel:colleagueOf`, `rel:collaboratesWith` or `rel:apprenticeTo`.

Some relationships do not extends the `foaf:knows` property and mainly represent reputations or feelings such as `rel:ambivalentOf`, `rel:influencedBy` or `rel:whouldLikeToKnow`.

The properties of this ontology are described with OWL primitives that enable us to perform inference on relationships and augment the social network with new relationships. While `foaf:knows` is not symmetrical, several properties of the RELATIONSHIP ontology are declared as symmetrical to represent reciprocal relationships, such as `rel:spouseOf`, `rel:friendOf` or `rel:worksWith`. Other relationships are declared as inverse properties such as `rel:mentorOf` and `rel:apprenticeTo`.

The combination of FOAF and RELATIONSHIPS offers a broad range of relationships that exist between persons. While these ontologies address the most common links between persons, many other relationships exist and can be represented as well by extending the basis of these two ontologies. The Figure 33 represents a small social network built with these two ontologies.

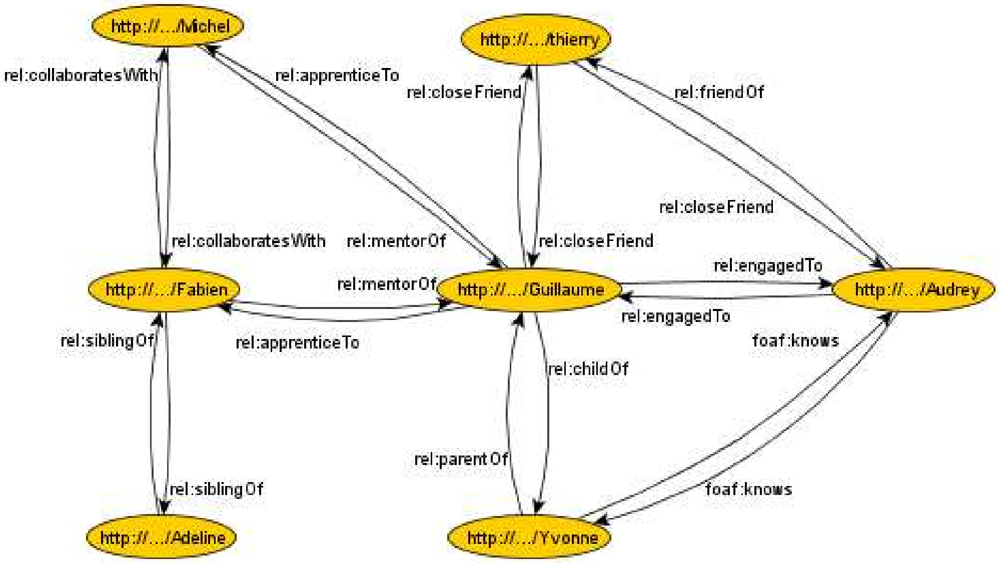


**Figure 33.** A toy social network built with the RELATIONSHIP and FOAF ontologies.

Typing and structuring relationships between persons offer a semantic dimension to the social network structure. The Figure 34 presents an example of a social network that is enhanced by a semantic schema, and more precisely in this case with a taxonomy of relationships.

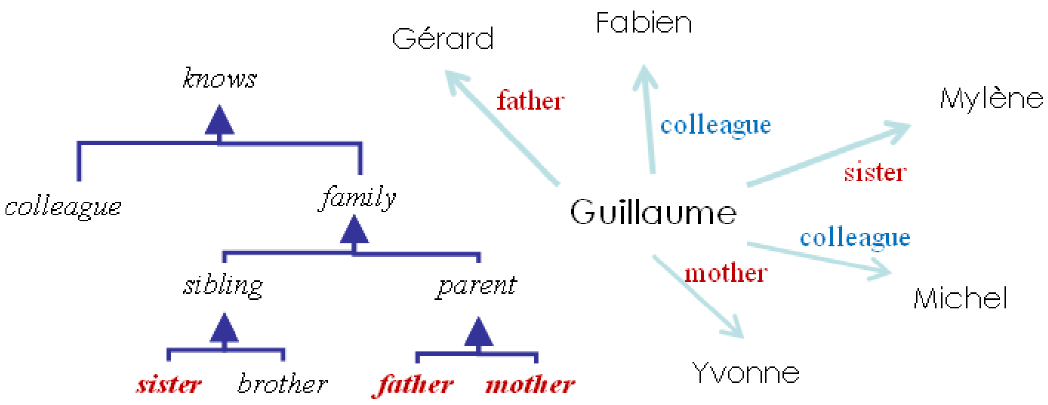


**Figure 34.** Typical social network represented with types relations and nodes.

Having the URL of the FOAF profile of a person that is given in parameter, the following query retrieves all its friends:

```
select ?friends where {
 param[url] rel:friendOf ?contact
}
```

If we query a relationship that is modelled with a property that has sub properties, the sub properties will be automatically considered during the mapping in triple stores having inference engines. For instance, in the relationship ontology, many properties are sub properties of `foaf:knows`. So when we ask for the `foaf:knows` neighbours of a person, we automatically focus on this sub network by considering, among others, the properties: `rel:friendOf`, `rel:parentOf` or `rel:worksWith`. In the previous query, if we parameterized the predicate of the only triple we obtain a reusable query to filter properties and focus on different sub networks:

```
select ?contact where {
 param[person] param[rel] ?contact
}
```

Here again, if we only have values of attributes of a FOAF profile with a unique combination of attributes identifying an agent, we can retrieve his neighbours in the network. Having the URL of an online account of a person, we can modify the previous query with this URL in parameter:

```
select ?contact where {
 ?person foaf:account param[url]
 ?person param[rel] ?contact
}
```

### 4.2.3 Representing and Linking User Accounts and Generated Content

The SIOC ontology provides the basis for describing social web sites and online communities [Breslin et al 2005]. In particular, SIOC defines primitives for describing a user, the content he produces and the actions of other users on this content. The SIOC ontology link a user account to the FOAF profile of the person that owns it, with the class `sioc:UserAccount` which is a sub class of `foaf:OnlineAccount`. Based on this SIOC provides a core of primitives to describe user generated content and user interactions on this content:

- The class `sioc:UserAccount` is used to type user accounts.
- The class `sioc:Item` is used to type published elements.
- The property `sioc:has_creator`, which domain is a `sioc:Item` and range is `sioc:UserAccount`, defines the publisher of an element.
- The property `sioc:has_reply` defines a response to an item. Its domain and its range are both of the type `sioc:Item`.
- The property `sioc:follows` describes that a user follows another user. Its domain and its range are both of the type `sioc:UserAccount`.
- The property `sioc:topic` describe the topic of a published item.

SIOC types, an extension of SIOC, provides different subclasses of the class `sioc:Item` in order to type more precisely the resources that are produced online. The class `sioc:Post`, `sioct:ImageGallery`, `sioct:Weblog` and `sioct:Comment` are respectively used to model posts, photo albums, blogs and comments on resources.

The SIOC primitives are sufficient to describe most of user generated content and user profiles across web 2.0 sites. The association of these SIOC primitives to FOAF profiles, thank to the property `foaf:account` and the class `foaf:OnlineAccount`, enables us to link the distributed activities and identities of an agent on the web. The Figure 35 is an illustration of the links between the data describing a person and its different user accounts.

Consequently we can query all the elements published by a person on its different user accounts with the URL of its FOAF profiles:

```
select ?item where {
 param[url] foaf:account ?userAccount
 ?item sioc:has_creator ?userAccount
}
```

Then we can modify the triples of this query and add others to perform any type of filtering and retrieve contributions with different perspectives. For instance we could ask only for the elements of the type given in parameter with their responses:

```
select ?item ?reply where {
 param[person] foaf:account ?userAccount
 ?item sioc:has_creator ?userAccount
 ?item rdf:type param[?type]
 optional{ ?item sioc:has_reply ?reply }
}
```

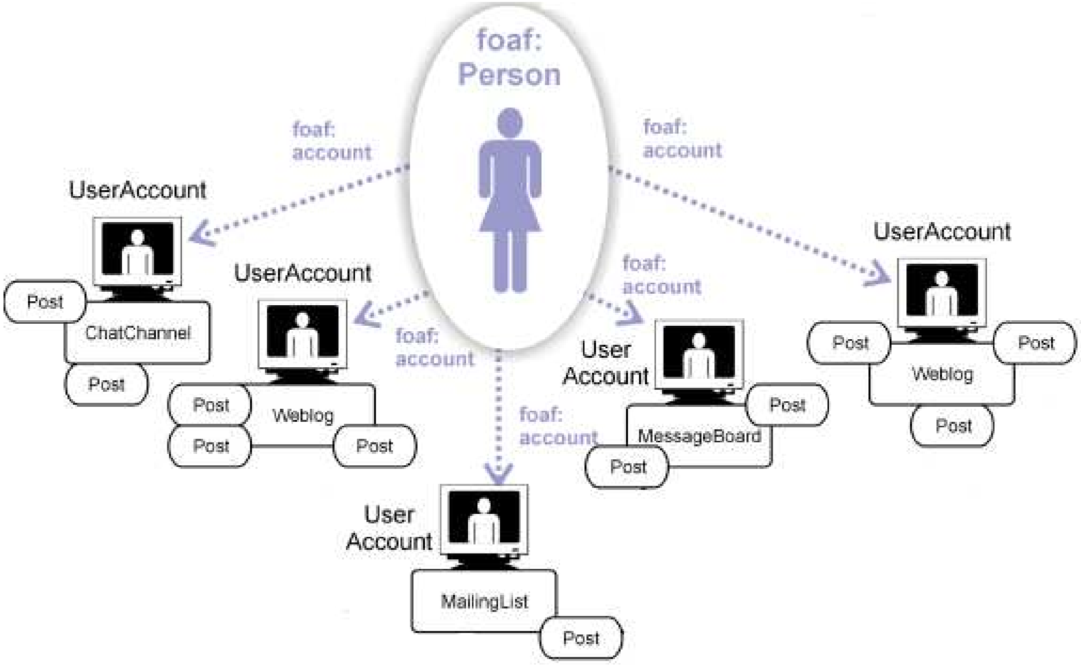


**Figure 35.** FOAF and SIOC link distributed identities and activities[29].

Using the combination of FOAF an SIOC, we are able to retrieve the activities of a whole network. A common perspective on the activities of a network is the timeline of a user, also named news feed that is proposed by most social web applications. This timeline proposes to a user different view of the content produced by its contacts.

The following query retrieves all the activities of the contacts of a person that are ordered by date of publications, with different parameters, the URL of its FOAF profile, the type of relationships and the type of item:

```
select ?contact ?item ?date where {
 param[url] param[rel] ?contact
 ?contact foaf:account ?userAccount
 ?item sioc:has_creator ?userAccount
 ?item rdf:type param[type]
 ?item sioc:created ?date
} order by ?date
```

In addition of linking agent profiles to their multiple user accounts and online activities, SIOC produces emergent interaction data which constitute valuable source of social network. First the property `sioc:follows` enables us to declare interests from a user for the activities and content produced by other users. This results in a social network of people that monitor the activities of each others. Then the property `sioc:has_reply` highlight mediated interaction that emerge in reaction of the content that is produced by users. The Figure 36 highlights how to represent online social network and user activities with FOAF, RELATIONSHIP and SIOC. At the top, we have the networking data produced by declared relationships. At the bottom, we have the networking data produced by the interactions mediated by generated content.

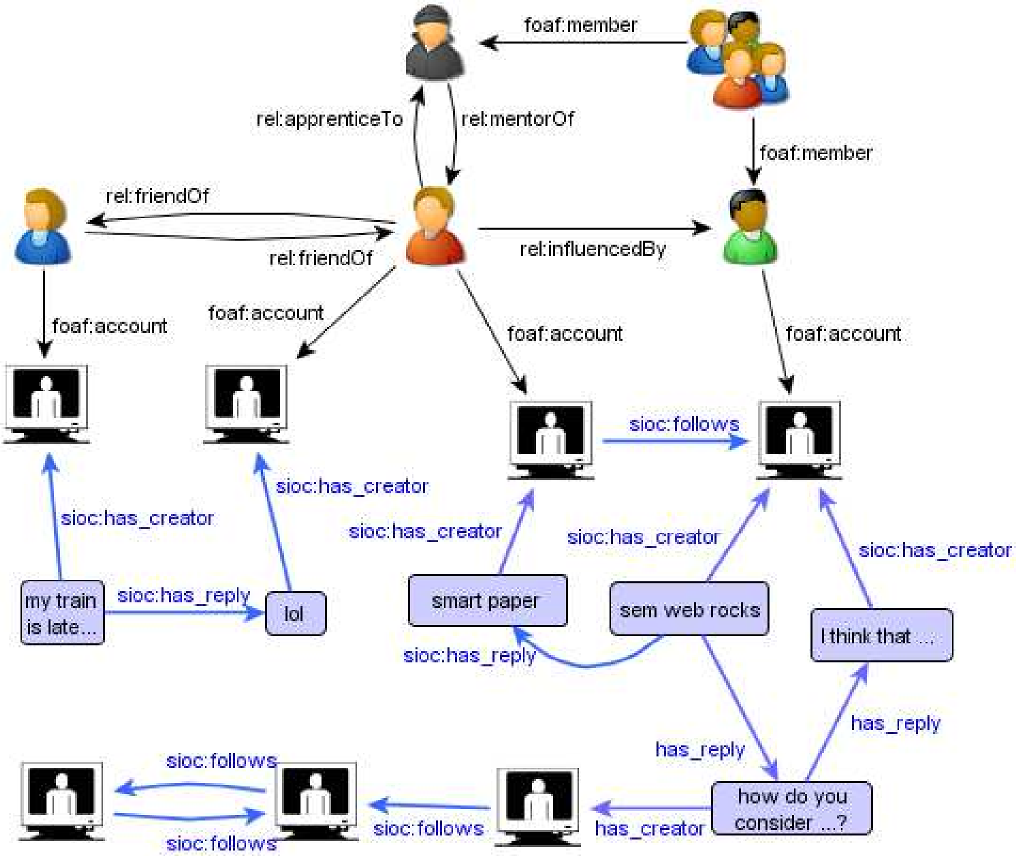


**Figure 36.** FOAF and RELATIONSHIPS enable us to describe direct and declared relationships while SIOC captures emergent interaction data.

Consequently we can modify the previous query by focusing on user interaction. For instance, the following query retrieves the activities of the users that are followed by the different user profiles of a person:

```
select ?contact ?item ?date where {
 param[url] foaf:account ?userAccount
 ?userAccount sioc:follows ?otherUserAccount
 ?item sioc:has_creator  ?otherUserAccount
 ?item sioc:created ?date
} order by ?date
```

### 4.2.4 Organizing and Structuring User Generated Content

While the SIOC ontology proposes a primitive to define the topic of published items, the SKOS ontology is a vocabulary for organizing concepts with lightweight semantic properties (e.g., `skos:narrower`, `skos:broader`, `skos:related`). This lightweight semantics can be used to structure and link the topics of SIOC descriptions. The Figure 37 illustrates how SKOS can be combined with FOAF and SIOC to structure the topics of user-generated content.

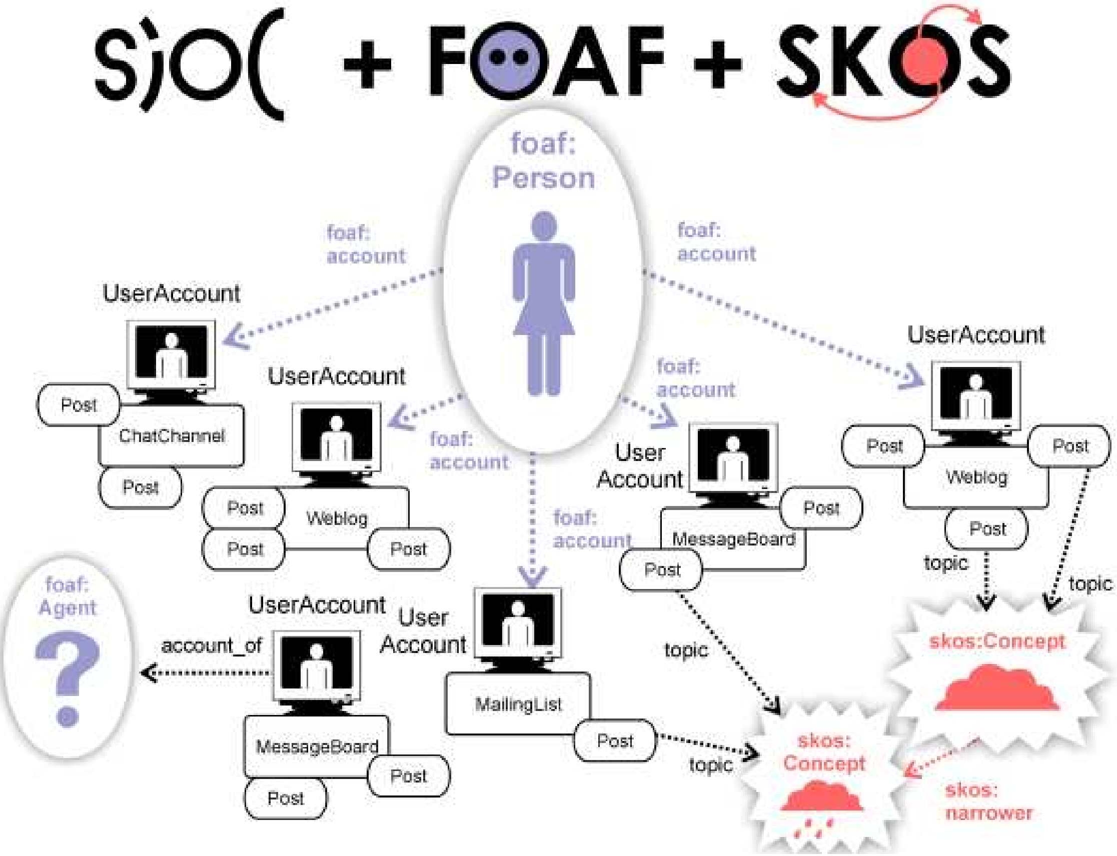


**Figure 37.** Alignment of FOAF, SIOC and SKOS[29].

Today, social tagging is the most popular practice to annotate and classify online resources. As explained in the previous chapter, social tagging consists in allowing users to associate freely chosen keywords, called tags, with the resources they publish and exchange such as blog posts, medias, or bookmarks. The result of the collection of such associations, called "taggings", is a *folksonomy*.

Tags in folksonomies remain at the stage of flat organization but folksonomies can be improved by adding semantics that structure and link tags together. First, successive ontologies were proposed to model social tagging such as [Gruber 2005] [Newman et al 2005] [Kim et al 2007]. These descriptions can deal with the author of the tag, or the tag itself as a character string, but also with additional properties such as the service where

[29] http://sioc-project.org/node/158

this tag is shared, or even a vote on the relevance of this tag. Then, since tags are neither explicitly structured nor semantically related, researchers proposed going further by linking tags with explanations of their meaning [Passant & Laublet 2008], or more generally, by bridging folksonomies, thesauri and ontologies to leverage the semantics of tags (see overview in [Limpens et al. 2008]). [Limpens et al 2010] proposes a methodology to structure tags with the lightweight semantic primitives of the SKOS vocabulary

The Figure 38, illustrates how, once semantically typed and structured, the relationships between topics and between topics and users also provide a new source of social networks. In fact social structures can be analyzed to type data produced by social actors and vice versa, data produced by social actors can be analyzed to type social networks. Consequently, the knowledge emerging from user interactions and affiliations can be used to link people, with the help of semantics (by identifying, for instance, communities that share same interests).

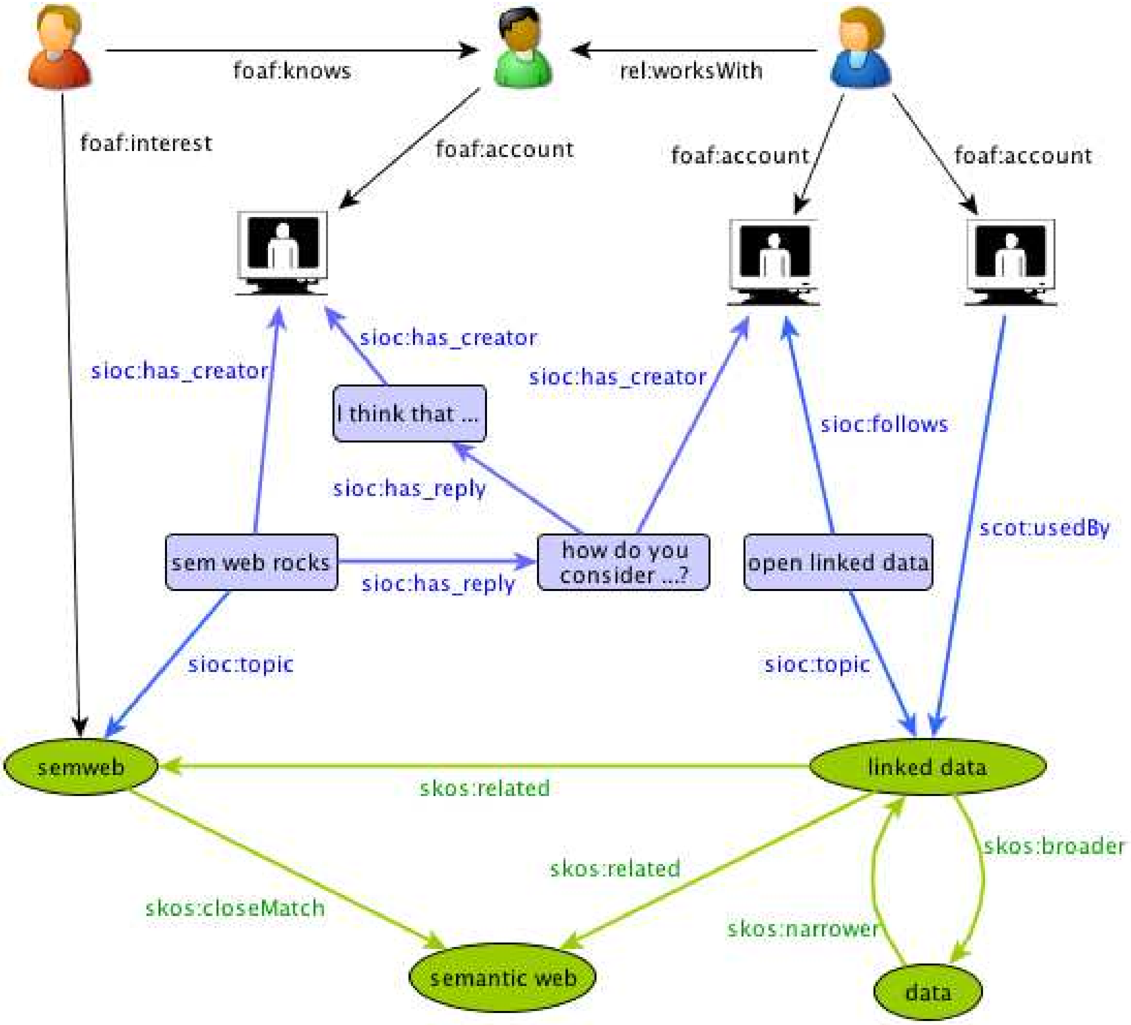


**Figure 38.** SKOS provides a new source of social network data by linking people with the knowledge that emerges through computer mediated interactions.

### 4.2.5 Analysis of Semantic Social Network Representation

Billion of triples are now published on the web and in particular social data with, among others, more than 60 Millions of FOAF profiles and more than 50 Millions of SIOC profiles[30]. This results in a massive amount of structured data that linked identities and activities across web applications.  This massive deployment was fostered by different initiatives. First, the FOAF ontology was soon adopted by web 2.0 platforms with large audiences to represent the profiles of their users with, for example, sites such as www.livejournal.net and www.tribe.net. Then, many exporting tools appeared for major web 2.0 services to export the data that are exposed by the mean of RDF API using the FOAF and SIOC schemas (e.g. FOAF exporter application for Facebook[31]). In addition, many plugins are now available for automatically exporting SIOC metadata from popular weblog engines (e.g. wordpress[32]), forums (e.g. phpBB[33]) and Content Management Systems (e.g. Drupal[34]). This constitutes a huge source of online SIOC metadata and valuable structuring and linking of emerging content. Finally, recent results proposed a decentralized single sign-on system, which combines signed FOAF profiles and SSL, for building decentralized social networks and give back to people the control of their information [Story et al 2009]. The progressive adoption of standardized ontologies for online social networks will lead to increasing interoperability between them and to the need for uniform tools to analyze and manage them.

While some researchers have applied classical SNA methods to these graphs for studying, mining, and extracting knowledge from the structure of these networks, others have extended SPARQL to leverage its expressivity and provide this RDF querying language with powerful graph operators.

A pioneering work by [Mika 2005] showed that folksonomies could be exploited using social network analysis in order to identify user groups and interest emergence. Moreover, the author exploited social network analysis on the hypergraph of folksonomies to infer lightweight semantics between terms. For instance, he observed that community detection enables us to find thematic communities (e.g travel, business, web design, etc.), and terms between these communities, with lowest clustering coefficient and highest betweenness centralities, are general terms (e.g. up, cool, hot, etc.). [Golbeck et al 2003] exploited the network structure of FOAF profiles linked by the property `foaf:knows`, for studying trust propagation in social networks using Semantic Web frameworks.  [Finin et al 2005] performed an analysis of a FOAF social network, using the property `foaf:knows`, to analyze the structure of such network and they verified the power law of the degrees and community structures in FOAF profiles.

[30] http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics

[31] http://www.facebook.com/apps/application.php?id=2626876931

[32] http://sioc-project.org/wordpress/

[33] http://sioc-project.org/phpbb

[34] http://sioc-project.org/drupal

In addition, they proposed rules to merge profiles representing same persons. [Goldbeck & Rothstein 2008] also worked on merging FOAF profiles, in order to unify identities used on different sites, while [Rowe & Ciravegna 2010] worked on disambiguating identities in identity web references. [Paolillo & Wright 2006] studied the acquaintance and interest networks respectively formed by the properties `foaf:knows` and `foaf:interest` properties of FOAF in order to identify communities of interest from the network of LiveJournal.com. [San Martin & Gutierrez 2009] presents how to use SPARQL to transform and enrich social networks, while using an export toward a raw graph format to use a classical social network analysis tool. In these studies, due to some limits of Semantic Web technologies, authors chose to build their own non typed graphs, each corresponding to a type of link from the richer RDF social graphs (e.g. [Paolillo & Wright 2006] has built 2 non typed graphs from the social network of FOAF profiles, one for the property `foaf:knows` and another one for the property `foaf:interest`). Unfortunately, knowledge is lost in this transformation while it could be used to parameterize social network indicators, improve their relevance and accuracy, filter their sources and customize their results.

In fact, SPARQL has been shown to be "equivalent from an expressiveness point of view to Relational Algebra" [Angles & Gutierrez 2008]. While SPARQL is well suited for retrieving and building rich semantics of a social network, it cannot perform global queries similar to the ones currently used in social network analysis (e.g., compute the diameter, the density, the centrality) [San Martin & Gutierrez 2009]. Consequently, researchers have tackled the expressiveness of SPARQL and, in particular, they extended it in order to find property paths between semantically linked resources in RDF-based graphs [Alkhateeb et al 2007] [Anyanwu et al 2007] [Corby 2008] [Pérez et al 2008] [Kochut & Janik 2007]. At the time of this writing, SPARQL 1.1 is being discussed[35] and the current version of the working draft[36] proposes operators to handle the querying of property paths.

## 4.3 Conclusion

In the first part of this chapter, we presented how Semantic Web technologies answer the problem of richly representing, exchanging, and querying online data. The RDF language provides the base primitives to perform triple descriptions of online resources, with URI and properties as the core of any assertion. RDF triple descriptions, `(subject, property, object)`, form directed labelled graph. Ontologies, described in RDFS or OWL, enable to structure and constrain the concept used to label RDF graph. They enable (1) to turn the labelling of RDF graph into a rich typing, and (2) to reason on this graph and enrich it. This rich typed graph can be then accessed, queried and transformed with SPARQL.

[35] At the time of this writing, the standard version is Sparql 1.0

[36] http://www.w3.org/TR/2010/WD-sparql11-query-20100601

In the second part of this chapter, we presented ontologies that were built to define, structure and link vocabularies of online social graphs. FOAF is a base vocabulary for describing people, their attributes, their acquaintances and their online account. The RELATIONSHIP and SIOC ontologies extend FOAF in order to precisely describe relationships between people and their activities on social web sites. SCOT and others ontologies propose vocabularies of social tagging, and tags as well as topics of web publications can be then semantically structures with the lightweight semantic primitives of SKOS. Analyses that have been conducted on the semantic representations did note exploit the rich semantics they contain, while it could have been used to refine their results.

This whole stack of tools provides the basis to richly represent, linked and query online social graph. We now dispose of a directed typed graph model to represent, link and query online social graphs across web sites. We have seen in chapter 3, that graph theory offers many algorithm and metrics to analyse the link structure of raw graph. We will see in the next chapter how we can merge this approaches to conduct a social network analysis that takes benefits of the ontological primitives use to type and structure online social graphs.

# 5. Analysis of Semantic Representation of Social Networks

On one side, social network analysis provides efficient operators for mining the flat graph structure of networks and for extracting their structural characteristics. However they do not leverage, for instance, the types of the links, the different profiles or the roles of the different actors. On the other side, Semantic Web technologies enable us to connect distributed social data on the web and to richly type both nodes and relationships of social graphs with classes and properties of domain ontology. However, while Semantic Web languages are efficient for representing, exchanging and extracting rich social data, they still lacks of graph operators for meeting SNA requirement, like global metric querying.

In this chapter, we address this challenge and propose a model that extends SNA operators using Semantic Web frameworks in order to include the semantics of these graph-based representations in the analysis. This makes it possible to take benefits of the diversity of the relationships and interactions.

We first present the stack of tools we designed for conducting the semantic social network analysis. Then we present the results of this approach on a real social network with 60,000 users connecting, interacting and sharing content.

## 5.1 A Semantic Web Framework for Social Network analysis

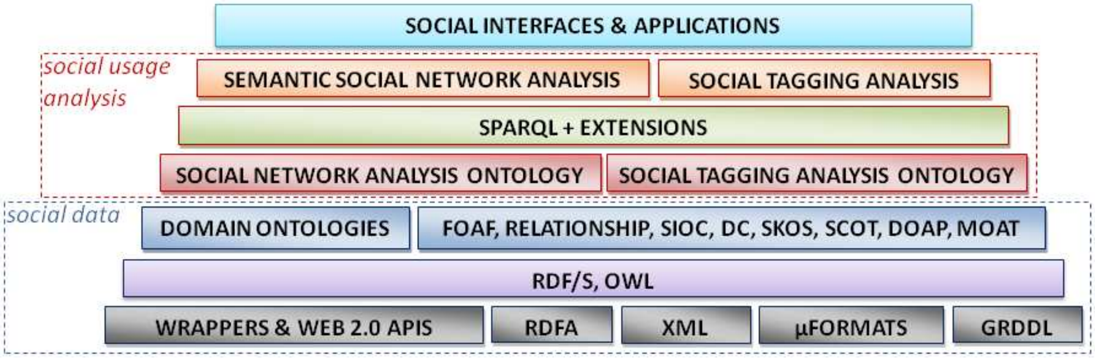

SOCIAL INTERFACES & APPLICATIONS
social usage analysis
SEMANTIC SOCIAL NETWORK ANALYSIS
SOCIAL TAGGING ANALYSIS
SPARQL + EXTENSIONS
SOCIAL NETWORK ANALYSIS ONTOLOGY
SOCIAL TAGGING ANALYSIS ONTOLOGY
social data
DOMAIN ONTOLOGIES
FOAF, RELATIONSHIP, SIOC, DC, SKOS, SCOT, DOAP, MOAT
RDF/S, OWL
WRAPPERS & WEB 2.0 APIS
RDFA
XML
µFORMATS
GRDDL

**Figure 39.** Abstraction Stack for Semantic Social Network Analysis.

The Figure 39 presents the stack of tools that we designed to conduct a semantic social network analysis. The goal of this stack is to provide a framework that enables us to consider not only the network structure embedded in social data, but also the schemas that are used to structure, link and exchange these data. This stack is composed of (1) tools for building, representing and exchanging social graphs and (2) tools for extracting social network analysis metrics and leveraging social graphs with their characteristics.

We represent the social graphs in RDF, which provides a directed typed graph structure. We showed in the chapter 4 that RDF is well suited for representing social graphs and that this format eases the sharing and the interoperability of social data between applications. Then we leverage the typing of nodes and edges with the primitives of existing ontologies together with specific domain ontologies if needed. With this rich typing, semantic engines are able to perform type inferences from data schemas for automatically enriching the graph and checking its consistency. For increasing interoperability, we recommend to use (and extend if needed) the ontologies of the linked data cloud presented in the chapter 4. Doing so will ease the discovery of related knowledge and the enrichment of the graph. In particular we detailed in the chapter 4, ontologies for representing social data on the web. We use the FOAF ontology for describing people, their relationships and their activity. The properties defined in the RELATIONSHIP ontology enable us to type more precisely relationships between persons (e.g. the relation `rel:livesWith` specializes the relation `foaf:knows`). We use the primitives of the SIOC ontology for modelling online user accounts, the content they publish and corresponding mediated interactions. Finally SCOT enables us to describe social tagging activities while SKOS offers a way to organize tags and topics of emerging content with lightweight semantic properties (e.g. `skos:narrower, skos:closeMatch, skos:related`).

Some social data are already readily available in a semantic format (e.g. RDF, RDFa) and can be used straightforward. Data provided in RDF can be directly queried in SPARQL. RDFa and µformats are mark-ups embedded in XHTML web pages for defining the semantics of some elements of the pages and have to be extracted and transformed in RDF before being queried. Algorithms for extracting this data from documents are available and GRDDL[37] defines how to link a document to these algorithms. However, today, most of the data are still only accessible through APIs (e.g. Flickr, Facebook, Twitter) that format their response in XML or JSON using different schemas. Other data are also retrieved by crawling web pages and need to be converted. Consequently crawled data and web 2.0 APIs have to be wrapped in order to be structured in RDF and to be interoperable and mashable with any data.

We propose to leverage social data with the results of their analyses. [Limpens et al 2010] designed a methodology to enrich social tagging data with semantics with a combination of human contributions and algorithms for analysing tags' labels and folksonomy structures. On our side, we designed SemSNA to enrich social data with the characteristics of the semantic social graph they form. This ontology that is detailed in section 5.1.3 defines different SNA metrics ranging from the annotation of strategic positions and strategic actors, to the description of the structure of the network. In addition, SemSNA enables us to qualify SNA metrics with the sub graph and the semantic of properties that were analyzed. With this ontology, we can (1) abstract social network constructs from domain ontologies to apply our tools on existing schemas by having them extend our primitives; and we can (2) enrich the social data with the SNA metrics that are computed on the network. These annotations enable us to manage more efficiently the life cycle of an analysis, by pre-calculating relevant SNA indices and updating them incrementally when the network changes over time. Moreover they can be used during the querying of social data for ordering and filtering the results.

On top of SemSNA we propose SPARQL formal definitions of SNA operators handling the typing of the semantic representations of social networks. Building on top of results on graph-based representation and reasoning for RDF/S and OWL [Corby et al 2004] [Baget et al 2008] [Corby 2008], we exploit the RDF graph representations of social networks with qualified queries. This querying, qualified by the typing of the graph, focuses automatically on specific path patterns, involving specific resource or property types. The SPARQL queries that we designed are based on the powerful extensions of the standard SPARQL language that are implemented in the semantic graph engine CORESE [Corby et al 2004]. In particular, the property path extension of CORESE [Corby 2008] enables us *to extract paths in RDF graphs* by specifying multiple criteria such as the type of the properties involved in the path with regular expressions, or edge directions or constraints on the vertices that paths go through. These extensions leverage the expressivity of CORESE, and we present in section 5.1.2 how it extends the definition of Entity Graphs and their mapping.

In this section we will see, (1) how to wrap social data with CORESE, (2) how SPARQL can be used to query RDF social data with different perspectives, and (3) the description of SemSNA and how it can be used to annotate social networks.

[37] http://www.w3.org/2004/01/rdxh/spec

### 5.1.1 Wrapping Social Data with CORESE

CORESE proposes SPARQL extensions that enable us to query heterogeneous sources of data and a rule engine for enriching RDF graphs [Corby et al 2009]. CORESE proposes functions to nest SQL and XPATH queries in the SELECT clause of a SPARQL query and to bind their variable with SPARQL variables. Consequently it offers to mash different sources of data of different formats with a single query. On top of this, CORESE allows us to combine a CONSTRUCT block with a SELECT block and reuse, in the construction of new triples, the resources that were computed by a function. Finally, once the resources are in the RDF format, they can be automatically enriched and transformed with rules, using the rule engine of CORESE.

#### 5.1.1.1 Querying XML and Relational Databases with CORESE

Corese has an extension that enables us to nest SQL and XPATH queries within SPARQL queries [Corby et al 2009]. This feature is useful for querying XML and relational databases and for designing wrappers for non RDF data.

CORESE offers to nest any function in a SPARQL query. In particular, the querying of databases is done by means of the `sql()` function that returns a sequence of results for each variable in the SQL select clause. The parameters of this function are the URL of the database server, the class of the `jdbc` driver, the login and the password for opening a session, and the SQL query. The results of this function, namely the SQL variables with their matched values, are bound to SPARQL variables. Similarly the querying of an XML source is done by means of an `xpath()` function, whose parameters are the URL of the XML source and the `xpath` expression to match. Here again, the results of this function, namely the elements of the queried XML document that match the `xpath` expression, are bound to SPARQL variables.

In the following example, we show how we retrieve the *friend* relationships from a relational database, having a table of relations with two columns, one for each actor of the relation. We use this `sql()` function and another one, `genIdUrl()`, that generates URIs from relational database primary keys:

```
construct { ?person1 rel:friendOf ?person2 }
select sql('jdbc:mysql://…/mysql_server',
 'com.mysql.jdbc.Driver', 'user', 'pwd',
 'SELECT user1_id, user2_id from relations) as (?id1,
 ?id2) fun:genIdUrl(?id1, 'http://…/people/') as ?person1
 fun:genIdUrl(?id2, 'http://…/people/') as ?person2
where {  }
```

The twitter API offers to query its social networks, and in particular to retrieve the followers of a user with its user name and the following URL:

```
http://api.twitter.com/1/followers/ids.xml?screen_name=username
```

For instance, to obtain all the followers of my twitter account in the XML, format, one should call the following URL:

```
 http://api.twitter.com/1/followers/ids.xml?screen_name=ereteog
```

The resulting XML document is a list of ids:

```
<?xml version="1.0" encoding="UTF-8"?>
<id_list>
 <ids>
   <id>161965514</id>
   <id>113573751</id>
   <id>55826821</id>
   (…)
 </ids>
</id_list>
```

The following query enables us to nest an `xpath` function that queries this document:

```
construct {
 ?follower sioc:follows <http://twitter.com/ereteog>
}
select
 xpath(<http://.../followers/ids.xml?screen_name=ereteog>
  , "/ids/id")) as ?userId
   fun:genIdUrl(?userId, 'http://twitter.com/')
 as ?follower
where { }
```

These CORESE extensions offer a simple way to construct RDF on top of frequently used formats for representing social data.

#### 5.1.1.2 Rules and Enrichment of RDF Social Network

Semantic Web frameworks offer different ways to enrich RDF data with reasoning mechanisms. In the chapter 4, we presented how ontologies that are formalized in RDFS and OWL are used by semantic graph engine to enrich RDF with new inferred triples, using the positive constraints that are expressed on classes and properties. However, other processing can also enrich the semantics of RDF graphs, such as rules crawling the network to add types or relations whenever they detect a pattern. In CORESE, once exported in RDF, a social network can be processed to leverage the semantics of its graph representation with (1) SPARQL queries using a `construct` block and (2) inference rules on the graphs.

[San Martin & Gutierrez 2009] have shown how to enrich a social network with a `construct` block. When you execute a `construct` SPARQL query as a rule to enrich your network, the `where` block will be the pattern to match and the `construct` block will be the conclusion. In fact using a `construct` SPARQL query as a rule could be viewed as a *forward chaining rule inference*.

For instance, if a user comments a post from another user, we could deduce that the two persons that own these accounts have acquaintances:

```
construct{ ?person2 rel:acquaintance ?person1}
where {
 ?person1 foaf:account ?user1
 ?person2 foaf:account ?user2
 ?user1 foaf:creator ?post
 ?post sioc:has_reply ?comment
 ?user2 foaf:creator ?comment
}
```

Such queries produce RDF triples in respect with the `construct` block, which can be stored next. However, Corese enables us to re-inject the knowledge produced directly into the knowledge base with an `add` clause, which replaces the `construct` clause:

```
add{ ?person2 rel:acquaintance ?person1}
where {
 ?person1 foaf:account ?user1
 ?person2 foaf:account ?user2
 ?user1 foaf:creator ?post
 ?post sioc:has_reply ?comment
 ?user2 foaf:creator ?comment
}
```

Similarly, some transformations can be automated by inference rules that are performed on a rule engine. Here, it consists in a *forward chaining rule inference*. We have first a condition block that describes a pattern to match and then a conclusion block that defines the triples to generate whenever the condition pattern is matched.

CORESE has a proprietary rule format that propose such *forward chaining rule inference*, and the previous example can be as well performed with the following CORESE rule:

```
<cos:if>
 { ?person1 foaf:account ?user1
   ?person2 foaf:account ?user2
   ?user1 foaf:creator ?post
   ?post sioc:has_reply ?comment
   ?user2 foaf:creator ?comment }
</cos:if>
<cos:then>
 { ?person2 rel:acquaintance ?person1 }
</cos:then>
```

The preceding syntax is specific to Corese but the Rule Interchange Format[38] (RIF) proposes XML dialects for expressing and exchanging rules on the Web, and providing interoperability between the different rule engines. These dialects include in particular SWX that proposes an interoperability of RIF with RDF, RDFS and OWL.

### 5.1.2 Extract SNA Concepts with SPARQL

The web evolves very quickly, new social platforms with new features and usages frequently appear with new forms of social exchanges; Semantic Web technologies are designed to handle such evolutions. Querying social data with SPARQL eases the evolution of social platform models and the integration of new ones as long as they are expressed with ontologies. We present here how to extract social network analysis metrics by combining structural and semantic characteristics of the network with queries performed with Corese.

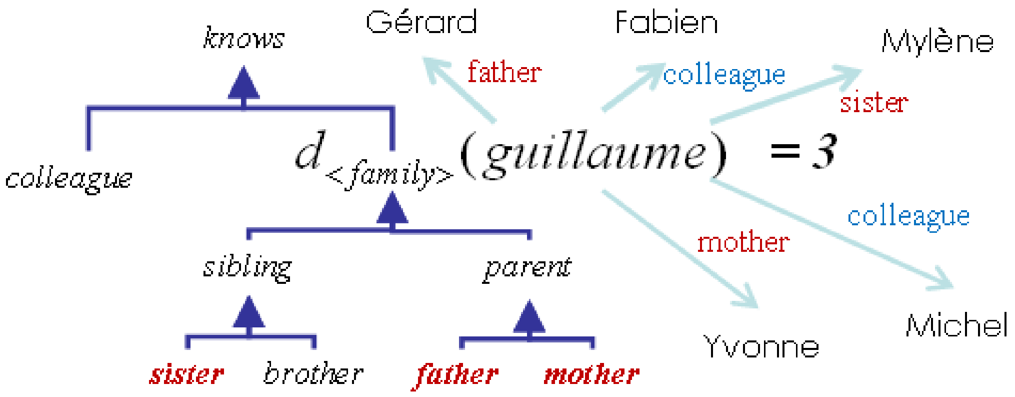
In [San Martin & Gutierrez 2009], researchers have shown that SPARQL "is expressive enough to make all sensible transformations of networks". However, this work also shows that SPARQL is not expressive enough to meet SNA requirements for global metric querying (density, betweenness centrality, etc.) of social networks. Such global queries are mostly based on result aggregation and path computation, which are missing from the standard SPARQL definition. Extensions are currently discussed to provide SPARQL 1.1 with operators that fit this void. However, the Corese search engine [Corby et al 2004] already provides such features with result grouping, aggregating functions (e.g. `sum()` or `avg()`) and path retrieving [Corby 2008].

#### 5.1.2.1 Exploit Rich Typing of Relationships

Corese implements SPARQL with the Homomorphism<x>, defined in chapter 4, which takes into account the semantic of schemas for mapping the graph of a SPARQL query on an RDF graph. Consequently, when querying an RDF social graph that is richly typed with ontological primitives, we are able to focus on different types of relationships while taking into account the sub networks implied by the defined

[38] RIF Working Group http://www.w3.org/2005/rules/wiki/RIF_Working_Group

patterns. Implementing SNA operators with SPARQL offers to directly work on RDF graphs and automatically consider it semantics without producing any other graph in another format. SNA operators become qualified by the properties that represent the analyzed relationships. The Figure 40 illustrates the computation of a qualified degree where only family relations are considered by exploiting the hierarchy of relationships.



$$d_{<R>}(p) = \left\| \{x ; \exists(p, R, x)\} \cup \{x ; \exists(x, R, p)\} \right\|$$

**Figure 40.** A parameterized degree that considers a hierarchy of relationships.

#### 5.1.2.2 Extract Complex Relationships with Property Paths

Paths in graph represent how nodes are interconnected and, in a social network, how resources are interacting. Providing SPARQL with property path operators offers to detect rich relationships among the diversity of online interactions (e.g. two users annotated a same resource with a same concept). The principle of the property path extension consists in searching a sequence of properties, which match a regular expression, between two resources in the RDF graph. For this we extend the definitions of ERGraph (Definition 53 in chapter 4) to include paths in the graphs and mappings:

**Definition 59. PERGraph:** A PERGraph relative to a set of labels L is a 4-tuple $G=(E_G, R_G, P_G, n_G, l_G)$ where:

- *$G'=(E_G, R_G, n_G, l_G)$ is an ERGraph*
- $P_G$ is a disjoint set from $E_G$ and $R_G$ of hyperedges called paths.
- $n_G : R_G \cup P_G \rightarrow E_G^*$ associates to each relation and path a finite tuple of entities called the arguments of the relation or the path.
- $l_G : E_G \cup R_G \cup P_G \rightarrow L$ is a labelling function of entities, relations and paths.

**Definition 60. PERMapping$_{<X>}$:** Let $G$ and $H$ be two PERGraphs, and X be a binary relation over L×L. A PERMapping from $H$ to $G$ is an ERMapping$_{<X>}$ $M$ from $H$ to $G$ such that: Let $H'$ be the SubERGraph of $H$ induced by $M^{1}(E_G)$, $\forall p' \in P_{H'}\ \exists p= (r_1,\ldots, r_n) \in R_G^n$ such that:

- $\forall\, 1 \leq i \leq card(n_G(r)),\ M(n_{H'}^{\ i}(r')) \in n_G(r_1) \cup n_G(r_n)$

- $\forall\, 1 \leq i \leq n\ (l_G(r_i),\ l_H(p')) \in X.$

To exploit this extension of the model, we introduced a syntactic convention in Corese for path extraction. A regular expression is used instead of the property variable to specify that a path is searched and to describe its characteristics. Corese implements a subset of the property path functionalities designed in SPARQL 1.1[39] as follow:

| Syntax Form | Matches |
|---|---|
| `Uri` | A URI or a prefixed name. A path of length one. |
| `(elt)` | A group path `elt`, brackets control precedence. |
| `elt1 / elt2` | A sequence path of `elt1`, followed by `elt2` |
| `elt1 \| elt2` | A alternative path of `elt1`, or `elt2` (all possibilities are tried). |
| `elt*` | A path of zero or more occurrences of `elt`. |
| `elt+` | A path of one or more occurrences of `elt`. |
| `elt?` | A path of zero or one `elt`. |

**Figure 41.** Elements and operators of the regular expression of a property path defined in SPARQL 1.1 Property Paths working draft[39] of 2010-01-26: “`uri` is either a URI or a prefixed name and `elt` is a path element, which may itself be composed of path syntax constructs”.

Corese proposes a supplementary operator, `!` (not), to exclude an expression from the matched path. The following example proposes a regular expression to retrieve a path between two resources `?x` and `?y` starting with zero or more `foaf:knows` properties and ending with the `rel:worksWith` property:

```
?x foaf:knows*/rel:worksWith::$path ?y
```

As Corese supports RDFS entailment, it implements the $PERMapping_{<X>}$ and thus takes into account sub-properties of the properties of the regular expression, unless specified otherwise.

Corese proposes additional features for retrieving paths. First it proposes to restrict the retrieved path with characteristics that are defined by adding options before the regular expression: `'s'` to retrieve one shortest path and `'sa'` to retrieve all shortest paths. For instance, the following query asks for only one shortest path between two resources `?x` and `?y` starting with zero or more `foaf:knows` properties and ending with the `rel:worksWith` property:

```
?x s foaf:knows*/worksWith ?y
```

Then, Corese enables us to bind a path with a variable specified after the regular expression. This variable can be reused as input of a defined function. In particular Corese proposes the function `pathLenght()` to compute the length of a path. The

[39] SPARQL 1.1 Property Paths http://www.w3.org/TR/2010/WD-sparql11-property-paths-20100126/

previous example is restricted in the following example to paths of length equal to or less than 3:

```
?x foaf:knows*/rel:worksWith::$path ?y
filter(pathLength($path) <= 3)
```

Finally, an extension enables us to enumerate and constrain the triples belonging to the path that has been found, by exploiting it as a graph. It uses the `graph` clause where the graph variable is a path variable. The example below enumerates the `foaf:knows` triples of the path and imposes that a least one of its resources *knows* Michel:

```
graph $path { ?x foaf:knows ?y }
?x foaf:knows <http://www.i3s.unice.fr/Michel>
```

Path retrieval enables us to exploit the hierarchy of properties, by taking into account sub-properties at each step. Consequently we compute SNA metrics with parameterized queries that accept as argument a regular expression of properties.

#### 5.1.2.3 Global Querying and Aggregating Operators

CORESE SPARQL extensions propose operators to group projections of a query on an RDF graph and to compute and extract not only patterns but also global characteristics of graphs.

The `group by` clause enables us to group results having the same values for a given list of variables (this option is also implemented in most existing RDF engines and defined in SPARQL 1.1). The priority of the grouping is performed in respect with the order of the variables in the given list. For instance the following query returns for each person the group of persons he knows:

```
select ?p1 ?p2 where {
   ?p1 ?foaf:knows ?p2
} group by ?p1
```

Then aggregating functions compute additional results from groups of projections. By definition they are used together with the `group by` clause. For instance for a group of results:

- the `count()` function counts the occurrences of matched results for a given variable
- the `sum()` function sums the matched numerical values for a given variable
- the `avg()` function compute the average value of the matched numerical values for a given variable.

The following query returns for each person the number of persons he knows:

```
select ?p1 count(?p2) as ?nbwhere {
   ?p1 ?foaf:knows ?p2
} group by ?p1
```

CORESE extends these sets of operators with additional ones. In particular, it extends the expressivity of result grouping. The keyword any offers to group results having the same value for any result variables. In the table of the Figure 43 this operator is used to retrieve components, namely connected sub graphs, with the following query:

```
select ?x ?y where {
 ?x param[rel] ?y
}group by any
```

The merge keyword merges all the projections in one single result with distinct values for each variable. It is an important feature for retrieving global metrics on the graph when no grouping criterion is possible between projections. For instance, in the table of the Figure 43, the following query computes the number of actors involved as subject of a given relationship:

```
select merge count(?x) as ?nbsubj where{
 ?x param[rel] ?y
}
```

#### 5.1.2.4 SPARQL Operationalization of Parameterized SNA Metrics

Based on the enhanced SPARQL language of CORESE, we propose a set of queries (Figure 43) to compute SNA metrics adapted to the directed labelled graphs described in RDF (Figure 42). These queries exploit different SPARQL extension, including property paths, grouping and aggregating functions. We implemented and tested all the presented operators.

In some cases, the implementation of complex SNA algorithms will require post-processing to deal with (1) remaining lacks of expressivity of our extended SPARQL version, and (2) performances issues of a complete delegation of the process to a semantic graph engine. Consequently, in these cases, we use SPARQL for any semantic treatments, for extracting relevant parts of the graph, and for pre-processing different steps of algorithms. Then we iterate on the results to compute the final centrality values.

The betweenness centrality is an example of algorithm that cannot be computed directly with a SPARQL query because it requires different processing steps on the graph. We use the CORESE engine to retrieve shortest property paths between resources of the graph and intermediary resources. Consequently the semantic processing are still delegated to the semantic engine that will takes into account the sub properties of the properties used in the regular expression for characterizing the searched paths. Then the definition of the betweenness centrality is computed with a post process on the results of the query.

| SNA indices and definition | Notation |
|---|---|
| <u>Graph:</u> defined in chapter 4. | $G=(E_G, R_G, n_G, l_G)$ where :<br>$E_G$ and $R_G$ are two disjoint finite sets respectively, of nodes and relations.<br>$n_G : R_G \rightarrow E_G^{*}$ associates to each relation a couple of entities called the arguments of the relation. If $n_G(r)=(e_1,e_2)$ we note $n_G^{i}(r)=e_i$ the $i^{th}$ argument of $r$.<br>$l_G : E_G \cup R_G \rightarrow L$ is a labelling function of entities and relations. |
| <u>Number of actors:</u> the number of actors of a given type (or subtype). | $nb^{actor}_{< type >}(G)=\lvert e\in E_G; l_G(e)\le type\rvert$ |
| <u>Number of actors:</u> the number of actors involved in a given relation of type *rel* (or subtype) as subject or object. | $nb^{actor}_{< rel >}(G)=\left\lvert\left\{\begin{array}{l} e\in E_G;\\ \exists r\in R_G, e'\in E_G, l_G(r)\le rel,\\ n_G(r)=(e,e') \Vert n_G(r)=(e',e)\end{array}\right\}\right\rvert$ |
| <u>Number of subject actors:</u> the number of actors involved in a given relation of type *rel* (or subtype) as subject. | $nb^{subject}_{< rel >}(G)=\left\lvert\left\{\begin{array}{l} e\in E_G;\\ \exists r\in R_G, e'\in E_G, l_G(r)\le rel,\\ n_G(r)=(e,e')\end{array}\right\}\right\rvert$ |
| <u>Number of object actors:</u> the number of actors involved in a given relation of type *rel* (or subtype) as object. | $nb^{object}_{< rel >}(G)=\left\lvert\left\{\begin{array}{l} e\in E_G;\\ \exists r\in R_G, e'\in E_G, l_G(r)\le rel,\\ n_G(r)=(e',e)\end{array}\right\}\right\rvert$ |
| <u>Number of relations:</u> number of pairs of resources linked by a property of type *rel* (or subtype). | $nb^{relation}_{< rel >}(G)=\left\lvert\left\{\begin{array}{l} (e_1,e_2)\in E_G^{2};\\ \exists r\in R_G, l_G(r)\le rel,\\ n_G(r)=(e_1,e_2)\end{array}\right\}\right\rvert$ |

| | |
|---|---|
| Path: a list of nodes of a graph *G* each linked to the next by a property of type *rel* (or subtype). | $p_{<rel>} = <x_0, x_1, x_2 .. x_n>$<br>$\wedge \forall i < n \exists r \in R_G;$<br>$n_G(r) = (x_i, x_{i+1}) \wedge l_G(r) \leq rel$ |
| Length: the number of relations involved in a path. | $length(\, p = < x_0, x_1, x_2 ... x_n >) = n$ |
| Density: proportion of the maximum possible number of properties of type *rel* (or subtype). | $Den_{<rel>}(G) = \dfrac{nb^{relation}_{<rel>}(G)}{nb^{actor}_{<domain(rel)>}(G) * nb^{actor}_{<range(rel)>}(G)}$ |
| Component: a connected sub graph for a given property *rel* (and sub-properties) with no link to resources outside the component | $Comp_{<rel>}(G) = (G_1, G_2, ..., G_k)$<br>where $G_k$ is a subgraph of G such that for every pair of nodes $n_i$, $n_j$ of $G_k$ there exist a path $p_{<rel>}$ from $n_i$ to $n_j$ in *G*. |
| Degree: number of paths of properties of type *rel* (or subtype) having *y* at one end and with a length smaller or equal to *dist*. It highlights local popularities. | $D_{<rel,dist>}(\, y) = \left\| \begin{Bmatrix} x_n ; \exists path\ p_{<rel>} = < x_0, x_1, x_2 ... x_n > \\ \wedge ((x_0 = y) \vee (x_n = y)) \wedge n \leq dist \end{Bmatrix} \right\|$ |
| In-Degree: number of paths of properties of type *rel* (or subtype) ending by *y* and with a length smaller or equal to *dist*. It highlights supported resources. | $D^{in}_{<rel,dist>}(\, y) = \left\| \begin{Bmatrix} x_n ; \exists path\ p_{<rel>} = < x_0, x_1, x_2 ... x_n > \\ \wedge (x_n = y) \wedge n \leq dist \end{Bmatrix} \right\|$ |
| Out-Degree: number of paths of properties of type *rel* (or subtype) starting by *y* and with a length smaller or equal to *dist*. | $D^{out}_{<rel,dist>}(\, y) = \left\| \begin{Bmatrix} x_n ; \exists path\ p_{<rel>} = < x_0, x_1, x_2 ... x_n > \\ \wedge (x_0 = y) \wedge n \leq dist \end{Bmatrix} \right\|$ |
| Geodesic between *from* and *to*: a geodesic is a shortest path between two resources for a properties of type *rel* (or subtype). | $g_{<rel>}(from, to) = < from, x_1, x_2 ..., x_n, to >$<br>such that:<br>$\forall p_{<rel>} = < from, y_1, y_2 ..., y_m, to >$ n≤m |
| Diameter: the length of le longest geodesic in the | $Diam_{<rel>}(G) = length\,(g_{<rel>}(e_1, e_2))$ |

| | |
|---|---|
| network for a property of type *rel* (or subtype). | Such that<br>$\forall e_3 , e_4 \in E_G ;$<br>$length(g_{<rel>}(e_3,e_4)) \le length(g_{<rel>}(e_1,e_2))$ |
| Number of geodesics between *from* and *to*: the number of geodesics between two resources for a properties of type *rel* (or subtype). | $nb^{g}_{<rel>}(from,to) = \left\|\{g_{<rel>}(from,to)\}\right\|$ |
| Number of geodesics between *from* and *to* going through node *b*: the number of geodesics between two resources going through a given intermediary node for a property of type *rel* (or subtype). | $nb^{g}_{<rel>}(b, from,to) = \left\|g_{<rel>}(from,to)\right\|$<br>Such that :<br>$g_{<rel>}(from,to) =< from,...,b,...,to >$ |
| Closeness centrality: the inverse sum of shortest distances to each other resource for a property of type *rel* (or subtype). | $C^{c}_{<rel>}(k) = \left[\sum_{x\in E_G} length(g_{<rel>}(k,x))\right]^{-1}$ |
| Partial betweenness: the probability for *k* to be on a shortest path of properties of type *rel* (or subtype) between *x* and *y*. | $B_{<rel>}(b,x,y) = \frac{nb^{g}_{<rel>}(b,x,y)}{nb^{g}_{<rel>}(x,y)}$ |
| Betweenness centrality: the sum of the partial betweenness of a node between each other pair of nodes for a a property of type *rel* (or subtype). | $C^{b}_{<rel>}(b) = \sum_{x,y\in E_G} B_{<rel>}(b,x,y)$ |

**Figure 42.** Definition of SNA notions in labelled oriented graphs and notations used

| SNA indices | SPARQL formal definition |
|---|---|
| $nb^{actor}_{<type>}(G)$ | `select merge count(?x) as ?nbactor from <G>`<br>`where{`<br>`  ?x rdf:type param[type]`<br>`}` |
| $nb^{actor}_{<rel>}(G)$ | `select merge count(?x) as ?nbactors from <G>`<br>`where{`<br>`  {?x param[rel] ?y}`<br>`  UNION{?y param[rel] ?x}`<br>`}` |
| $nb^{subject}_{<rel>}(G)$ | `select merge count(?x) as ?nbsubj from <G>`<br>`where{`<br>`  ?x param[rel] ?y`<br>`}` |
| $nb^{object}_{<rel>}(G)$ | `select merge count(?y) as ?nbobj from <G>`<br>`where{`<br>`  ?x param[rel] ?y`<br>`}` |
| $nb^{relation}_{<rel>}(G)$ | `select cardinality(?p) as ?card from <G>`<br>`where {`<br>`{ ?p rdf:type rdf:Property`<br>`   filter(?p ^ param[rel]) }`<br>`UNION`<br>`{ ?p rdfs:subPropertyOf ?parent`<br>` filter(?parent ^ param[rel]) }`<br>`}` |
| $Comp_{<rel>}(G)$ | `select ?x ?y from <G> where {`<br>` ?x param[rel] ?y`<br>`}group by any` |
| $D_{<rel,dist>}(y)$ | `select ?y count(?x) as ?degree where {`<br>`{?x (param[rel])*::$path ?y`<br>`filter(pathLength($path) <=  param[dist])}`<br>`UNION`<br>`{?y param[rel]::$path ?x`<br>`filter(pathLength($path) <=  param[dist])}`<br>`}group by ?y` |
| $D^{in}_{<rel,dist>}(y)$ | `select ?y count(?x) as ?indegree where{`<br>`?x (param[rel])*::$path ?y`<br>`filter(pathLength($path) <=  param[dist])`<br>`}group by ?y` |
| $D^{out}_{<rel,dist>}(y)$ | `select ?x count(?y) as ?outdegree where {`<br>`?x (param[rel])*::$path ?y`<br>`filter(pathLength($path) <=  param[dist])`<br>`}group by ?x` |
| $g_{<rel>}(from,to)$ | `select ?from ?to $path pathLength($path) as`<br>`?length where{`<br>`?from sa (param[rel])*::$path ?to`<br>`}group by ?from ?to` |
| $Diam_{rel>}(G)$ | `select pathLength($path) as ?length from <G>`<br>`where {`<br>`?y s (param[rel])*::$path ?to`<br>`}order by desc(?length)`<br>`limit 1` |
| $nb^{g}_{<rel>}(from,to)$ | `select ?from ?to count($path) as ?count`<br>`where{`<br>`?from sa (param[rel])*::$path ?to`<br>`}group by ?from ?to` |

| | |
|---|---|
| $nb^{g}_{<rel>}(b, from, to)$ | select ?from ?to ?b count($path) as ?count<br>where{<br>?from sa (param[rel])*::$path ?to<br>graph $path{?b param[rel] ?j}<br>filter(?from != ?b)<br>optional { ?from param[rel]::$p ?to }<br>filter(!bound($p))<br>}group by ?from ?to ?b |
| $C^{c}_{<rel>}(y)$ | select distinct ?y ?to pathLength($path) as<br>?length (1/sum(?length)) as ?centrality<br>where{<br>?y s (param[rel])*::$path ?to<br>}group by ?y |
| $B_{<rel>}(b, from, to)$ | select ?from ?to ?b<br>(count($path)/count($path2)) as ?betweenness<br>where{<br>?from sa (param[rel])*::$path ?to<br>graph $path{?b param[rel] ?j}<br>filter(?from != ?b)<br>optional { ?from param[rel]::$p ?to }<br>filter(!bound($p))<br>?from sa (param[rel])*::$path2 ?to<br>}group by ?from ?to ?b |
| $C^{b}_{<rel>}(b)$ | Non SPARQL post-processing on shortest paths. |

**Figure 43.** Formal definition in SPARQL of semantically parameterized SNA indices.

### 5.1.3 SemSNA: the Ontology of Social Network Analysis

SemSNA is an ontology of Social Network Analysis to annotate social networks with their characteristics. This allows us to reinject the results of an analysis in the RDF representation of the social network. The presented version models strategic positions, based on Freeman's definition of centrality [Freeman 1979], and different definitions of groups with useful indices to characterize their properties. The primitives of SemSNA can be decomposed in 3 groups, (1) core primitives that describe the context of the analysis, (2) primitives that describe strategic position and strategic resources, (3) primitives that describe the network structure. The Figure 44 proposes a global view of this ontology.

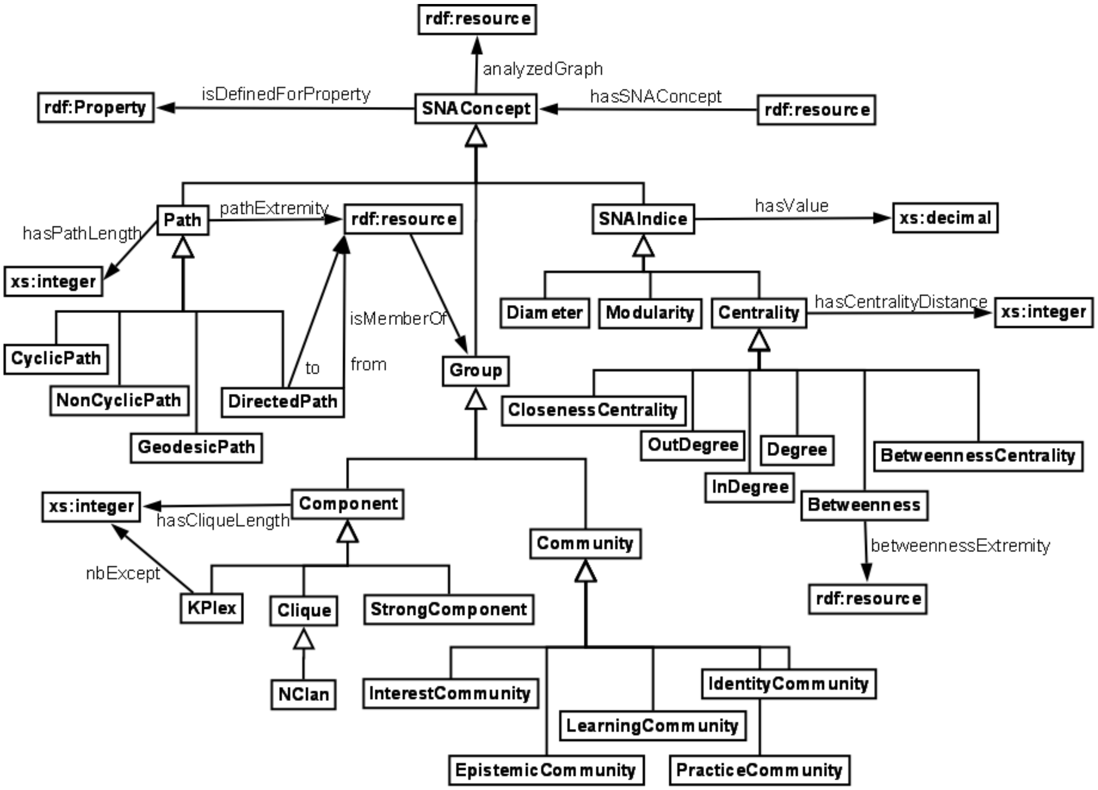


**Figure 44.** Schema of SemSNA: the ontology of Social Netwtok Analysis

### 5.1.3.1 SemSNA core

A core of primitives enables us to define an SNA concept and its context, namely the graph that has been analyzed and the properties for which it has been computed. The Figure 45 proposes a schema of this core of primitives. The main class semsna:`SNAConcept` is used as the super class for all SNA concepts. All the classes representing a concept of social network analysis in SemSNA extend this class. The property `semsna:isDefinedForProperty` indicates for which relationship, i.e., sub-network, an instance of the SNA concept is defined. In addition, the same relationships should be analyzed on different graphs, and the property `semsna:analyzedGraph` specifies the named graph [Carroll et al 2005] in which the concept has been computed. This property enables analysis to be shared, and exchanged across applications. The resource that is described by a metrics is attached to an SNA concept with the property `semsna:hasSNAConcept`. Finally, the class `semsna:SNAIndice` is a subclass of `semsna:SNAConcept`, and describes valued concepts such as centrality; the associated value is set with the property `semsna:hasValue`.

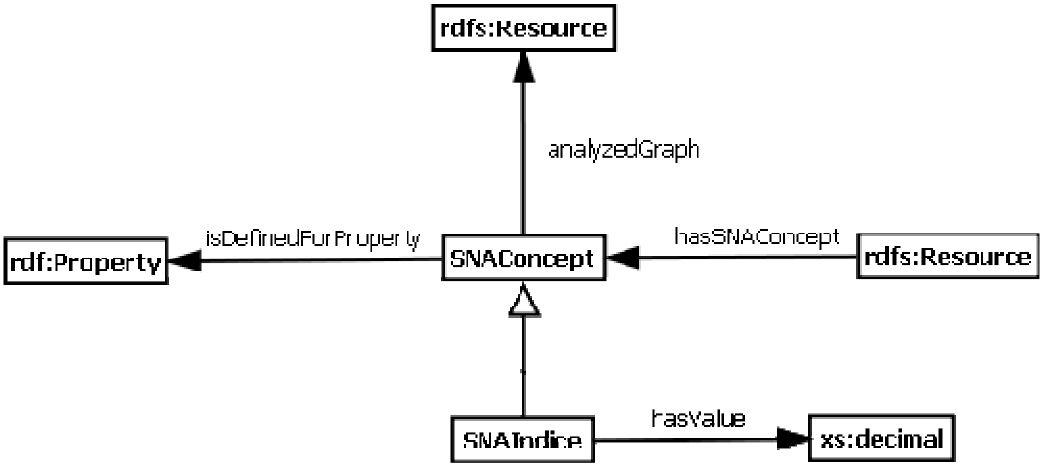


**Figure 45.** Schema of SemSNA: core concepts

### 5.1.3.2 Strategic Positions

Strategic positions are related to strategic paths in the network and different definitions of the concept of centrality. Consequently, SemSNA defines a super class for describing path, `semsna:Path`, and a super class for defining Centrality, `semsna:Centrality`.

The class `semsna:Path`, which is a sub class of `semsna:SNAConcept`, with the properties `semsna:hasPathLength` and `semsna:pathExtremity`, which respectively describe the length an the extremity of a path, are the basic primitives to represent a path. The class `semsna:DirectedPath` represents a directed path, which source and destination are defined by two sub-properties of `semsna:pathExtremity`, `semsna:from` and `semsna:to`. Then, different types of paths are represented by the sub-classes of `semsna:Path`: `semsna:CyclicPath`, `semsna:NonCyclicPath`, and `semsna:GeodesicPath`

The centrality is a valued concept, so the class `semsna:Centrality` is a sub-class of `semsna:SNAIndice`. The property `semsna:hasCentralityDistance` defines the distance of the neighbourhood taken into account to measure the centrality. SemSNA defines sub classes of `semsna:Centrality` to represent different centrality definitions. The classes `semsna:Degree`, `semsna:InDegree`, `semsna:OutDegree`, represent the three definitions of degree centrality. The classes `semsna:BetweennessCentrality` and `semsna:Betweenness`, respectively represent the betweenness centrality and the partial betweenness. The property `semsna:betweennessExtremity` specify for which nodes the partial betweenness has been computed. The closeness centrality is represented by the class `semsna:ClosenessCentrality`.

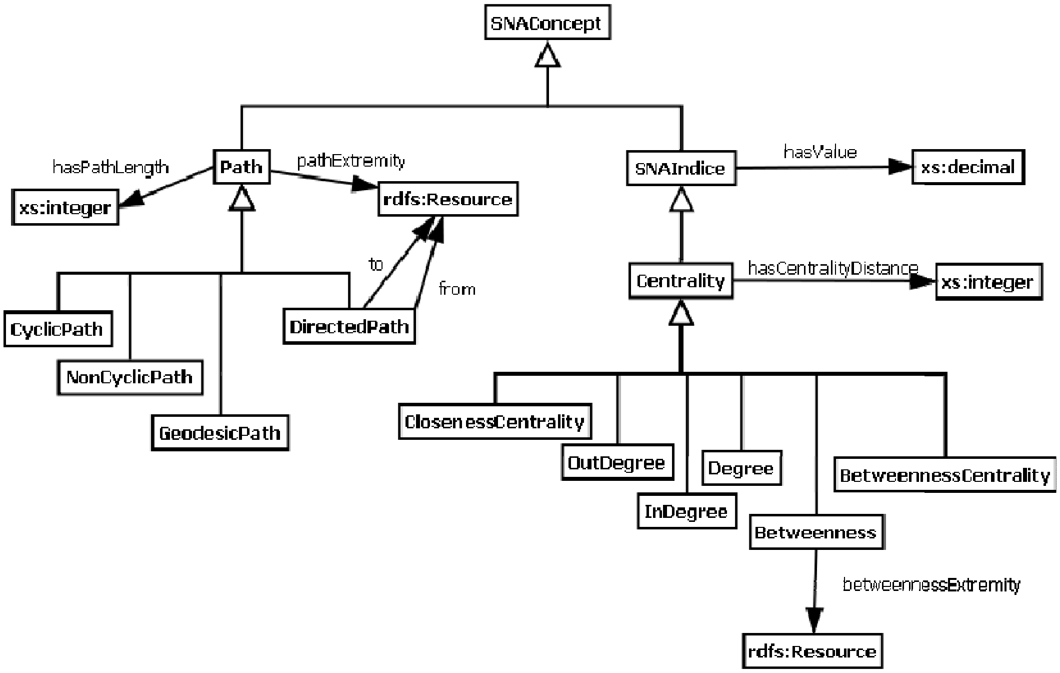


**Figure 46.** Schema of SemSNA: paths and strategic positions.

### 5.1.3.3 Network Structure

We propose a set of primitives to define and annotate groups of resources linked by particular properties. The class `semsna:Group` is a super class for all classes representing alternative concepts of group of resources. The class `semsna:Component` represents a set of connected resources. The class `semsna:StrongComponent` defines a component of a directed graph where the paths connecting its resources do not contain any change of direction. The class `semsna:Diameter`, subclass of `semsna:Indice`, defines the length of the longest geodesics (shortest paths between resources) of a component. The property `semsna:maximumDistance` enables us to restrict component membership to a maximum path length between members. A clique is a complete sub graph, for a given property according to our model. An n-clique extends this definition with a maximum path length (n) between members of the clique; the class `semsna:Clique` represents this definition, and the maximum path length is set by the property `semsna:maximumDistance`. Resources in a clique can be linked by shortest paths going through non clique members. `semsna:NClan` is a restriction of a clique that excludes this particular case. `semsna:KPlex` relaxes the clique definition to allow connecting to *k* members with a path longer than the clique distance; *k* is determined by the property `semsna:nbExcept`. Finally the concept `semsna:Community` supports different community definitions: `InterestCommunity`, `LearningCommunity`, `GoalOrientedCommunity`, `PraticeCommunity` and `EpistemicCommunity` [Conein 2004] [Henri & Pudelko 2003]. These community classes are linked to more detailed ontologies like [Vidou et al 2008] used to represent communities of practice.

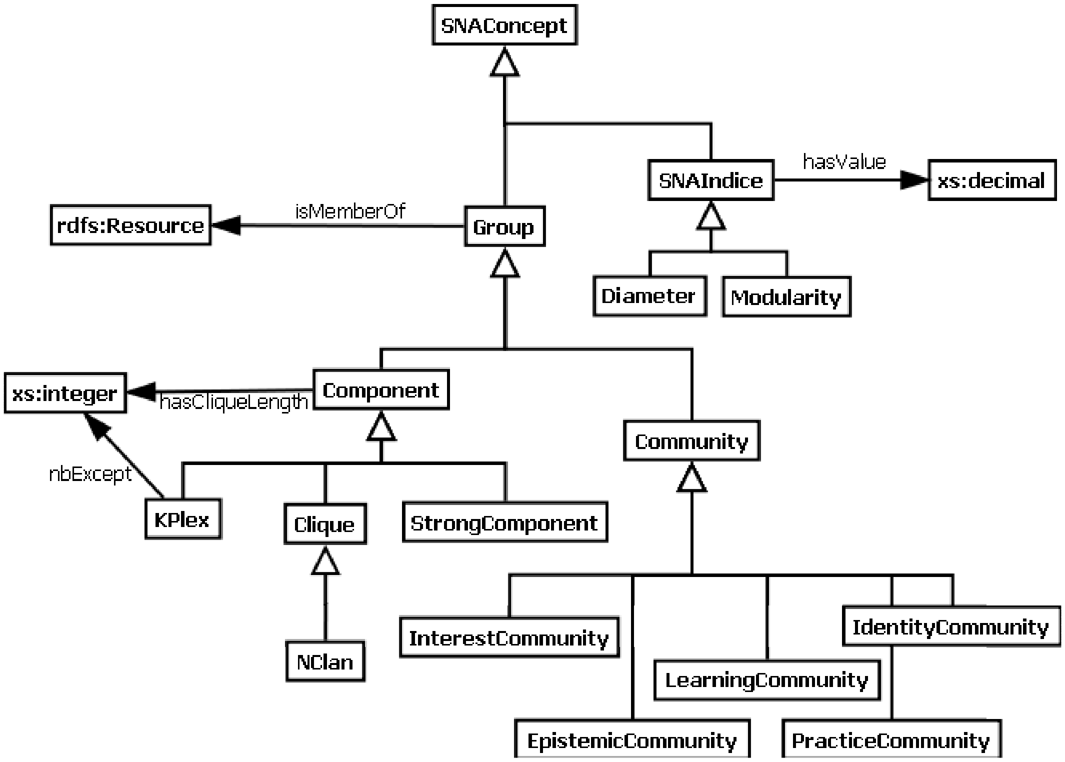


**Figure 47.** Schema of SemSNA: groups and communities

Annotating social data with SemSNA enables us to query directly the social network in a cheaper way and to focus on important values of indices. Moreover, time-consuming queries can't be done in real time. We compute them as batch reporting and generate relevant SNA annotations enriching the graph in order to respond quickly to queries on demand. The Figure 48 proposes an example of a semantic social graph that is enriched with SemSNA primitives. In this social network, Guillaume has both *family* and *professional* relationships. The SemSNA annotations state that: the degree of Guillaume for a neighbourhood at distance 2 and the property `colleague` (a super property of `supervisor`) is 4.

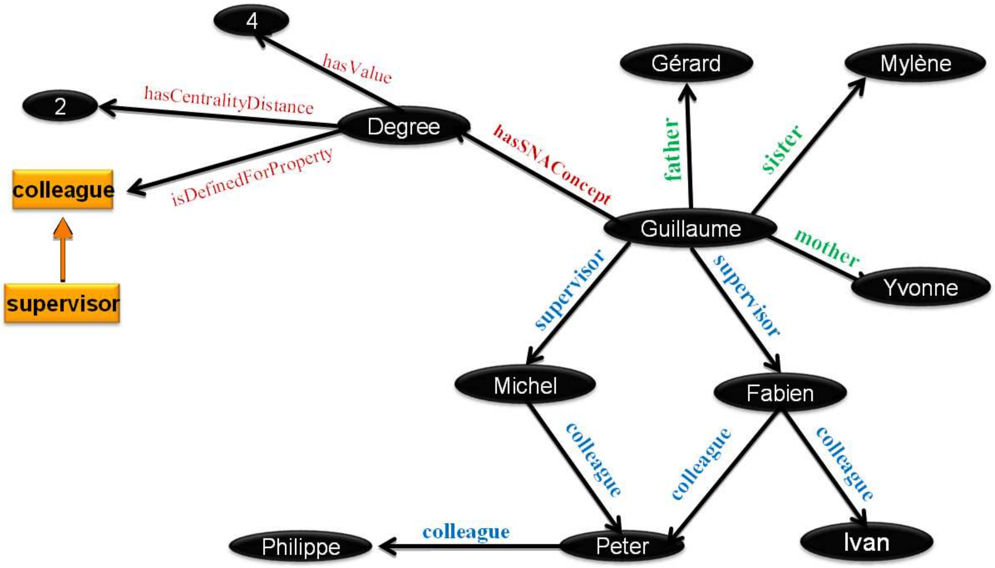


**Figure 48.** Example of a social graph enriched with social network analysis results.

## 5.2 Experiment and Results

We have experimented and evaluated this framework on a real online social network: Ipernity.com. This social network, offers users several options for building their social network and sharing multimedia content. Every user can share pictures, videos, music files, create a blog, a personal profile page, and comment on other's shared resources. Every resource can be tagged and shared. To build the social network, users can specify the type of relationship they have with others: *friend*, *family*, or *favourite* (simple contact you follow). Relationships are not symmetric, Fabien can declare a relationship with Michel but Michel can declare a different type of relationship with Fabien or not have him in his contact list at all; thus we have a directed labelled graph. Users have a homepage containing their profile information and pointers to the resources they share. Users can post on their profile and their contacts' profiles depending on access rights. All these resources can be tagged including the homepage. A publisher can configure the access to a resource to make it public, private or accessible only for a subset of its contacts, depending on the type of relationship (*family*, *friend* or *favourite*), and can monitor who visited it. Groups can also be created with topics of discussion with three kinds of visibility, public (all users can see it and join), protected (visible to all users, invitation required to join) or private (invitation required to join and consult).

### 5.2.1 Representing and Leveraging Ipernity Relational Data with Semantics

The social data of Ipernity.com were provided as a dump of their database that we mined to build a semantic social network. Existing ontologies like FOAF and SIOC presented in chapter 4 were sufficient to model the core elements of the social graph of Ipernity. However, we extended the primitives of these ontologies for finely handling the semantics of the rich interactions embedded in this social network. We designed these primitives in a domain ontology: SemSNI, that stands for Semantic Social Network Interactions.

We used FOAF in combination of RELATIONSHIP to represent persons in Ipernity, and their relationships. First, we declared an instance of `foaf:Person` for any user of Ipernity, to represent the person that hold the Ipernity account. Then, we linked the created persons with their relationships. We used the properties `rel:friendOf` and `rel:knowsByReputation` to represent respectively the *friend* and *favourite* relationships. RELATIONSHIP proposes many properties to represent *family* links however it does not provide a super property for them, which would have been suited for the *family* links of Ipernity. Consequently, we defined a sub property of `foaf:knows` to type the family relationships: `semsni:family`. Finally we created an instance of `sioc:UserAccount` for any Ipernity user that we linked to the corresponding `foaf:Person` instance with the property `foaf:account`. The Figure 49 presents how these bases represent persons, their account, and the structure of their social network.

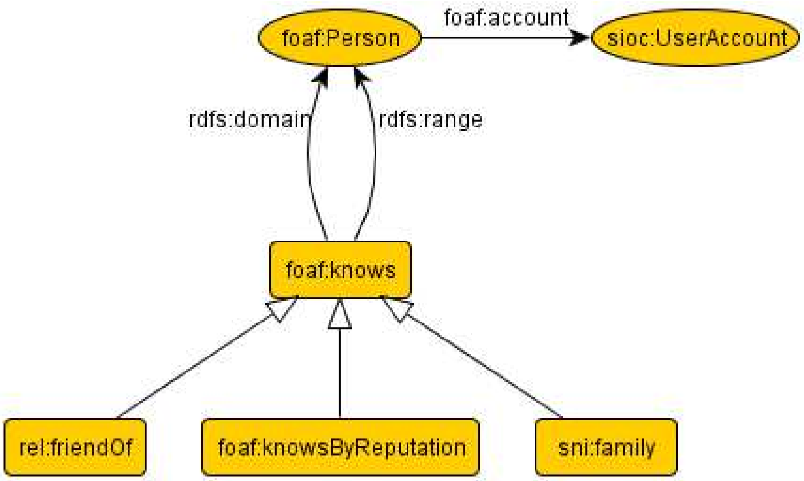


**Figure 49.** Schema of basic primitives representing persons, users and content in Ipernity.

Then we represented the content produced by Ipernity users, which mediates their interactions. Primitives that extend the class `sioc:Item` represent shared content. The Figure 50 presents an overview of the ontology we designed to model this social network, and how this ontology is linked to existing models. In order to model homepages, private messages, discussion topics, and documents that do not exist in SIOC types with the required semantics, we designed different primitives. SemSNI defines the class `semSNI:UserHome, semSNI:PrivateMessage, semSNI:Topic` and `semSNI:Document` as subclasses of `sioc:Item` (`semSNI:Document` also extends `foaf:Document`). The class `semSNI:Visit` and the properties `semSNI:visitedResource` and `semSNI:hasVisitor` enable us to describe the visits of a user to a resource. In order to infer new relationships between users from their interactions and the content they share, SemSNI defines the class `semsni:Interaction` and the property `semSNI:hasInteraction` (domain:

sioc:User, range: sioc:User). The classes representing exchanges on a content (sioct:Comment, semSNI:Visit and semSNI:PrivateMessage) are defined as subclasses of semSNI:Interaction and we can infer a relation semSNIhasInteraction between the creator of the resource and its consumer. We did not type more precisely such relations, but we can extend this property in order to increase the granularity in the description of interactions between users. We use the types of access defined in the Access Management Ontology (amo:Public, amo:Private, amo:Protected) [Buffa & Faron-Zucker 2010] in combination with the property semSNI:sharedThroughProperty to model the kind of sharing for resources.

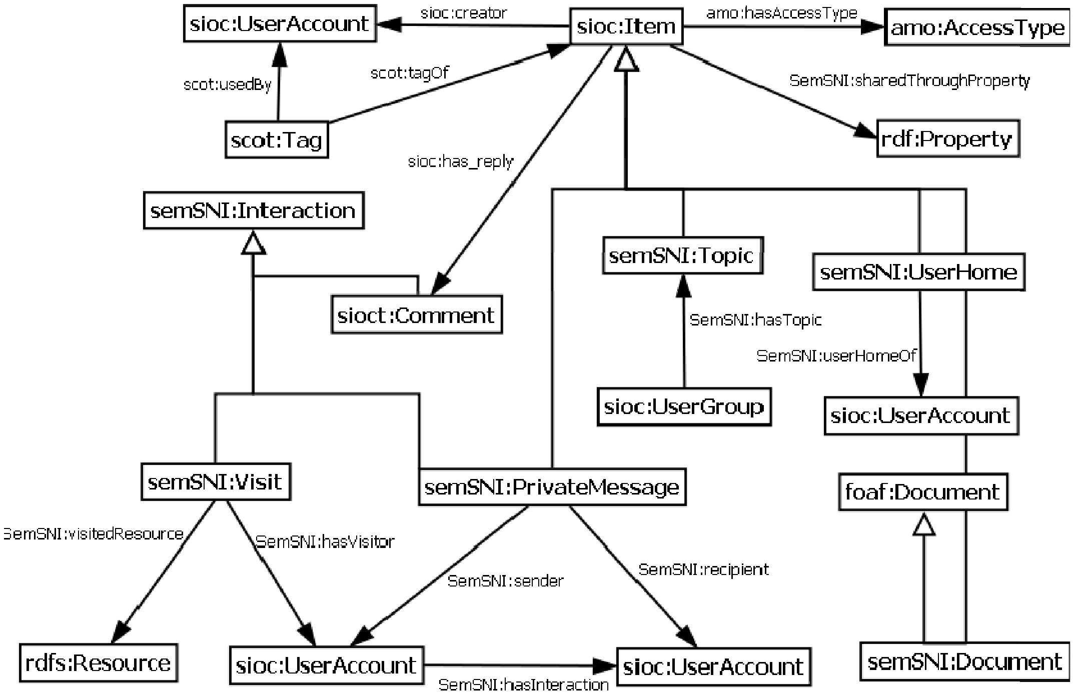


**Figure 50.** Schema of SemSNI; an ontology of Social Network Interactions for modelling and enriching the social data of Ipernity.com

Based on these presented primitives, we used the Corese graph engine to extract a semantic social graph from the database of Ipernity, with the method presented in section 5.1.1. Once in RDF, we queried this social network and exploited its semantics with Corese. In the next section we present and interpret the results of analyzing this social network with different semantic perspectives using the queries presented in section 5.1.2.

### 5.2.2 Results

We tested our algorithms and queries on a bi-processor quadri-core Intel(R) Xeon(R) CPU X5482 3.19GHZ, 32Gb of RAM. We applied the defined queries on relations and interactions from Ipernity.com. We analyzed the three types of relations separately (*favourite*, *friend* and *family*) and also used polymorphic queries to analyze them as a whole using their super property: foaf:knows. We also analyzed the interactions produced by exchanges of private messages between users, as well as the ones produced by someone commenting someone else's shared item.

We first applied quantitative metrics to get relevant information on the structure of the network and activities: the number of links and actors, the components and the diameters. 61,937 actors are involved in a total of 494,510 relationships. These relationships are decomposed in 18,771 *family* relationships between 8,047 actors, 136,311 *friend* relationships involving 17,441 actors and 339,428 *favourite* relationships for 61,425 actors. These first metrics show that the semantics of relations are globally respected, as *family* relations are less used than *friend* and *favourite*. 7,627 actors have interacted through 2,874,170 comments and 22,500 have communicated through 795,949 messages. All these networks are composed of a largest component containing most of the actors (Figure 51) and few very small components (less than 100 actors) that show "the effectiveness of the social network at doing its job" [50], in connecting people. The interaction sub networks have a very small diameter (3 for comments and 2 for messages) due to their high density. The *family* network has a high diameter (19), consistent with its low density. However the *friend* and *favourite* networks have a low density and a low diameter revealing the presence of highly intermediary actors.

The betweenness and degree centralities confirm this last hypothesis. The *favourite* network is highly centralized, with five actors having a betweenness centrality higher than 0, with a dramatically higher value for one actor who has a betweenness centrality of 1,999,858 and the 4 other ones who have a value comprised between 2.5 and 35. This highest value is attributed to the official animator of the social network who has a *favourite* relationship with most actors of the network, giving him the highest degree: 59,301. In the *friend* network 1,126 actors have a betweenness centrality going from 0 to 96,104 forming a long tail, with only 12 with a value higher than 10,000. These actors do not include the animator, showing that the *friend* network has been well adopted by users. The *family* network has 862 actors with a betweenness centrality from 0 to 162,881 with 5 values higher than 10,000. Only one actor is highly intermediary in both *friend* and *family* networks. The centralization of these three networks presents significant differences showing that the semantics of relations have an impact on the structure of the social network. The betweenness centralities of all the relations, computed using the polymorphism in SPARQL queries with the `foaf:knows` property, highlight both the importance of the animator who has again the significantly highest centrality and the adoption by users with 186 actors playing a role of intermediary. The employees of Ipernity.com have validated these interpretations of the metrics that we computed, showing the effectiveness of a social network analysis that exploits the semantic structure of relationships.

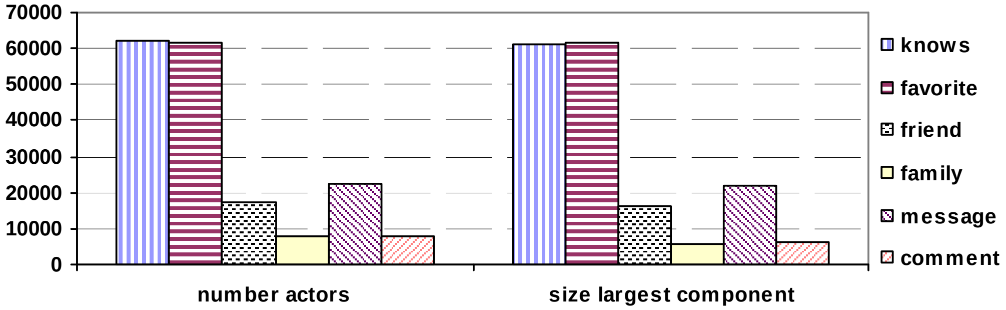


**Figure 51.** Number of actors and size of the largest component of the studied networks

The Corese engine works in main memory and such an amount of data is memory consuming. The 494,510 relations declared between 61,937 actors use a space of 4.9 Gb. Annotations of all messages use 14.7 Gb and the representation of documents with their comments use 27.2 Gb. On the other hand working in main memory allows us to process the network very rapidly. The path computation is also time and space consuming and some queries had to be limited to a maximum number of graph projections when too many paths could be retrieved. However, in that case approximations are sufficient to obtain relevant metrics on a social network, i.e., for centralities [Brandes and Pitch 2007] [Bader et al 2007] [Geisberg et al 2008]. Moreover, we can limit the distance of the paths we are looking for by using others metrics. For example, we limit the depth of paths to be smaller or equal to the diameter of the components when computing shortest paths. The table of the Figure 52 summarizes the performances of some queries on this social network using different parameters.

| Indice | Relation | Query time | Nb of graph projections |
|---|---|---|---|
| $Diam_{rel>}(G)$ | Knows | 1 m 41.93 s | 10,000,000 |
| | Favourite | 1 m 51.37 s | 10,000,000 |
| | Friend | 1 m 42.62 s | 10,000,000 |
| | Family | 1 m 54.28 s | 10,000,000 |
| | Comment | 35.06 s | 1,000,000 |
| | Message | 1 m 50.84 s | 10,000,000 |
| $nb_{actors<rel>}(G)$ | Knows | 1 m 9.62 s | 989,020 |
| | Favourite | 2 m 35.29 s | 678,856 |
| | Friend | 11.67 s | 272,622 |
| | Family | 0.68 s | 37,542 |
| | Message | 17.62 s | 1,448,225 |
| | Comment | 8 m 27.25 s | 7,922,136 |
| $Comp_{<rel>}(G)$ | Knows | 0.71 s | 494,510 |
| | Favourite | 0.64 s | 339,428 |
| | Friend | 0.31 s | 136,311 |
| | Family | 0.03 s | 18,771 |
| | Message | 1.98 s | 795,949 |
| | Comment | 9.67 s | 2,874,170 |
| $D_{<rel,1>}(y)$ | Knows | 20.59 s | 989,020 |
| | Favourite | 18.73 s | 678,856 |
| | Friend | 1.31 s | 272,622 |
| | Family | 0.42 s | 37,542 |
| | Message | 16.03 s | 1,591,898 |
| | Comment | 28.98 s | 5,748,340 |
| Shortest paths used to calculate $C_{b<rel>}(b)$ | Knows | Path length <= 2: 2h 56m 34.13s | 1,000,000 |
| | Favourite | Path length <= 2: 5h 33m 18.43s | 2,000,000 |
| | Friend | Path length <= 2: 1m 12.18 s | 1,000,000 |
| | Family | Path length <= 2 : 27.23 s<br>Path length <= 3 : 1m 10.71 s | 1,000,000<br>1,000,000 |

**Figure 52.** Performance of queries

## 5.3 Partial Conclusion

Our work aims at extending social network analysis to ontology-based representations of users, communities, links and relationships. We propose to bring ontology modelling to social network representation and analysis by extending social network analysis to ontology-based social graph inferences to detect and stimulate communities of interest.

We exploit ontology-based representation of social networks to support new algorithms for discovering and monitoring a community's activity. The ultimate goal is to support functionalities that foster exchanges using ontology-based representations, and to exploit feedback from usage to drive the evolution of these representations.

Our framework allows analyzing these rich typed representations of social networks and handling the diversity of interactions and relationships with parameterized SNA metrics. Classical SNA ignores the semantics of richly typed graphs like RDF and classical Relational Database approaches do not offer simple mechanisms for handling the semantics of type lattices. Subsumption relations are natively taken into account when querying the RDF graph in SPARQL with an engine like CORESE. Parameterized operators formally defined in SPARQL rely on this to allow us to adjust the granularity of the analysis of relations. Moreover, a new range of pre-processing can be used such as rules crawling the network to add types or relations whenever they detect a pattern (e.g., an actor frequently commenting on posts by another actor is linked to him by a relation "monitors").

New queries that compute new operators can be defined at anytime and SemSNA can be extended. Network assortativity [Newman 2003a] is an example of future operators that could both leverage the semantics of the schemas (e.g., similarity between two nodes) and extension mechanisms of SPARQL (e.g., counting the number of shared connections). In addition, using a schema to add the results of our queries (or rules) to the network also allows us to decompose complex processing into two or more stages and to factorize some computation among different operators, e.g., we can augment the network with in-degree calculation and betweenness calculation and then run a query on both criteria to identify nodes with an in-degree > y and a betweenness > x.

Furthermore we validated this framework on a real social network and revealed the importance of considering the diversity of relationships and their semantic links. The sub-networks we analyzed present different characteristics that highlight in particular the strategic actors and the partitioning of the different activities. The approach is applied as batch processing on large RDF triple store (CORESE is a freeware handling millions of nodes but other engines with the same extensions could be used just as well). Consequently we annotate the social data with the results of these parameterized SNA metrics using SemSNA ontology to provide services based on this analysis (e.g. filter social activity notifications), to use them in the calculation of more complex indices or (in the future) to support iterative or parallel approaches in the computations. Computation is time consuming and even if CORESE runs in main memory, experiments reported in the paper show that handling a network with millions of actors is out of our reach today. We started to study different approaches for addressing that problem: (1) identifying computation techniques that are iterative, parallelizable, etc. (2) identifying approximations that can be used and under which conditions they provide good quality results (3) identifying graph characteristics (small worlds, diameters, etc.) that can help us cut the calculation space and time for the different operators.

# 6. SemLP: When Semantics Improve Community Detection in Folksonomies

Building on top of our results on semantic social network analysis, we present a community detection algorithm, SemTagP, that takes benefits of the semantic data that were captured while structuring the RDF graphs of social networks. SemTagP not only offers to detect but also to label communities by exploiting (in addition to the structure of the social graph) the tags used by people during the social tagging process as well as the semantic relations inferred between tags. Doing so, we are able to refine the partitioning of the social graph with semantic processing and to label the activity of detected communities. We tested and evaluated this algorithm on the social network built from Ph.D. theses funded by ADEME, the French Environment and Energy Management Agency. We extracted from this dataset 1853 actors, 13,982 relationships, 6,583 tags and 3,570 `skos:narrower` relations between 2,785 tags and we showed the communities that can be detected.

## 6.1 Community Detection by Label Propagation and Folksonomies

Community detection helps understanding the distribution of actors and activities. Algorithms that tackle this problem are either hierarchical or based on heuristics (see overview in chapter 3). Hierarchical algorithms produce a tree of community partitions by iteratively dividing the network into sub communities (top-down) or by merging communities into larger one (bottom-up). Heuristics based algorithms, for instance the ones based on random walk or the ones based on analogies with electrical networks, exploit network's characteristics to determine densely connected group of nodes. Among heuristics based algorithms, one uses label propagation [Raghavan and al 2007]. This algorithm, also known as RAK in reference to the initials of its authors, proposes to detect communities by propagating labels in the social network as follows:

(1) The algorithm assigns a unique random label to each node.

(2) Each node *n* replaces its label by the label most used by its neighbors (adjacent nodes) in the graph, if its own label is different. In case several labels are the most used, one is chosen randomly.

(3) If at least one node changed its label, go to step 2

(4) Else nodes that share the same label form a community.

The Figure 53 presents this algorithm on a toy example.

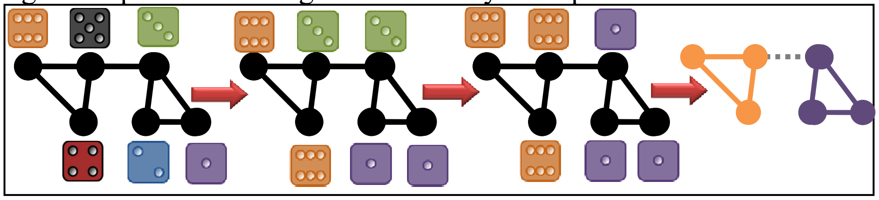


**Figure 53.** Toy example of the execution of the RAK algorithm.

Social web applications made social tagging popular: users categorize resources (e.g. media, blog posts, etc.) with freely chosen keywords called tags. This process generates a folksonomy: a set of actors describing a set of objects with a set of tags. A pioneering work by Peter Mika [Mika 2005] investigated folksonomies as lightweight ontologies emerging from the usages of communities. Each tag may represent a community of interest that is composed of all the actors using this tag. Tags enable people to easily classify online resources for their personal use or for targeted communities, and to freely join online interactions. Tags shared by several users form a new source of links between users: "interaction produces similarity, while similarity produces interaction" [Mika 2005]. For instance, during the Iran election, people overcame the media censorship with the Twitter social network by annotating their posts with the same tag, #iranelection, in order to interact and gather their information. Tags enable to link users and to label their emerging community.

Some tags are semantically related (hyponyms, synonyms, etc.) and a set of linked tags can also be viewed as a vocabulary shared by members of a community. Different approaches were proposed to structure folksonomies and identify semantic relations between tags with automatic processing or user contributions (see overview in [Limpens 2010]). Recently, [Limpens 2010] defined a method to combine automatic processing and manual user contributions to help online communities semantically enrich folksonomies and structure their own vocabularies. Once folksonomies are typed and structured, the relations between the tags and between tags and users provide a new source of affiliation networks, which enables us in this article to refine the labeling process of communities.

In this chapter we propose to merge these three approaches (RAK, tag based labeling and folksonomy structuring) in order to perform community detections that take benefits, not only of the link structure of the social network, but also of the emerging semantics of folksonomies. We first introduce SemTagP, an algorithm that turns the RAK random label propagation into a semantic tag propagation in order to detect communities and meaningfully label them. Then we present how we implemented this algorithm with Semantic Web frameworks in order to take benefits of the ontological primitives used to type RDF graphs. Finally, we present the result that we obtained with a social network built from Ph.D. theses funded by the ADEME, the French Environment and Energy Management Agency.

## 6.2 SemTagP: Semantic Tag Propagation in Networks

SemTagP is an algorithm that detects and characterizes communities from the directed typed graph formed by RDF descriptions of (social) networks and folksonomies. Using existing ontologies to represent online social networks (see chapter 4), we can link and type online social networks, associate their actors to tags and semantically relate tags to each other (see Figure 54).

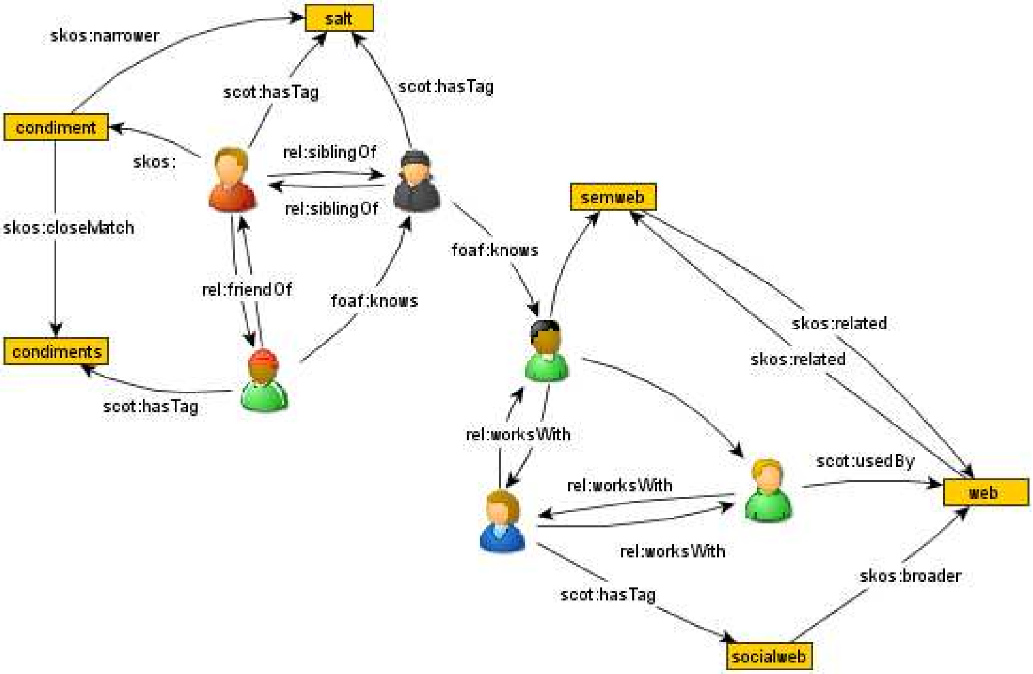


**Figure 54.** Example of an RDF description of a social network and a structured folksonomy.

SemTagP is an extension of the RAK algorithm that turns the label propagation into a semantic propagation of tags: instead of assigning and propagating random labels, we assign to actors the tags they use and we propagate them using generalization relations between tags (e.g. `skos:narrower/skos:broader`) to merge over specialized communities and generalize their labels to common hyperonyms.

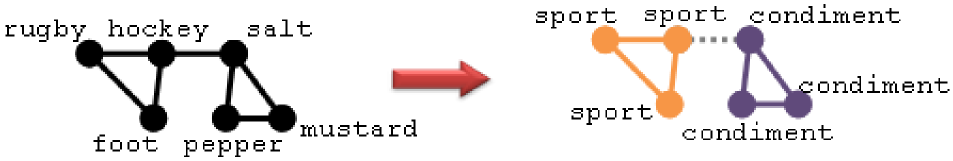


**Figure 55.** Toy example of a semantic tag propagation.

We use the directed modularity on RDF directed graphs [Leitch & Newman 2008] (see Definition 47 in chapter 3) to assess the quality of the community partition obtained after each propagation loop. When a partitioned network has a high modularity, it means that there are more connections between nodes within each community than between nodes from different communities.

SemTagP iteratively propagates the tags in the network in order to get a new partitioning: nodes that share the same tag form a community. During a propagation loop each actor chooses the most used tag among its neighbors, for a tag *t* we count 1 occurrence for each neighbor using *t* and 1 occurrence for each neighbor using a `skos:narrower` tag of *t*. We iterate until the modularity stops increasing. The penultimate partitioned network is the output of the algorithm.

In our previous results on semantic social network analysis (see chapter 5), we highlighted the importance of considering the diversity and the semantic of links between actors. Propagating tags through different types of relations, namely in different sub-networks, could produce different community partitions. Consequently, SemTagP is parameterized by the type of the analyzed relation. We formalize SemTagP as follow:

```
Algorithm SemTagP(RDFGraph network, Type relationType)
 DO
   old_network = network
   //propagate tags (i.e. compute new partitions)
   FOREACH user in network.users
     user.tag = mostUsedNeighborTag(user, relationType)
   END
 WHILE modularity(network) > modularity(old_network)
 RETURN old_network
Algorithm mostUsedNeighborTags(User user, Type
relationType)
 resultTag = null; max = 0; tagTable = new hashTable()
 FOREACH agent in user.neigbors[relationType]
   IF tagTable.exists(agent.tag)
     tagTable[agent.tag] ++
   ELSE
     tagTable[agent.tag] = 1
   IF(max < tagTable[agent.tag]){
     resultTag = agent.tag;  max = tagTable[agent.tag]
   FOREACH broaderTag in agent.tag.broaders
    IF tagTable.exists(broaderTag)
      tagTable[broaderTag] ++
    ELSE
      tagTable[broaderTag] = 1
    IF max < tagTable[broaderTag]
      resultTag = broaderTag; max = tagTable[broaderTag]
   END
 END
 RETURN resultTag
```

In our first experimentation, we witnessed that some tags with many `skos:narrower` relations absorbed too many tags during the propagation phase, such as the tag *environnement* (environment), which is ubiquitous in the corpus of the ADEME agency.

Such tags grouped actors in very large communities. Consequently, we added an option to refine manually the results: after the first propagation loop we present the current community partition and labeling to a user that can reject the use of `skos:narrower` relations of tags labeling too large communities. Then, we restart the algorithm and repeat this process until no more relation is rejected, before completing the algorithm described above. For instance, during the partitioning of a social network with tags related to *web* topics, the user can reject `skos:narrower` relations of *web* such as *web* `skos:narrower` *semantic web*, in order to reveal the *semantic web* community.

We formalized here our algorithm. We will now see how we implemented this algorithm with the semantic graph engine KGRAM [Corby & Faron-Zucker 2010] that supports SPARQL 1.1 RDF query language. We delegate all the semantic processing performed on the graph to the semantic graph engine, taking benefits of SPARQL queries to exploit semantic relations between tags. Notice that the pattern matching mechanism of KGRAM's SPARQL implementation is based on graph homomorphism that is an NP complete problem. However, many optimizations enable us to significantly cut the time calculation of the RDF graph querying.

### 6.2.1 Semantic Tags Assignment and Folksonomy Enrichment

Different ontologies have been proposed to model folksonomies and social tagging activities and are used to generate RDF annotations. In particular, the SCOT ontology provides "a consistent framework for expressing social tagging at a semantic level in machine-understandable way". Tagging ontologies identify tags with URIs and consequently turn these social labels into real objects (in the RDF sense) that can be semantically described. Thus we can leverage the meaning of these apparently flat labels by using them as the subject or the object of a triple. In particular, we can infer semantic relations between tags in order to structure the folksonomy with lightweight semantics. Recently, a complete life-cycle has been proposed to enrich folksonomies by combining automatic processing of tags and users' contributions through user-friendly interfaces [Limpens 2010]. This cycle starts with a composite metric that combines several string-based metrics to reveal 3 main types of relations between tags: related, spelling variant and hyponym. The SKOS model is used to respectively represent these relations with the properties: `skos:related,` `skos:closeMatch` and `skos:narrower` (see Figure 56). Then users can validate, reject, or propose semantic relations through a web navigation tool, and emerging conflicts are solved by a referent user that maintains a consensual point of view. This cycle is iteratively restarted to maintain a folksonomy consensually augmented with semantic assertions.

We describe in the next section the way we use the resulting structured folksonomy to propagate tags, taking benefit of RDF typed graphs and SPARQL requests to ease the implementation of the different steps required by the algorithm.

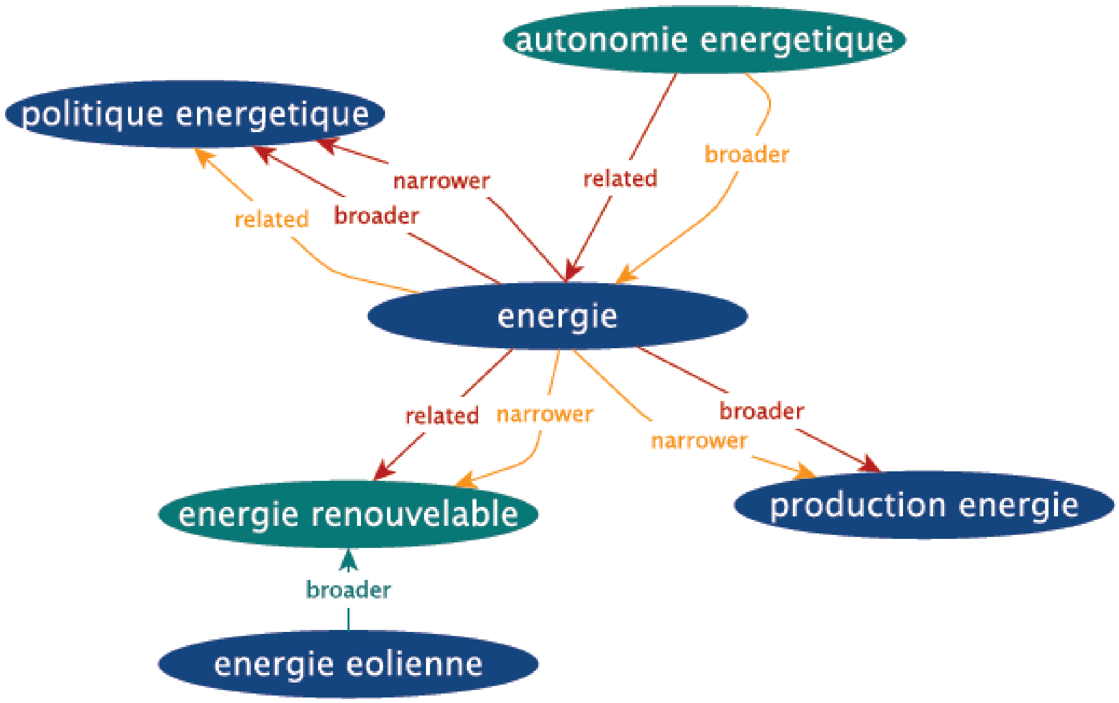


**Figure 56.** Sample of the structured folksonomy obtained in [Limpens 2010]

### 6.2.2 Semantic Tag Propagation

The propagation step consists in iteratively assigning to each actor the most frequent tag among the actors he is linked to. In order to consider generalization relations between tags, we strengthen the score of a tag with the score of its `skos:narrower` tags. For instance, we exploit the semantic statement *energy* `skos:narrower` *renewable energy* by counting one more occurrence of the tag *energy* for each occurrence of the tag *renewable energy*.

We start each loop with a query that extracts for each actor the tags of its neighbors (for a given parameterized relation), their broader tags, and we order the results by actors and tags:

```
select ?user ?tag ?y where {
  ?user param[rel] ?neighbor
  { {?neighbour scot:hasTag ?tag }
   UNION
   { ?neighbour scot:hasTag ?tag2 .
     ?tag skos:narrower ?tag2
     filter(exists{?x scot:hasTag ?tag})}}
} order by ?user  ?tag
```

Different parts of the `mostUsedNeighboursTags()` function described above are encoded in this query:

- line 2 encodes the selection of a user's neighbour
- line 3 encodes the selection of the tag of a user's neighbours
- lines 5 to 7 encode the selection of a tag that is broader than the tag of a user 's neighbor
- line 8 orders the projections for each user and tag to ease the post processing

After the completion of this request we perform a post processing on the result and replace the tag of each actor by the best ranked tag among its neighbors.

In order to handle the rejection of a generalization between two tags, we add a filter clause in the second block of the UNION clause (line 5 to 7) to exclude the use of a the specified broader tag, e.g. `filter(?tag != <http://ademe.fr/energie>)`.

Notice that the analyzed relationship is parameterized and can be replaced by any type of relation defined in the RDF graph (e.g. `sioc:follows`, `rel:worksWith`, `foaf:member`).

### 6.2.3 Computing the Modularity of an RDF Graph

The triples of an RDF description form a directed labelled graph that can be seen as the labelled arcs of an Entity-Relation graph [Baget et al 2008] (see Definition 53 in chapter 4). Thus, we define the modularity of an Entity-Relation graph as follow:

**Definition 61. modularity of an ERGraph**: the modularity of an Entity-Relation graph $G=(E_G,R_G,n_G,l_G)$ relative to a set of label *L*, for a given label of relation $p \in L$, is:

$$Q(G,p)=\frac{1}{\left|R_G^p\right|}\sum_{i,j\in E_G}[A_{ij}^p-\frac{d_{<p,i>}^{out}(G)d_{<p,j>}^{in}(G)}{\left|R_G^p\right|}]\delta(c_i,c_j)$$

Where:

- $R_G^p = \{r \in R_G ; l_G(r) = p\}$
- $A_{i,j}^p = 1$ if $\exists r \in R_G^p ; n_G^1(r) = i$ and $n_G^2(r) = j$, 0 otherwise.
- $d_{<p,i>}^{in}(G)$ and $d_{<p,i>}^{out}(G)$ are respectively the number of relations $r^{in}, r^{out} \in R_G^p ; n_G^2(r^{in}) = i$ and $n_G^1(r^{out}) = i$, namely the in and out degree of *i* for the relation labelled with *p*.

We implement this definition of the modularity by querying the RDF graph with SPARQL queries that compute different parts of this formula. In chapter 5, we defined queries to retrieve different network metrics that enable us to compute $R_G^p$, $d_{<p,i>}^{in}(G)$ and $d_{<p,i>}^{out}(G)$. First we compute $R_G^p$ with a query that simply retrieves the number of pairs of RDF resources that are linked by the property *p*. Then we retrieve the in and out degrees of all the RDF resources linked by a property *p*, with two queries that compute $d^{in}_{<p,i>}(G)$ and $d^{out}_{<p,i>}(G)$ for every possible value of *i*. For instance, the in-degrees of all resources linked by a parameter given property are computed by:

```
select ?agent count(?y) as ?indegree where {
  ?y param[property] ?agent
}group by ?agent
```

We compute the formula by iterating on the results of the two queries below.

The following query retrieves all pairs of connected resources belonging to the same community for the property given as a parameter:

```
select ?user1 ?user2 ?tag where {
  ?user1 param[property] ?user2
  ?user1 scot:hasTag ?tag
  ?user2 scot:hasTag ?tag
}group by ?user1 ?user2 ?tag
```

The following query retrieves all pairs of disconnected resources belonging to the same community for the property given as a parameter:

```
select ?user1 ?user2 ?tag where {
  ?user1 scot:hasTag ?tag
  ?user2 scot:hasTag ?tag
  filter(?user1 != ?user2)
  filter(not exists{?user1 param[property] ?user2})
} group by ?user1 ?user2 ?tag
```

We then perform a post processing on the outputs of the above queries to compute the modularity of the corresponding community partition.

## 6.3 Experiments and Results

In order to validate the benefits of our approach, we applied our algorithm on a dataset of the Ph.D. theses funded by the ADEME. Ph.Ds theses have been classified using tags and involve several actors that form a social network made of ADEME employees and academic researchers that collaborate on the funded theses. Academic agents are the Ph.D. students, the Ph.D. supervisors, and the laboratories and institutes they belong to. On the ADEME side, each thesis is followed by an engineer and attached to an internal organization called a "secteur" (sector). Free labels are used to tag the theses, for classifying purposes. From this dataset, we extracted an RDF graph (that comprises both the folksonomy and a description of the network), then we applied our algorithm in order to understand the community structure and activities of the different actors, labeled with the tags that have been used.

### 6.3.1 Dataset

The ADEME dataset we analyzed was provided as a relational database and we used the method presented in section 5.1.1 to build the corresponding RDF descriptions. Figure 57 shows a schema of the concepts we used to represent the ADEME Ph.D. network with the ontologies described in section 4.2 and an ADEME domain ontology that we designed for this analysis. Persons (engineers, students and supervisors) are declared as instances of `foaf:Person` and laboratory and sectors as instances of `foaf:Organization`. The membership of a person to an organization is described with the property `foaf:memberOf`. A student is linked to its supervisor by the property `rel:mentorOf` and to its thesis by the property `dc:creator`. We created the property `ademe:follows`, to link an ADEME engineer to a Ph.D. thesis he follows. Finally, we generated a URI for each tag used to describe a Ph.D. thesis and we used the `scot:hasTag` property to link a thesis to its tags.

Figure 58 describes how we enriched the RDF descriptions of the ADEME Ph.D. theses in order to reveal and structure the corresponding social network. We linked two persons working on the same Ph.D. with the property `rel:worksWith`. We specifically defined the property `ademe:collaboratesWith` to link two `foaf:Agent` (`foaf:Person` or `foaf:Organization`) implicated in the same thesis. Two engineers of the same sector are linked with a `rel:colleagueOf` property. We structured these social links by declaring the property `rel:worksWith` as a subproperty of `ademe:collaboratesWith`. Finally, we attached the tags of a Ph.D. to all its involved actors with the property `scot:hasTag`, producing a folksonomy with agents associating tags to thesis. We semantically enrich this folksonomy with the output of the experiment of [Limpens et al 2010] that was conducted also in our research group on the same dataset.

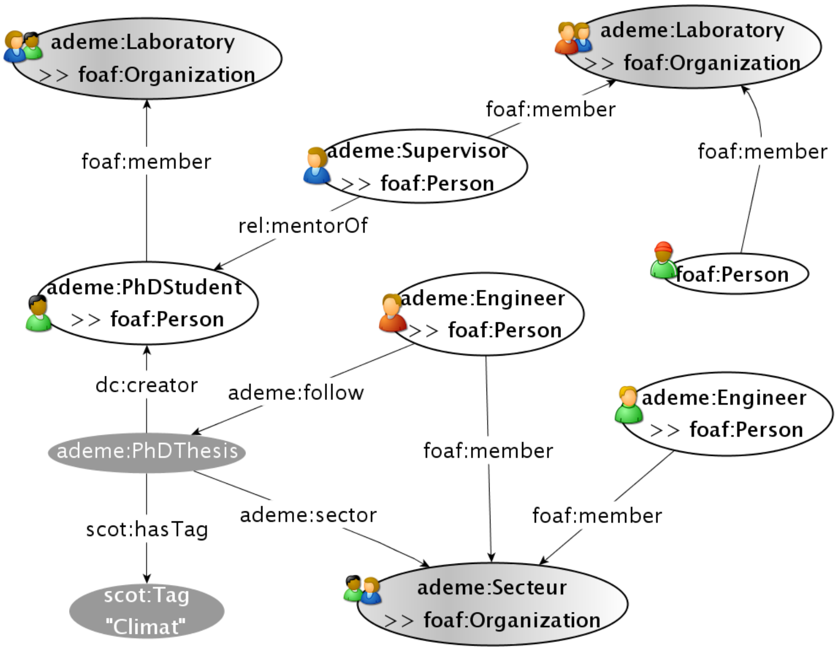


**Figure 57.** Ecosystem of a Ph.D. funded by the ADEME.

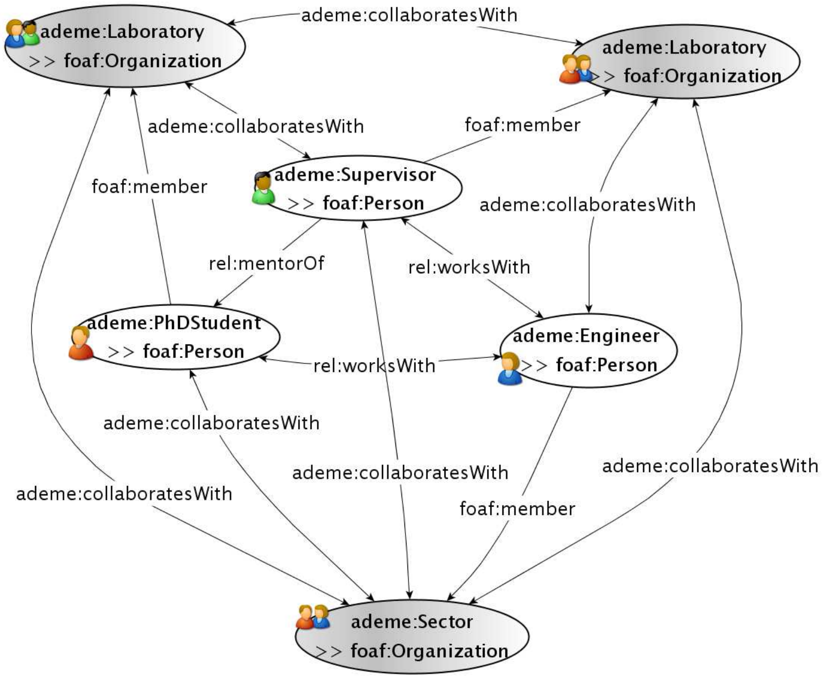


**Figure 58.** Social network of the PhDs funded by the ADEME.

### 6.3.2 Experiment

We focused our experiment on the sub network of relationships among Ph.D. academic supervisors and ADEME engineers, which are the most active actors of this network. Using the semantic social network analysis method we detailed in chapter 5, we measured the characteristics of this dataset:

- 1,853 agents with 1,597 academic supervisors and 256 ADEME engineers.

- 13,982 relationships with 10,246 `rel:worksWith` relations between ADEME engineers and academic supervisors, and 3,736 `rel:colleagueOf` relations between ADEME engineers.
- 6,583 tags, with 3,570 `skos:narrower` relations between 2,785 tags (forming a tree with a depth of 3).

This network is a connected graph that has a diameter of 8, but has a low density (0,004) and a low clustering coefficient (0,031). This network is highly centralized around the 256 engineers that have a total of 8859 relationships while the 1,597 academic actors have a total of only 5,123 relationships. Indeed, engineers follow several Ph.D. theses and have colleagues inside the ADEME while the most active academic actors supervised a maximum of 14 Ph.D. theses. Figure 59 and Figure 60 represent respectively the degree distribution of the ADEME engineers and academic supervisors, which highlight the very centralized nature of the ADEME Ph.D. social network: the 256 engineers have a total of 8859 relationships while the 1597 academic actors have a total of only 5123 relationships.

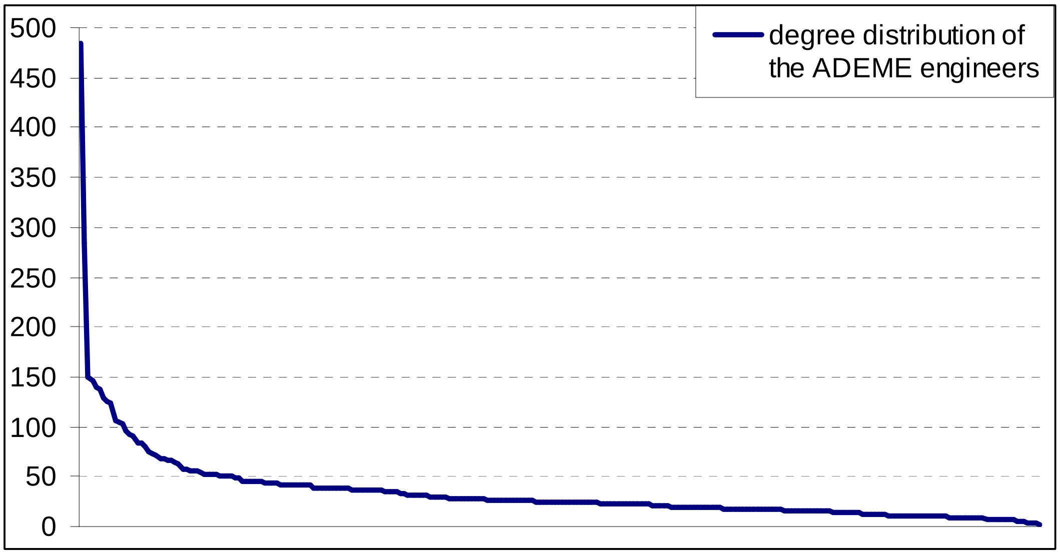


**Figure 59.** Distribution of the degrees (Y axis) of the ADEME engineers (X axis).

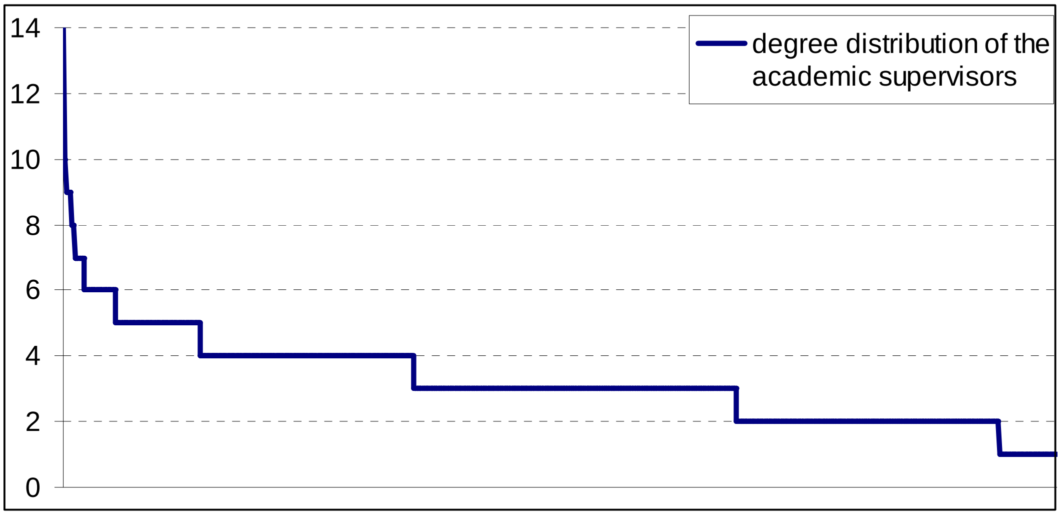


**Figure 60.** Distribution of the degrees (Y axis) of the academic supervisors (X axis).

In order to evaluate the benefits of introducing semantics in the label propagation, we compared the community that we detected with 4 different algorithms on this dataset (algorithm 2, 3, 4 are variants we developed for comparison purposes):

1. RAK: random label propagation.
2. TagP (Tag Propagation): propagation of tags without exploiting semantic relations between tags.
3. SemTagP without manual intervention.
4. Controlled SemTagP, which introduces a manual control to avoid the use of some relations between tags. We use the notation SemTagP(tag1, tag2, ...) to specify between parenthesis the tags which `skos:narrower` relations are ignored; e.g., SemTagP(env, energ, model) excludes `skos:narrower` relations with the tags *environnement*, *energetique* and *modelisation*.

We analyzed *the evolutions of the modularity* of the community partition given by the 4 algorithms and we compared these evolutions in order to observe the added-value of propagating tags (instead of random labels) and exploiting their semantics. Fig. 6 presents the curves of the evolution of the modularity of the community partition obtained after each propagation loop. We observe that SemTagP(env, energ, model) offers a community partition, which modularity outperforms the result of RAK, TagP and SemTagP. The RAK algorithm offers the weakest community partition quality on this dataset that is highly centralized with a low density of links. In other words the social links of this datasets are not sufficient enough for revealing the community structure of this social network, using RAK random label propagation. TagP and SemTagP produce community partitions with a significantly better modularity than RAK, however, when considering semantics between tags with SemTagP, we still have a modularity value close to the modularity obtained with TagP. This is due to a very broad tag: *environnement* (environment), that has many `skos:narrower` relations and that aggregates most of the actors in a single community. With SemTagP(env), we exclude the exploitation of `skos:narrower` relations with the tag *environnement*, this considerably improves the modularity value, but with lots of actors in one community tagged with energetique (energetic). Finally we obtain a pretty good modularity, 0.12, with SemTagP(env, energ, model) that excludes the use of `skos:narrower` relations of the tags: *environnement* (environment), *energetique* (energetic) and *modelisation* (modeling).

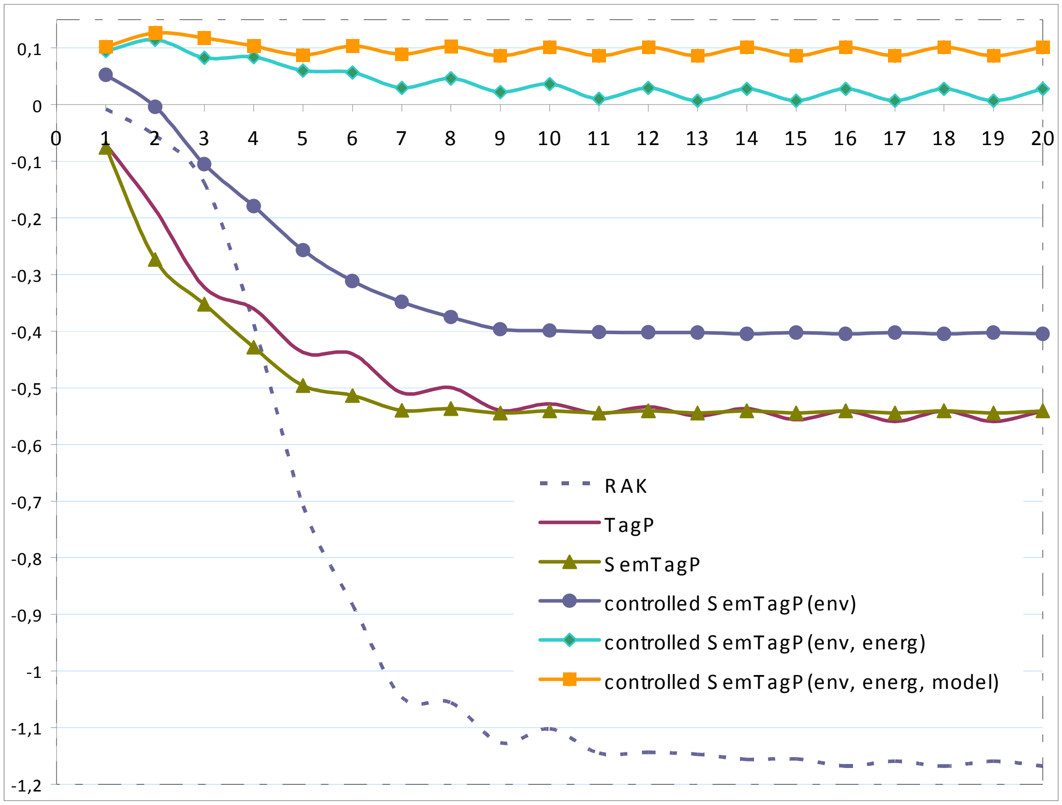


**Figure 61.** Modularity (Y axis) of the community partition obtained, after each propagation loop (X axis), with RAK, TagP, SemTagP, and 3 controlled SemTagP.

Figure 62 highlights the numerous `skos:narrower` relations of the tag *environnement* (environment) and the `skos:narrower` relations of the corresponding `skos:narrower` tags. Ignoring the use of `skos:narrower` relations of the tag *environnement* is equivalent of removing a root of the `skos:narrower` tree of the structured folksonomy. Figure 63 presents the remaining relations when we ignore these relations. This highlight the numerous tags that are no more absorbed during the semantic propagation and that can take benefits from their own `skos:narrower` relations.

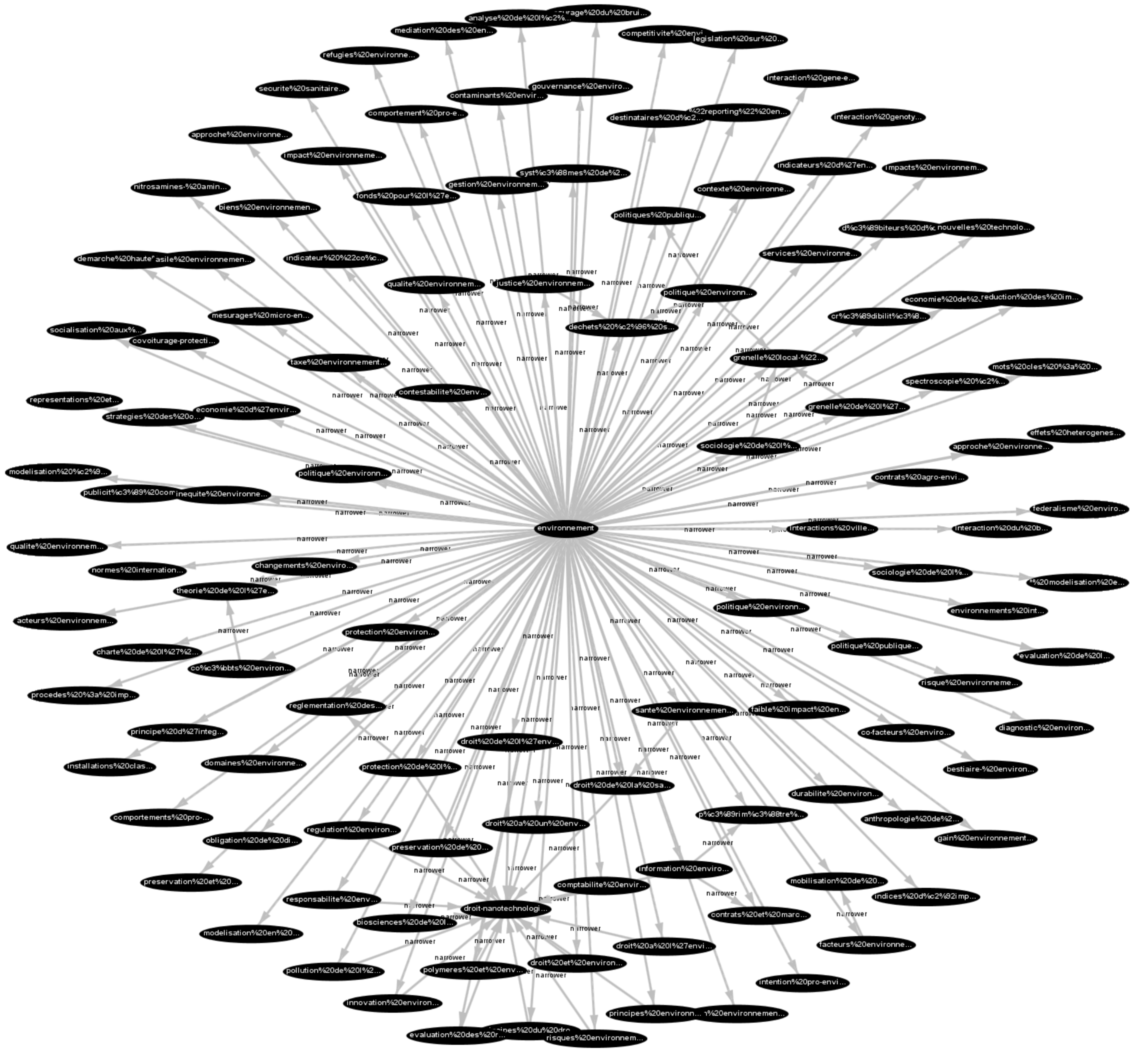


**Figure 62.** Visualization of the graph of `skos:narrower` relations of the tag *environnement* (environment) and `skos:narrower` relations between these narrower tags.

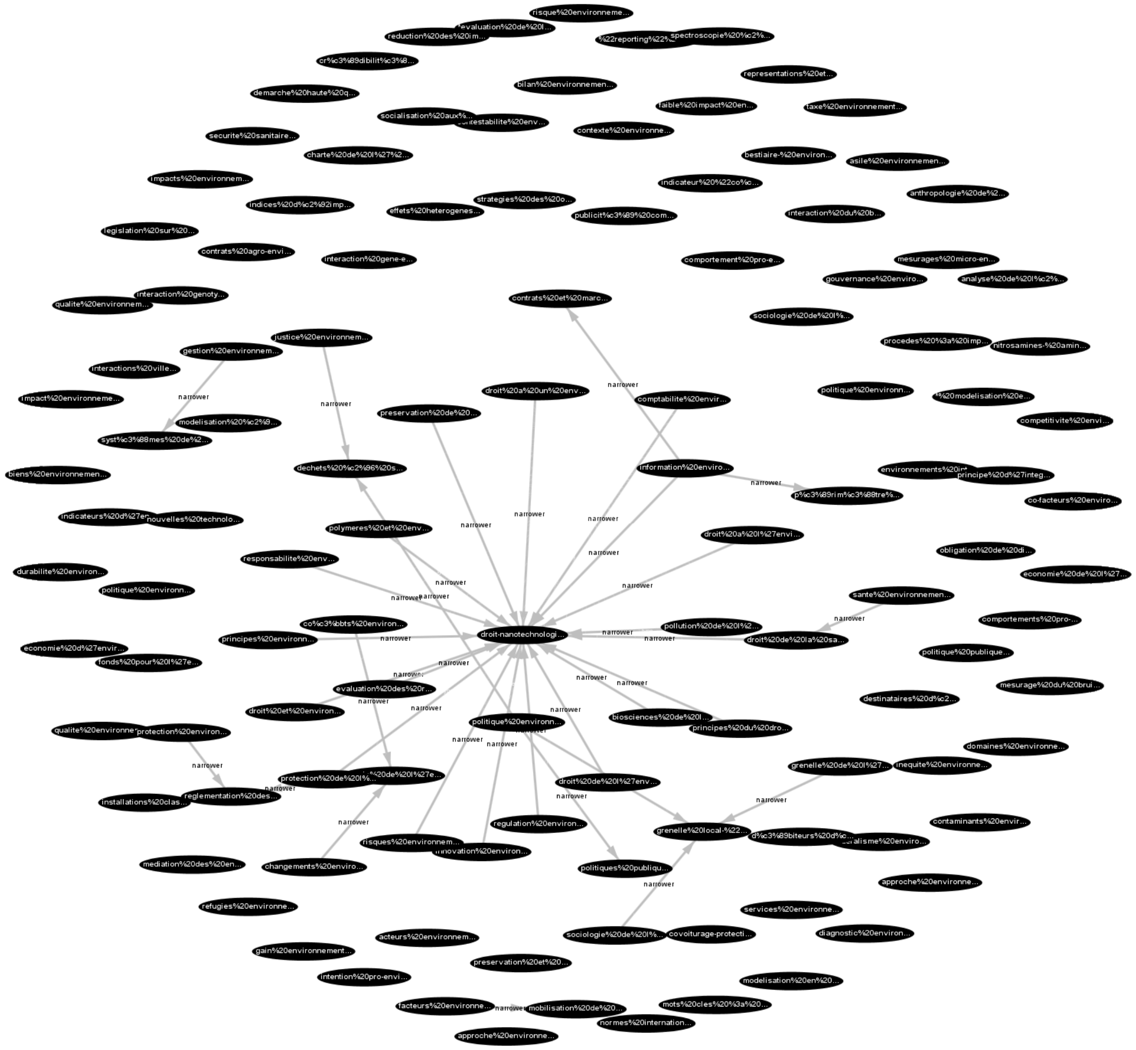


**Figure 63.** Visualization of the graph of `skos:narrower` relations between the tags that are `skos:narrower` than the tag environnement (environment).

The table of Figure 64 presents the distribution of the 30 most used tags in the initial folksonomy, with the tag propagation without semantic, and with SemTagP(env, energ, model). We observe a significant difference between these three folksonomies, which highlight that the number of users of a tag in a folksonomy is not sufficient enough to determine the existence of a community. The connectivity of the actors that use a tag and its semantic links with other tags are important elements for determining the importance of a community and its activities. TagP is effective at revealing tags that are used by well connected groups of actors, but reveals smaller communities because it cannot link tags. SemTagP amplifies this benefits and reveals larger communities labelled with tags representing semantically related tags.

| Initial folksonomy | | TagP | | SemTagP (env, energ, model) | |
|---|---|---|---|---|---|
| Tags | number of actors | Tags | number of actors | Tags | number of actors |
| Modelisation | 250 | Modelisation | 103 | Pollution | 301 |
| Developpement durable | 204 | developpement durable | 85 | Dechets | 276 |
| Environnement | 187 | Adsorption | 58 | Biomasse | 175 |
| Pollution | 105 | environnement | 39 | developpement durable | 114 |
| Optimisation | 86 | Absorption | 38 | Modelisation | 92 |
| Changement climatique | 79 | Aerosols | 33 | Metaux | 87 |
| Energie | 77 | changement climatique | 33 | Solaire | 80 |
| metaux lourds | 75 | Biomasse | 33 | Chimie | 67 |
| Pollution atmospherique | 68 | Agriculture | 28 | Analyse | 66 |
| Biomasse | 68 | Arsenic | 26 | Economie | 57 |
| Dechets | 67 | Acv | 25 | developpement | 54 |
| Energies renouvelables | 65 | Biodiversite | 20 | Agriculture | 41 |
| Sols | 64 | Aide a la decision | 19 | Transports | 40 |
| Simulation | 63 | biocarburants | 18 | Mobilite | 36 |
| Sol | 59 | air interieur | 16 | co2 | 35 |
| Efficacite energetique | 59 | absorption racinaire | 14 | Energie | 32 |
| Experimentation | 56 | amenagement du territoire | 13 | Silicium | 29 |
| Gouvernance | 55 | Bois | 12 | Membrane | 27 |
| Energie renouvelable | 54 | Atmosphere | 12 | Changement | 24 |
| Adsorption | 54 | bioremediation | 12 | Moteur | 17 |
| Evaluation | 54 | Batiment | 10 | pile a | 16 |
| Transport | 51 | énergies renouvelables | 10 | Batiment | 15 |
| Metaux | 51 | amenagement | 10 | Adsorption | 15 |
| Photovoltaique | 49 | cellule solaire silicium polycristallin couches minces | 9 | Emissions | 13 |
| Innovation | 48 | Dechets | 9 | Diphasique | 12 |
| Recyclage | 48 | bioaccumulation | 9 | caracterisations | 11 |
| France | 47 | analyse du cycle de vie | 9 | Acv | 10 |
| Cov | 46 | Analyse | 9 | Recyclage | 9 |
| Evaluation environnementale | 45 | accumulateur li-ion | 9 | Composes organiques volatils | 8 |
| Biodiversite | 44 | co2 | 9 | Architecture | 8 |
| Electrochimie | 42 | Adaptation | 9 | Cov | 7 |

**Figure 64.** comparison of the tag distribution of the 30 most used tags in the initial folksonomy, with TagP, and with SemTagP(env, energy, model).

We observe 4 different patterns of tag propagation in the ADEME network that highlight the exploitation of both the link structure and of the emerging semantics of folksonomies. On one side the tag propagation helps partitioning the network into densely linked groups of actors, and on the other side the use of semantic relations between tags helps preserving the identity of small communities, aimed to disappear during the propagation, by gathering them into broader but semantically related communities:

- Some tags used by scattered users in the social network tend to quickly disappear, even if they are used by a large number of users, and do not label a community in the final partition.
- Some tags used by well connected group of users are strengthened by the propagation and still labelling a community in the resulting partition.
- Some tags used by well connected group of users are generalized to broader tags that include and label their community in the resulting community partition.
- Some tags are strengthened by the exploitation of the semantic relations that enable the algorithm to connect semantically related tags and to gather actors working on similar topics but using narrower tags representing different sub topics.

The table of Figure 65 compares the size of the communities labelled with 7 tags, initially used by a similar number of users (ranged between 48 and 54), with TagP and SemTagP(env, energ, model). We observe the 4 different propagation patterns described above:

- the tags *évaluation* (evaluation), *photovoltaïque* (photovoltaic) and *innovation* disappeared in both cases because these tags and their `skos:narrower` tags were used by scattered users in the networks.
- the tags *adsorption* and *recyclage* have respectively only 1 and 2 `skos:narrower` relations (with tags used by less than 5 actors). These tag have not been absorbed during the propagation phase, nor with TagP, nor with SemTagP(env, energe, model.
- The tag *transport* disappeared with both propagations but has been generalized by SemTagP(env, energ, model) to a spelling variant, considered as a broader tag: *transports, which* has 38 `skos:narrower` tags.
- The tag *metaux* (metals) that nearly disappeared with TagP is reinforced with SemTagP by its semantic relations. In particular, this tag has a `skos:narrower` relation with the tag *metaux lourds* (heavy metal) that is used by 75 actors in the initial folksonomy.

| Tag | Inititial folksonomy | TagP | SemTagP(env, energ, model) |
|---|---|---|---|
| adsorption | 54 | 58 | 15.<br>This tag has 1 non pertinent `skos:narrower` relation with *absorption spectroscopy* (only 2 users) |
| Evaluation<br>(evaluation) | 54 | 4 | 0.<br>This tag has 0 `skos:narrower` tags. |
| Transport | 51 | 1 | 0<br>This tag has 28 `skos:narrower` tags. *transports* `skos:narrower` *transport*. |
| Métaux<br>(metal) | 51 | 2 | 87<br>This tag has 14 `skos:narrower` tags. |
| photovoltaïque<br>(photovoltaic) | 49 | 5 | 0<br>2 `skos:narrower` tags. |
| Innovation | 48 | 0 | 0<br>6 `skos:narrower` tags. |
| Recyclage<br>(recycling) | 48 | 8 | 9<br>2 `skos:narrower` tags. |

**Figure 65.** Comparison of the of the size of communities labelled with 7 tags (used by a similar number of actors in the initial folksonomy) in the initial folksonomy, with TagP and SemTagP (env, energ, model).

The Figure 66 presents a visualization of the ADEME social network with the tags of the communities output by SemTagP(env, energ, model). We used a graph visualization tool, GEPHI, with a force layout. The size of the nodes is proportional to their degrees, and the size of the tags is proportional to the size of the labeled communities. Groups of densely linked actors are gathered around few tags, which highlight the efficiency of the algorithm at partitioning the network. Moreover, communities that are labeled with tags representing related topics are close in the visualization, which enable us to build thematic area of the network using the labeling of the communities.

In Figure 68, communities displayed in framed area are respectively labeled with tags related to: pollution (1), sustainable development (2), energy (3), chemistry (4), air pollution (5), metals (6), biomass (7), and wastes (8). For instance, the area 3 contains tags related to energy production and consumption with the tags *energie* (energy), *silicium*, *solaire* (solar), *moteur* (motor), *bâtiment* (building) and *transports*. This observation shows that SemTagP labeled closest communities with related labels.

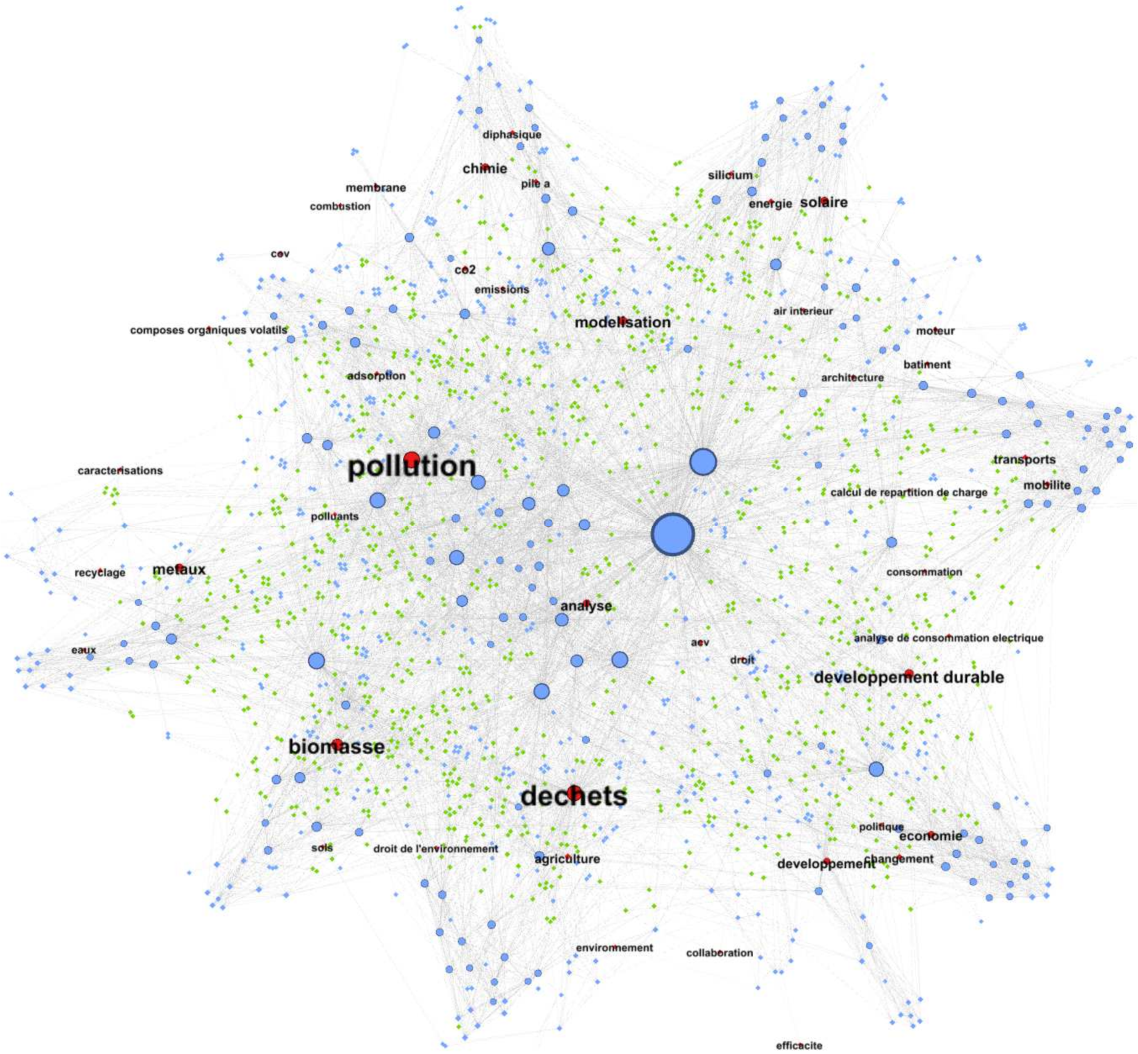


**Figure 66.** Ph.D. social network of the ADEME with tags labeling the communities obtained with SemTagP(env, energ, model). Red, blue and green nodes are respectively the tags, the ADEME's engineers and the academic supervisors.

Figure 68 to Figure 96 present a detailed view of the main communities of the 8 thematic areas highlighted by the Figure 67. In particular, we focus on the 30 biggest communities obtained with SemTagP(env, energy, model) that are described in tha table of Figure 64.

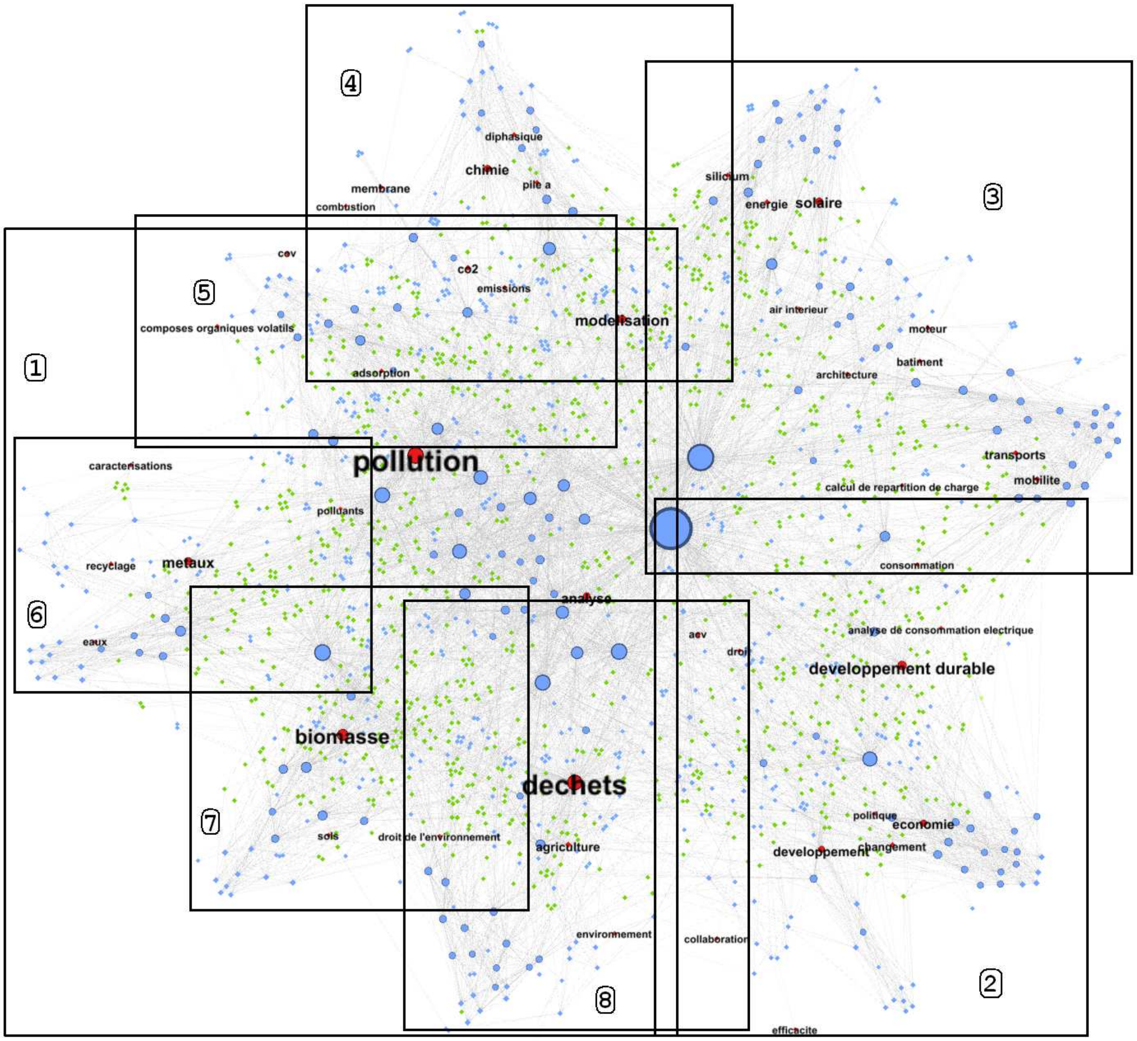


**Figure 67.** Decomposition in thematic areas of the visualization of Figure 67. The communities of the areas are labelled with tags related to: pollution (1), sustainable development (2), energy (3), chemistry (4), air pollution (5), metals (6), biomass (7), wastes (8).

The Figure 68 presents a focus of the thematic area that groups tags related to pollution, and Figure 69 to Figure 78 highlight the different communities that are embedded in this area. The Figure 69 shows all the actors of this area that are members of the biggest community that is described with the broader tag of this area: *pollution*. This community is the main one of this area but many other tags are important and represent more specialized communities. In fact, this area is the biggest and can be decomposed into 4 other smaller areas having tags related to different pollution topics: air pollution, metals, biomass and waste (see Figure 68).

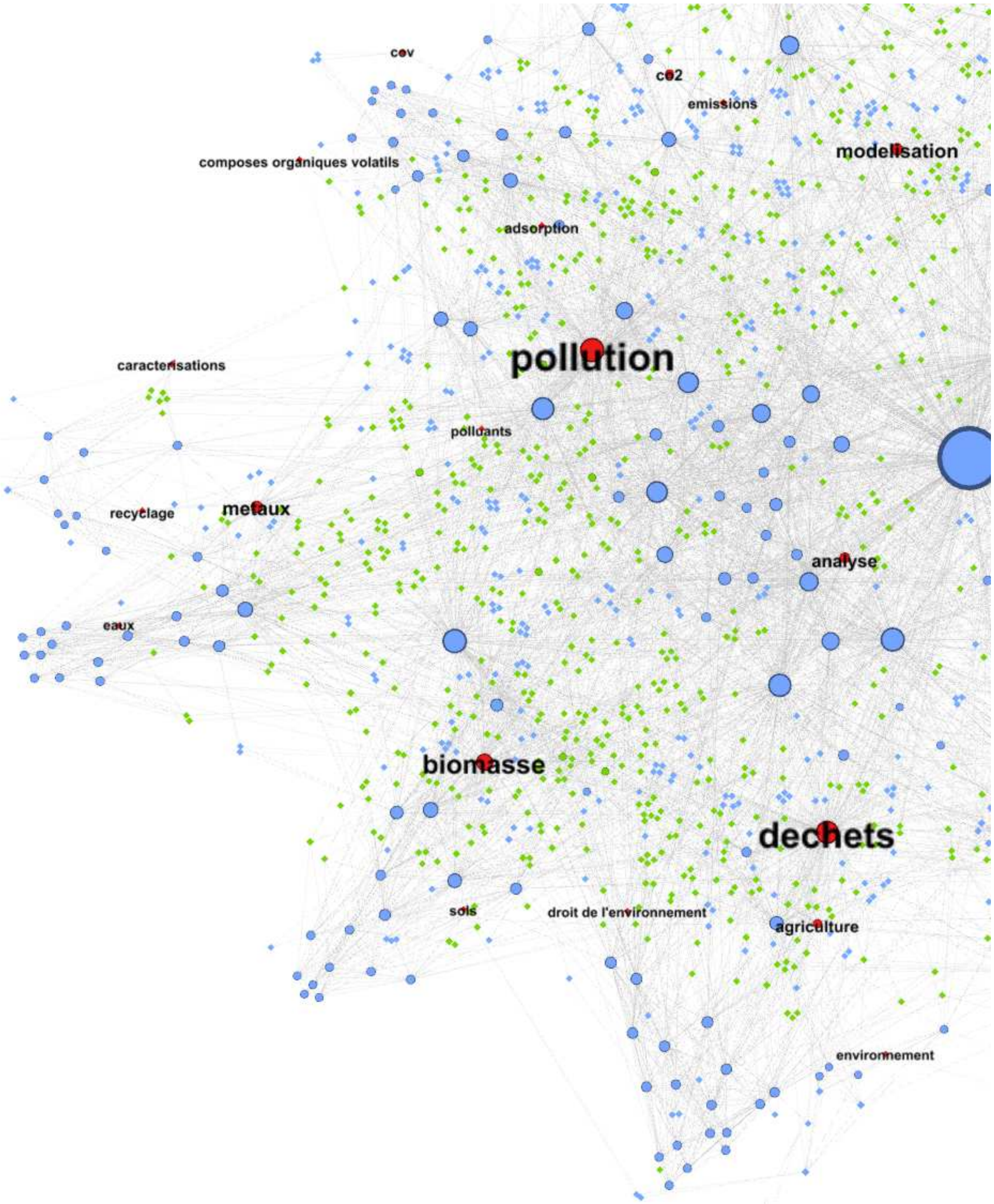


**Figure 68.** Visualization of the thematic area composed of communities labeled with the tag pollution.

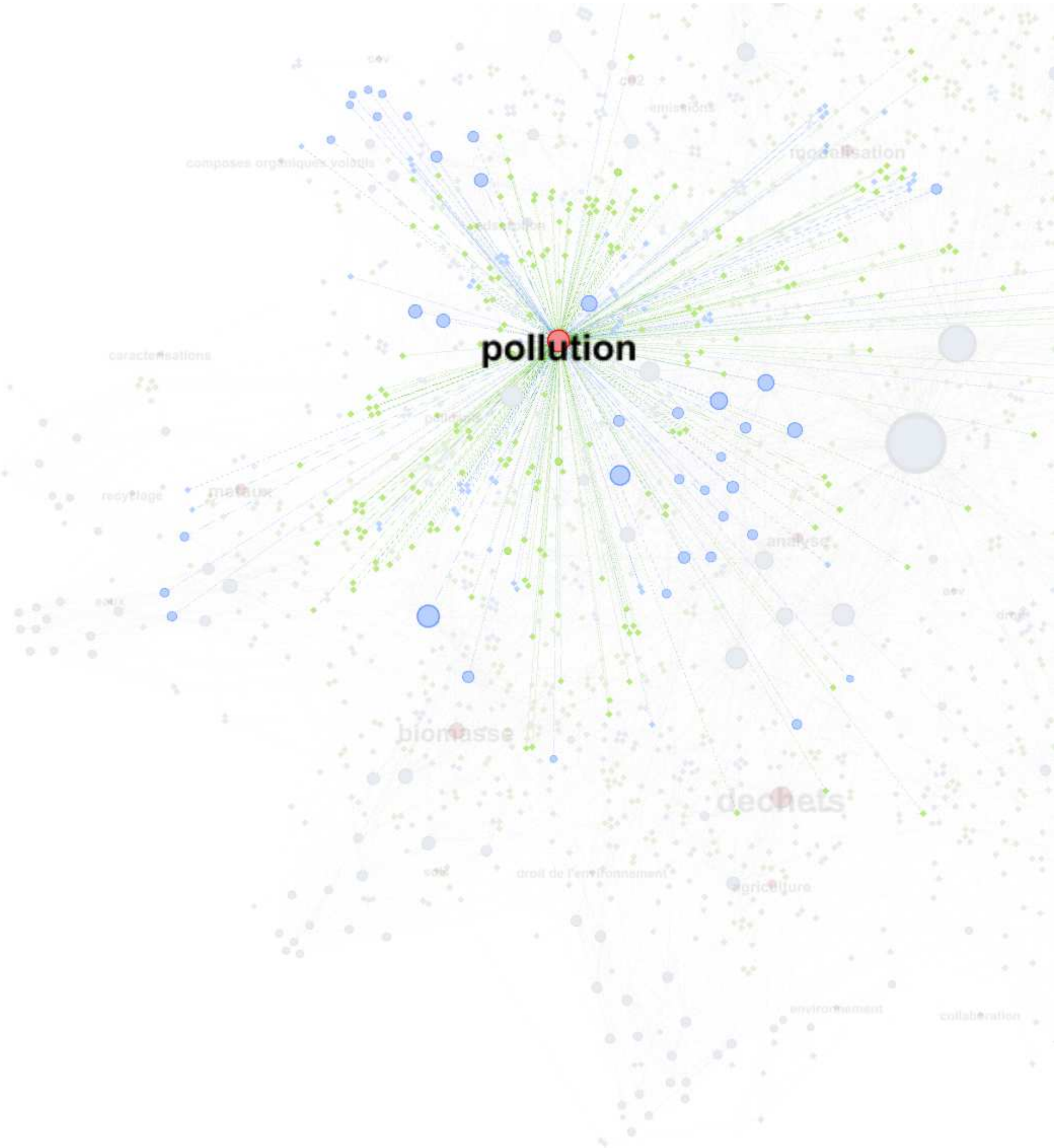


**Figure 69.** Visualization of the community labeled with the tag pollution.

Figure 70 and Figure 71 highlight very close communities labeled with tags that could not be considered as semantically related: *dechets* (wastes) and *agriculture*. However, the agriculture is an important source of soil pollution due to all the wastes that are produced by this activity. Consequently these two communities are related and very close in visualization.

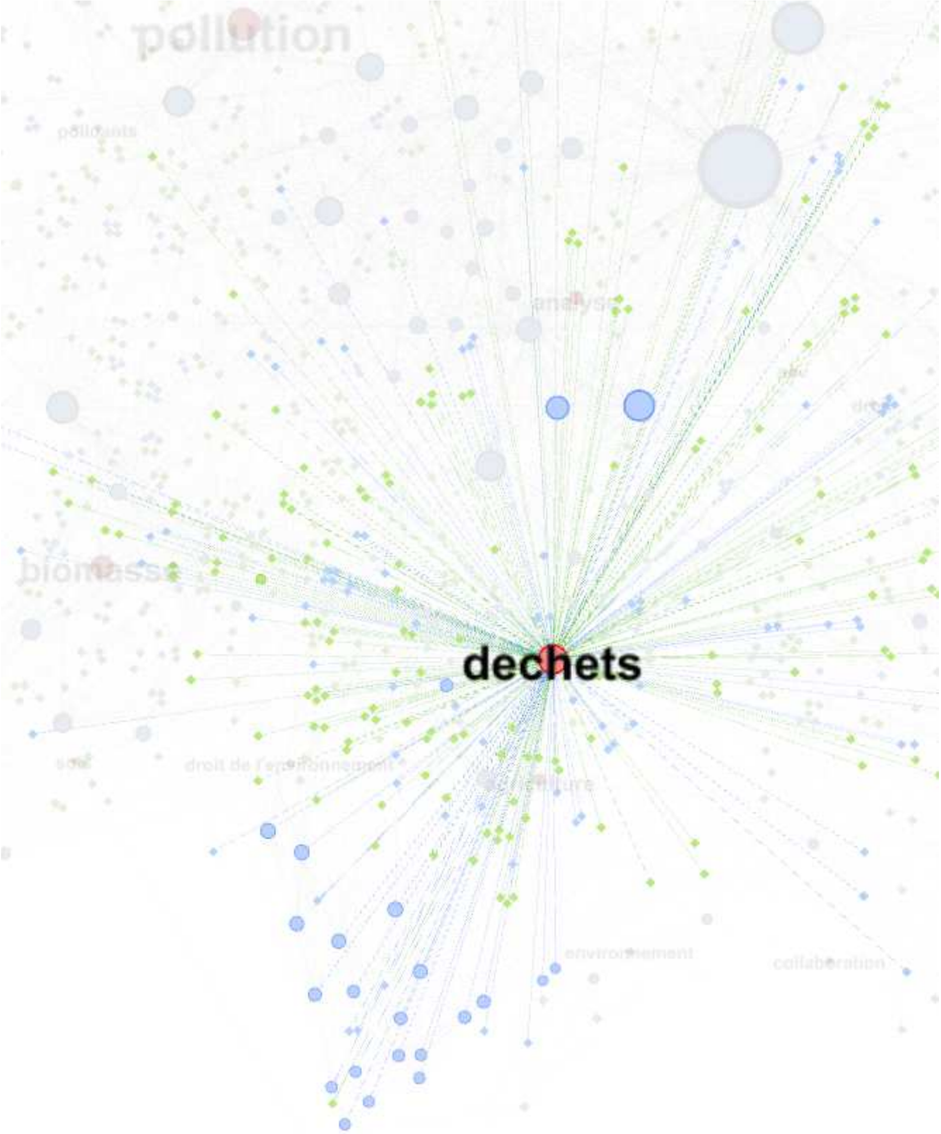


**Figure 70.** Visualization of the community labeled with the tag *dechets* (wastes).

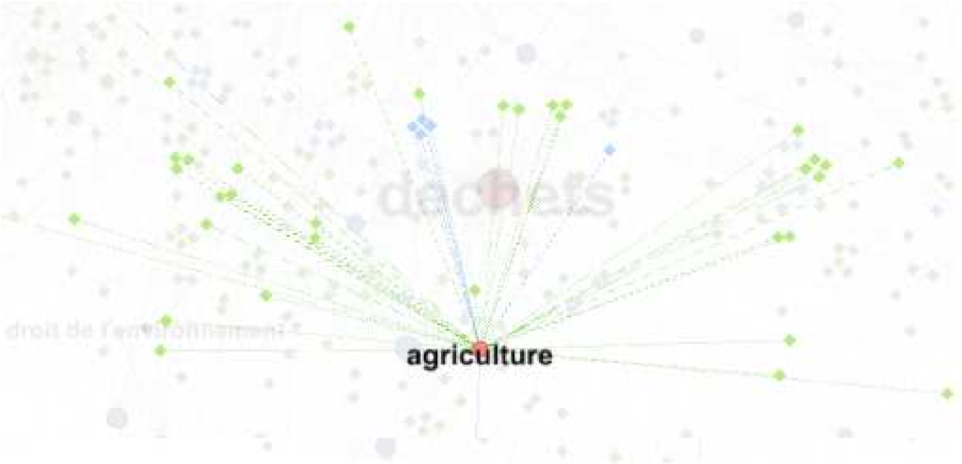


**Figure 71.** Visualization of the community labeled with the tag agriculture.

Figure 72 and Figure 73 propose the visualization of two close communities: *metaux* (metals) and *recyclage* (recycling). Metals are one of the main materials that recycled and recycling them is also an important challenges for the environment.

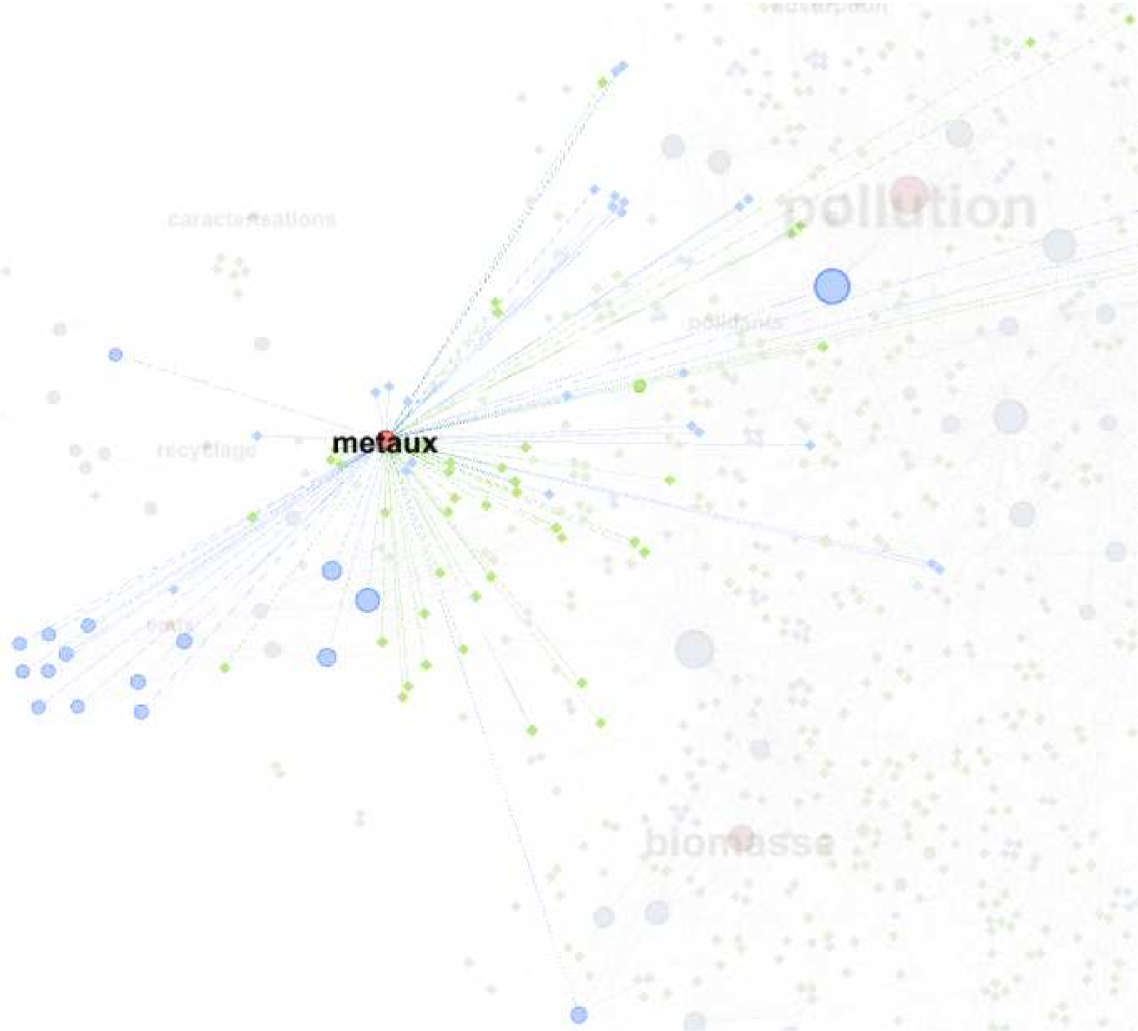


**Figure 72.** Visualization of the community labeled with the tag *metaux* (metals).

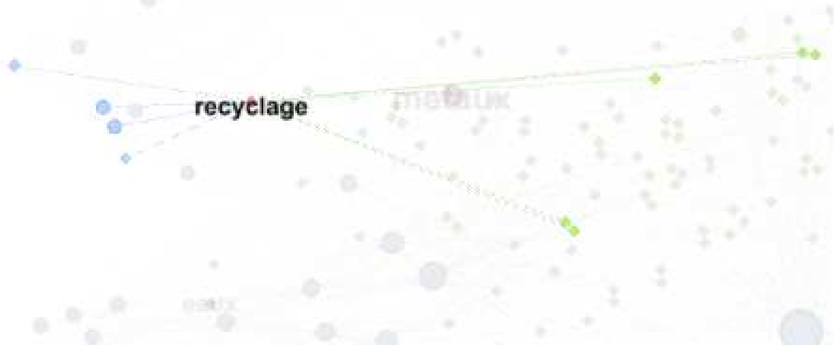


**Figure 73.** Visualization of the community labeled with the tag *recyclage* (recycling).

Figure 74 represents the community labelled with the tag biomass. This is the third biggest community and it is closely related to wastes and metals in the visualization. This community could probably be decomposed into two communities to reflect the point of view of the notion of biomass:

- In ecology, the biomass is “the mass of living biological organisms in a given area or ecosystem at a given time”[40].
- As a renewable energy source, the biomass is “biological material from living, or recently living organisms, such as wood, waste, (hydrogen) gas, and alcohol fuels”[41].

Tags that are `skos:narrower` than *biomass*, reflect both points of view with for instance the tags *gazefication de la biomasse* (biomass gasification) or *biomass algale* (algal biomass).

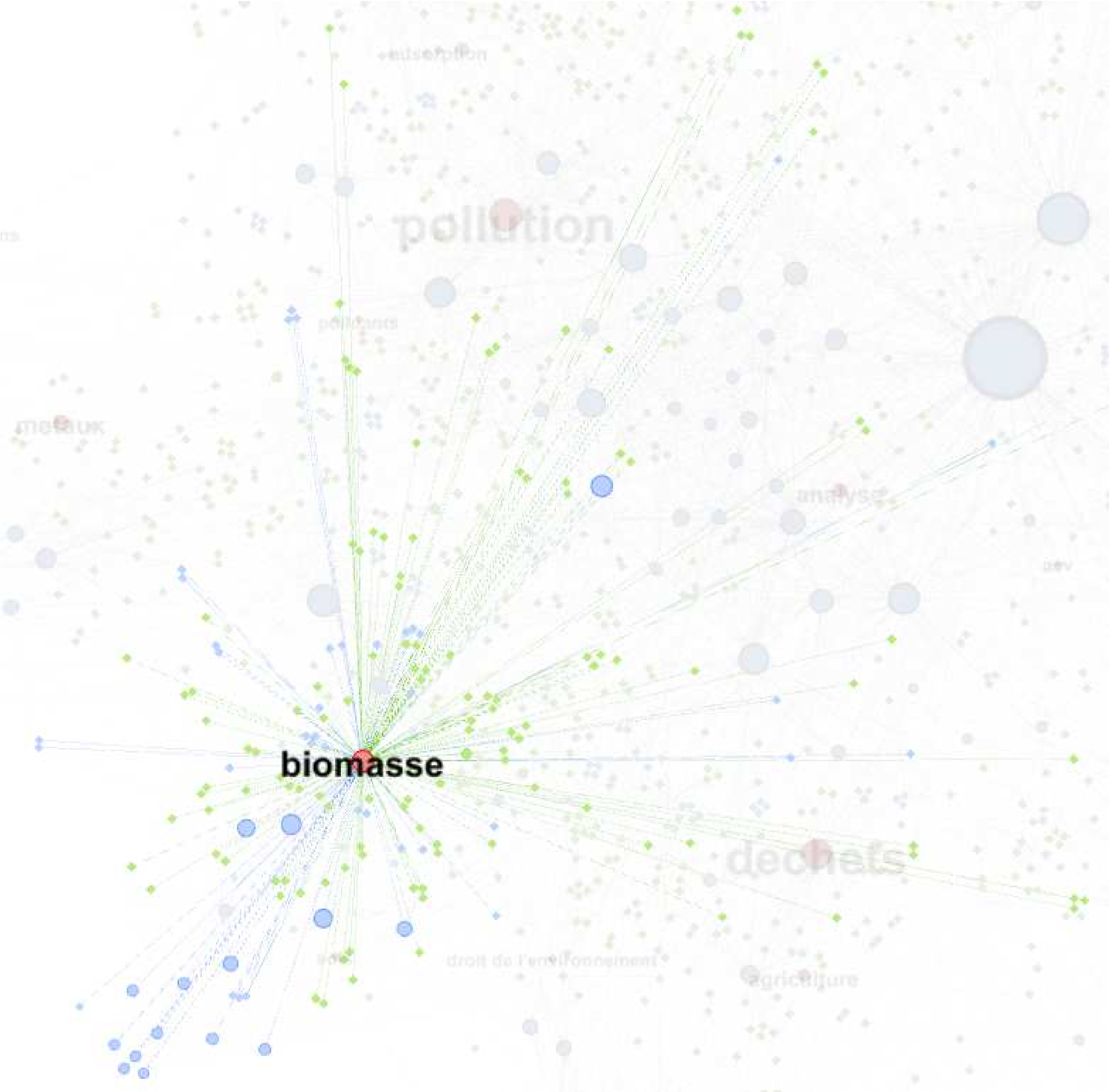


**Figure 74.** Visualization of the community labeled with the tag *biomasse* (biomass).

Figure 75 to Figure 79 represent communities labeled with tags related to air pollution. In particular Figure 75 and Figure 76 represent 2 close communities that are labeled

[40] http://en.wikipedia.org/wiki/Biomass_(ecology)

[41] http://en.wikipedia.org/wiki/Biomass

with tags representing the same subject *composes organiques volatils* (organic volatile compounds) and *cov* which is simply an acronym of the first one. This observation could be used to semantically relate these tags and merge these two communities.

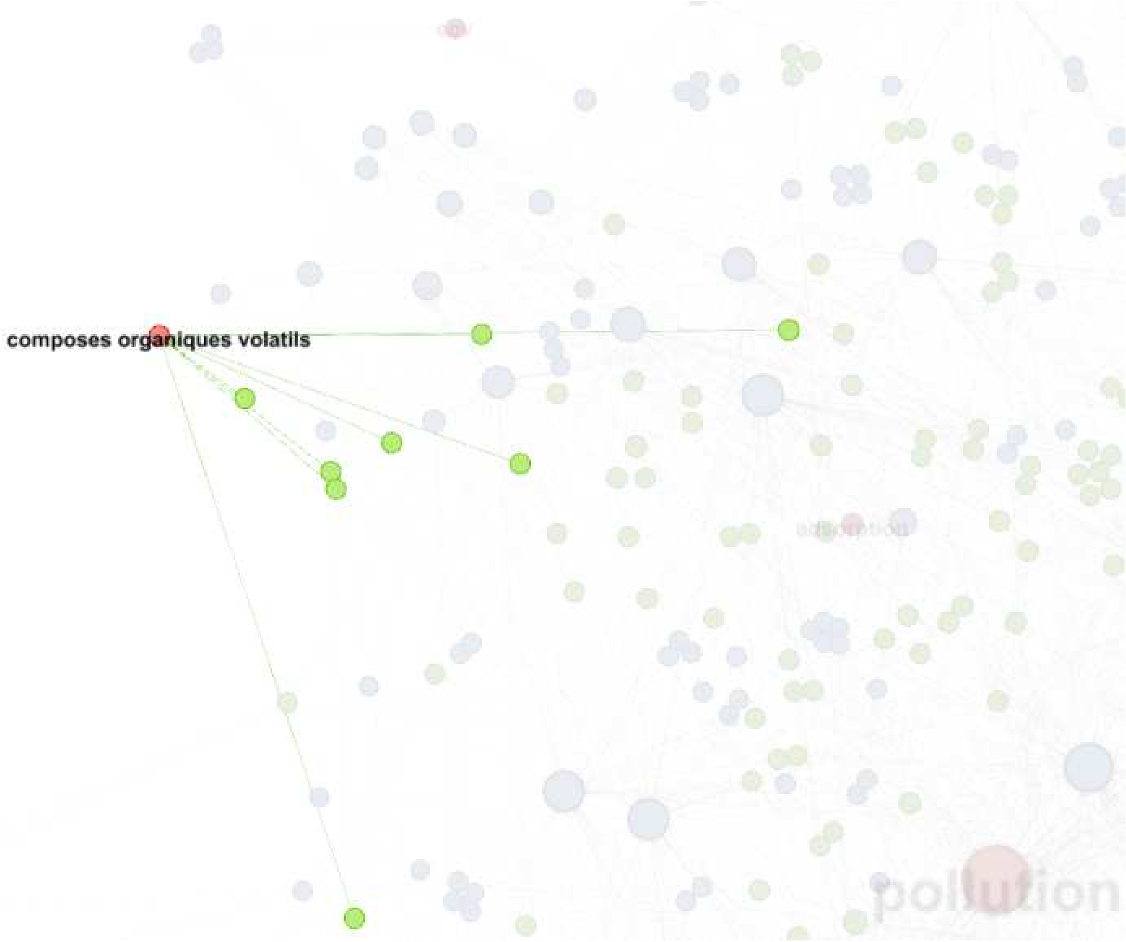


**Figure 75.** Visualization of the community labeled with the tag *composes organiques volatils* (volatile organic compounds).

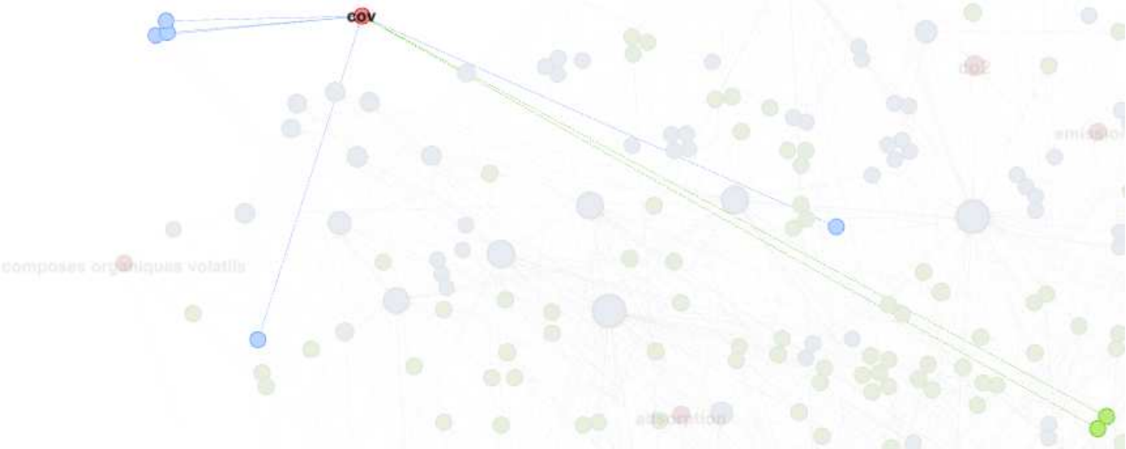


**Figure 76.** Visualization of the community labeled with the tag *cov*, which stands for *composes organiques volatils* (volatile organic compounds).

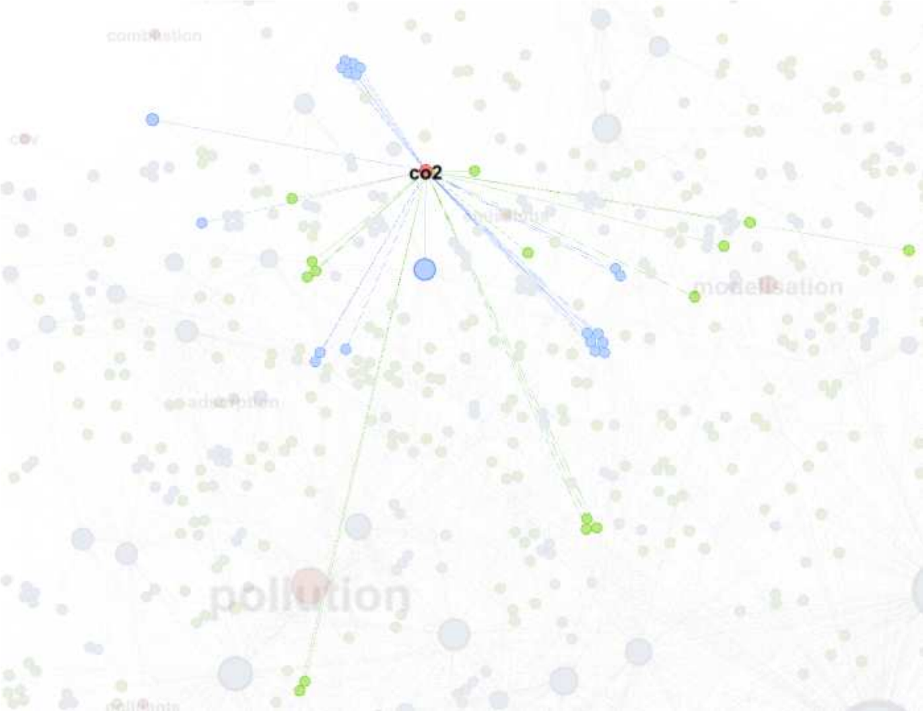


**Figure 77.** Visualization of the community labeled with the tag *co2*.

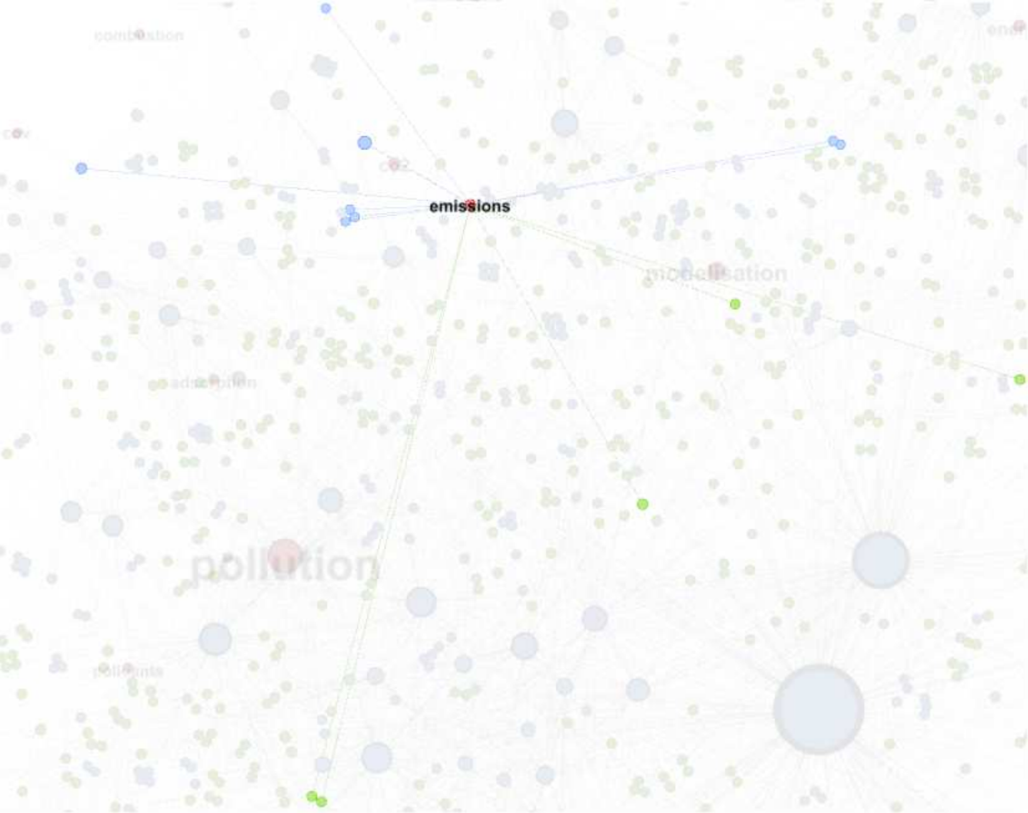


**Figure 78.** Visualization of the community labeled with the tag emissions.

Figure 77 and Figure 78 represent close communities labelled with tags that represent concepts that are frequently associated: *co2* and *emissions*. In this case we could probably also only consider one community about *co2 emissions*.

Figure 79 proposes a visualisation of the community labelled with the tag *adsorption*. The adsorption, which "is the adhesion of atoms, ions, biomolecules or molecules of gas, liquid, or dissolved solids to a surface"[42], is a process used in particular for capturing wastes in water or recovering gaseous pollutant emissions.

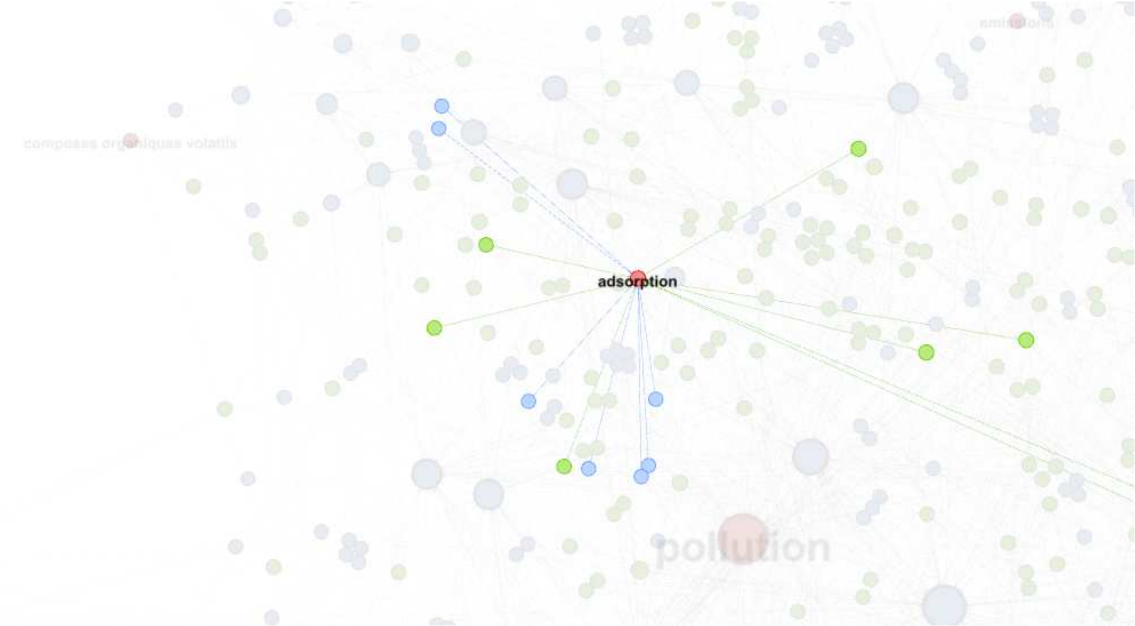


**Figure 79.** Visualization of the community labeled with the tag adsorption.

Figure 80 to Figure 84 represent communities labeled with tags related to sustainable development. Figure 81, Figure 82, Figure 83 and Figure 84 highlight the communities respectively labeled with the tags: *developpement durable* (sustainable development), *developpement* (development), *changement* (change), and *economie* (economy).

The community labeled with the tag *economie* (economy), has the particularity to group people working on two different topics representing the two notions represented by the term economy. Economy can be seen either as (1) "the quality of being efficient or frugal in using resources"[43], e.g. the tag *economie* is `skos:broader` than *economie d'energie* (energy economy), or (2) "the human activity that consists in producing, exchanging, distributing, and consuming goods and services"[43], e.g. the tag *economie* is `skos:broader` than *economie de l'environnement* (environment economy). However these two definitions are both related to sustainable development which consists in conducting economic and social developments that preserve environmental and social resources.

[42] http://en.wikipedia.org/wiki/Adsorption

[43] http://en.wikipedia.org/wiki/Economy_(disambiguation)

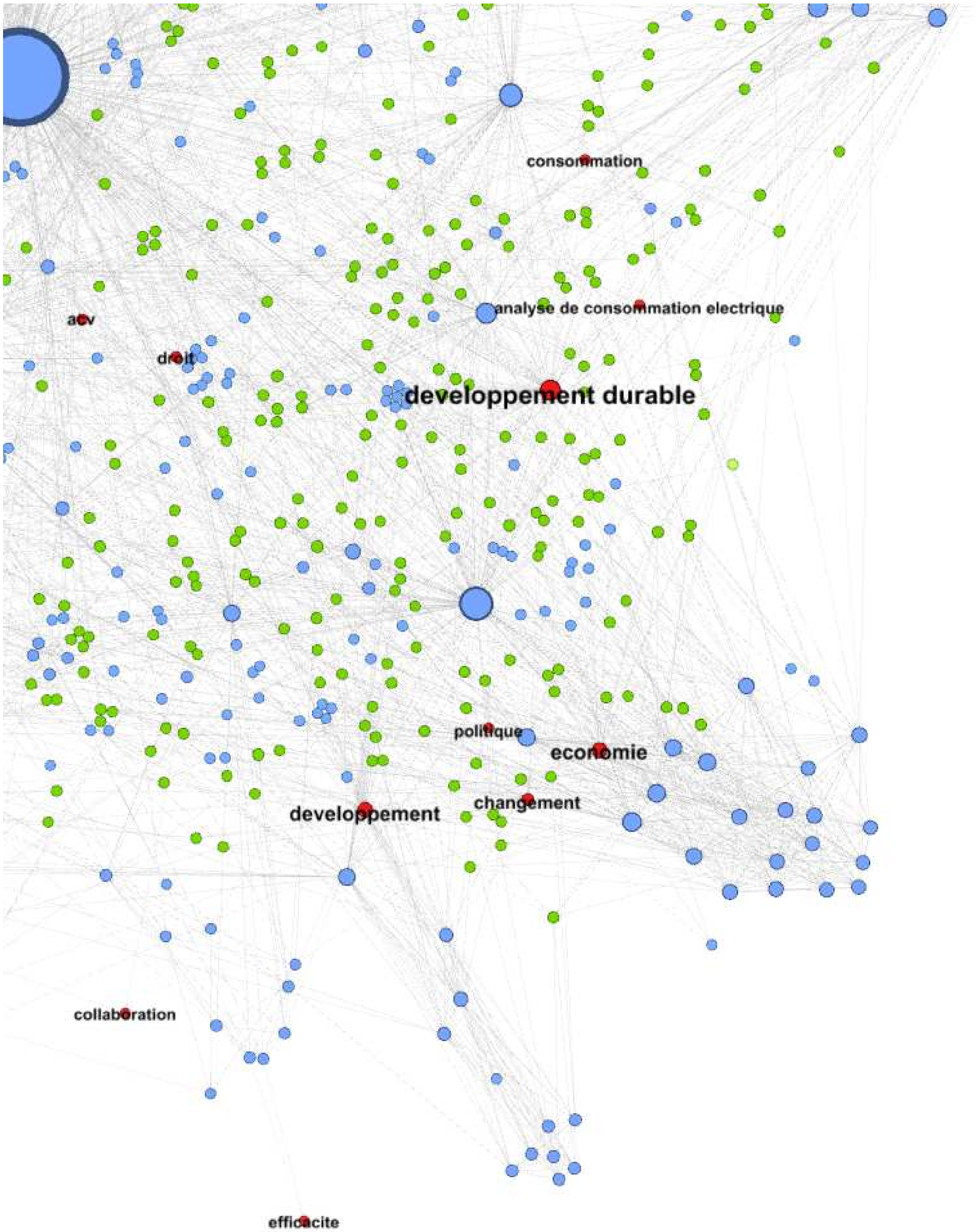


**Figure 80.** Visualization of the thematic area having communities labeled with tags related to sustainable development.

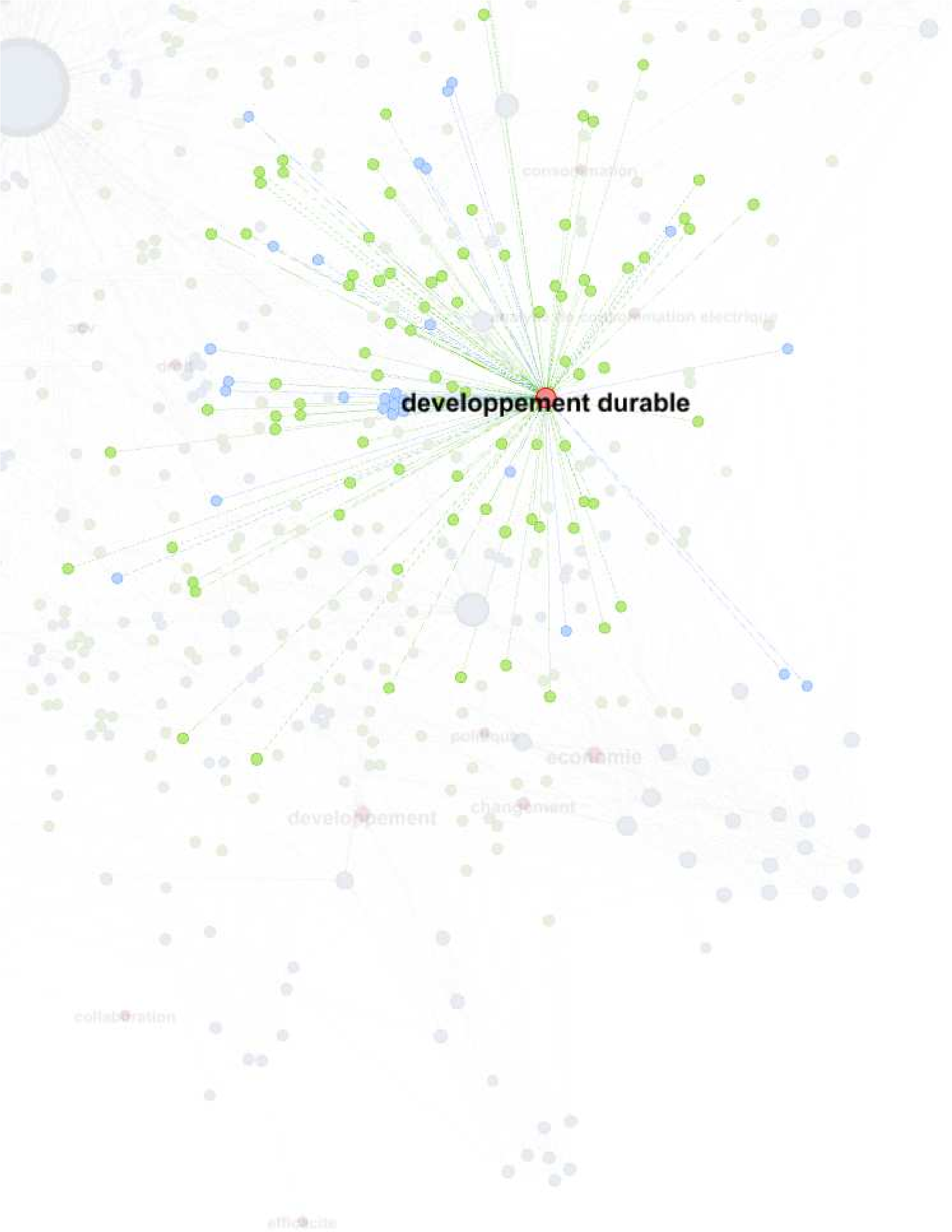


**Figure 81.** Visualization of the community labeled with the tag *developpement durable* (sustainable development).

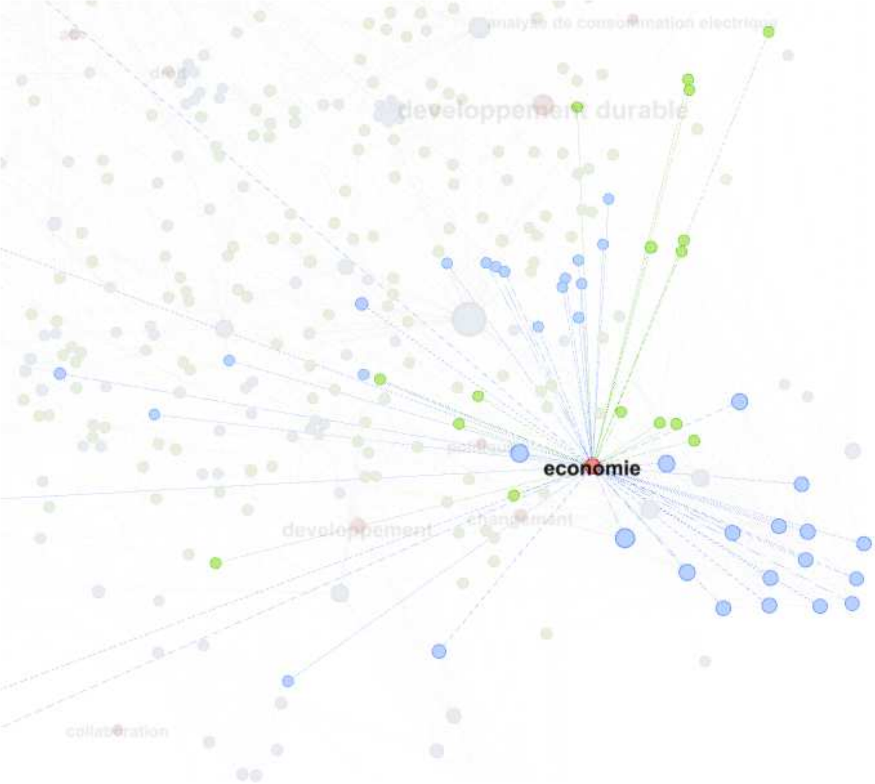


**Figure 82.** Visualization of the community labeled with the tag economie (economy).

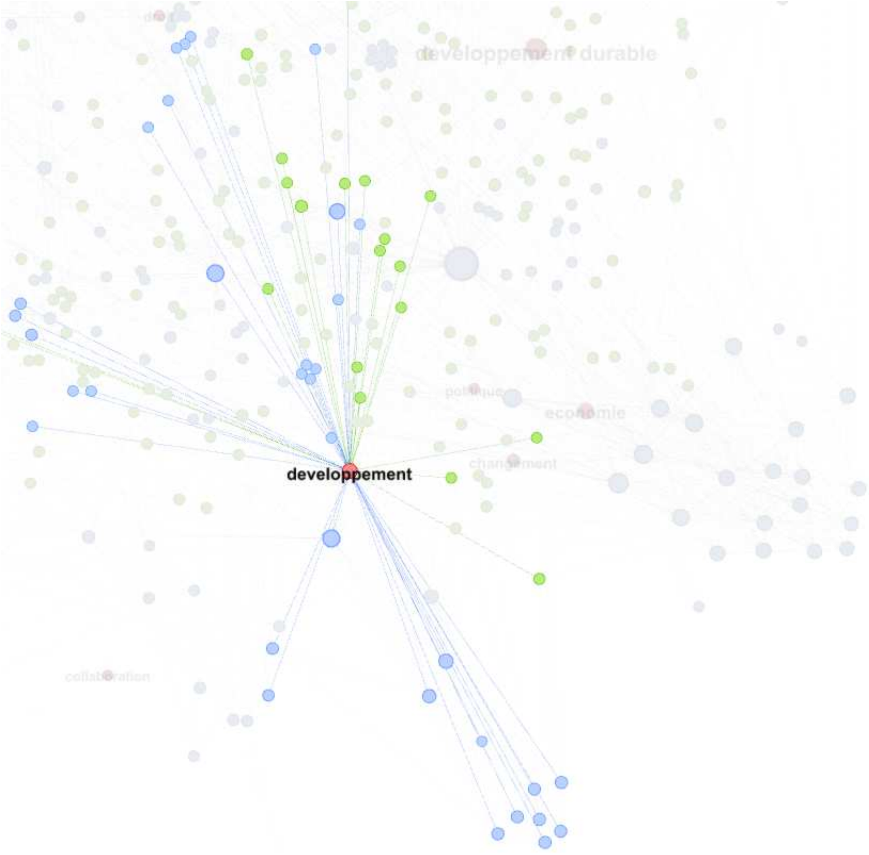


**Figure 83.** Visualization of the community labeled with the tag economie (economy).

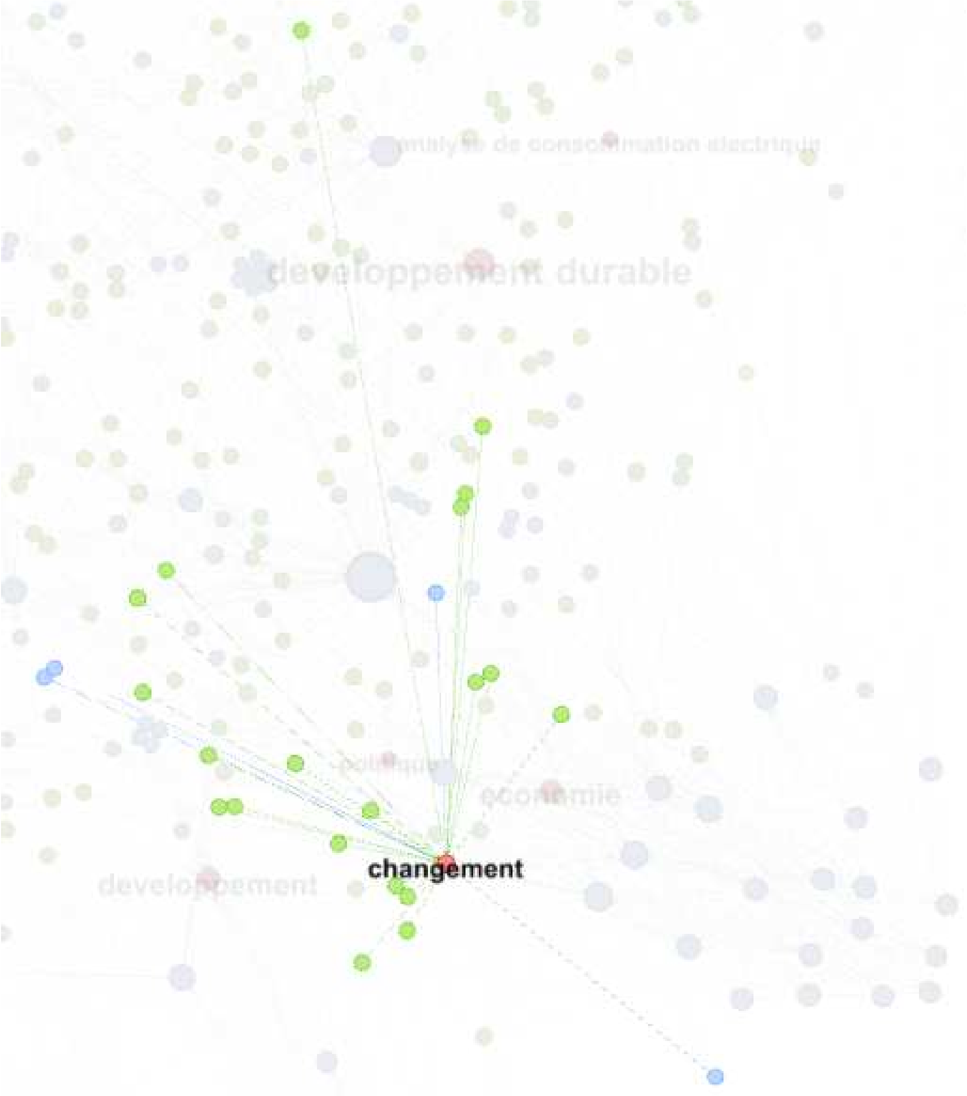


**Figure 84.** Visualization of the community labeled with the tag changement (change).

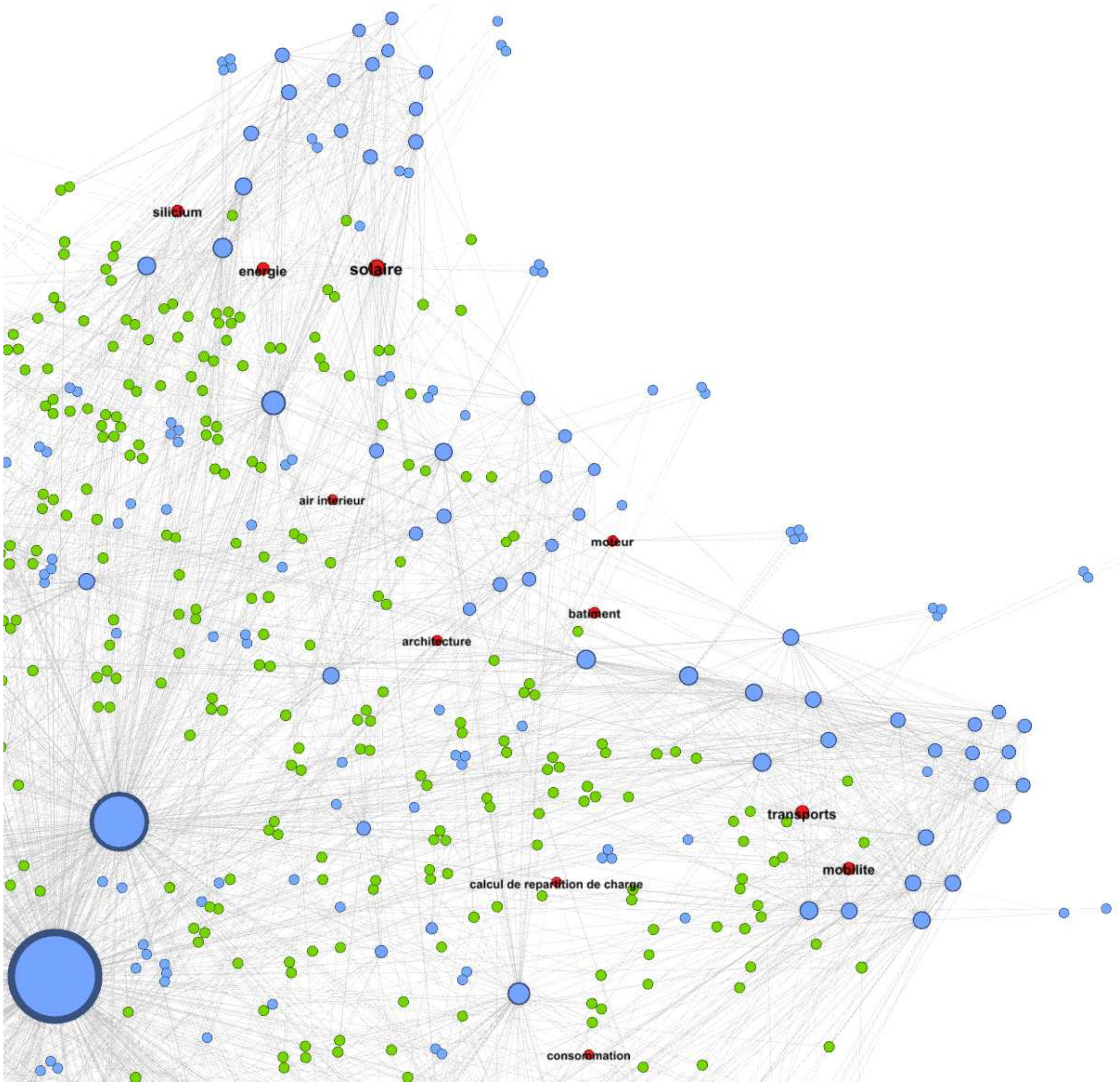


**Figure 85.** Visualization of the energy community, which is composed of tags representing topics like solar, building or transport.

Figure 85 to Figure 93 represent communities related to energy. Figure 85 proposes an overview of this area, which is mainly composed of tags related to solar energy and energy consumption. Figure 86, Figure 87 and Figure 88 show communities mainly related to the notion of solar energy with the tag *energie* (energy), *solaire* (solar) and *silicium* (used in photovoltaic cells). Figure 90 to Figure 93 highlight communities labeled with tags related to energy consumption: *moteur* (motor), *architecture*, *batiment* (building), *transports*, *mobilite* (mobility).

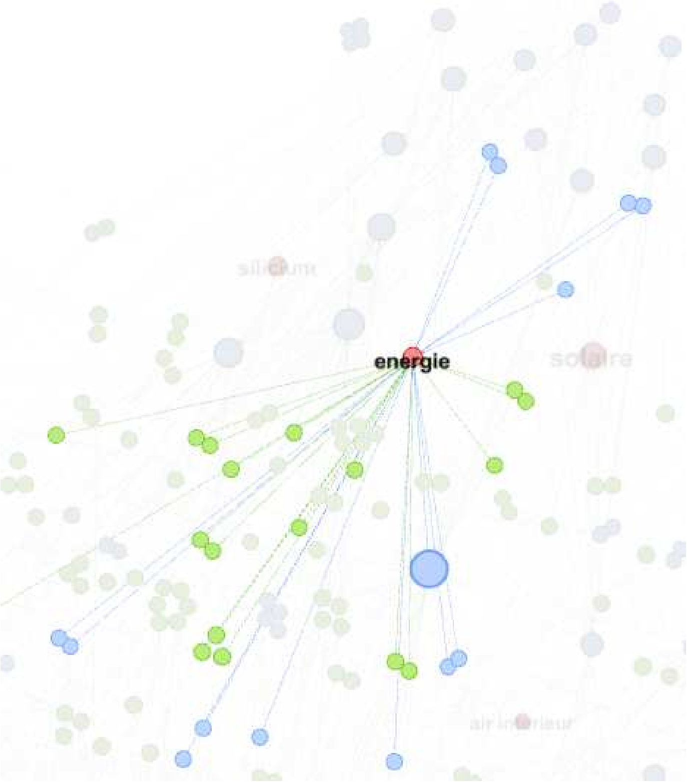


**Figure 86.** Visualization of the community labeled with the tag *energie* (energy)

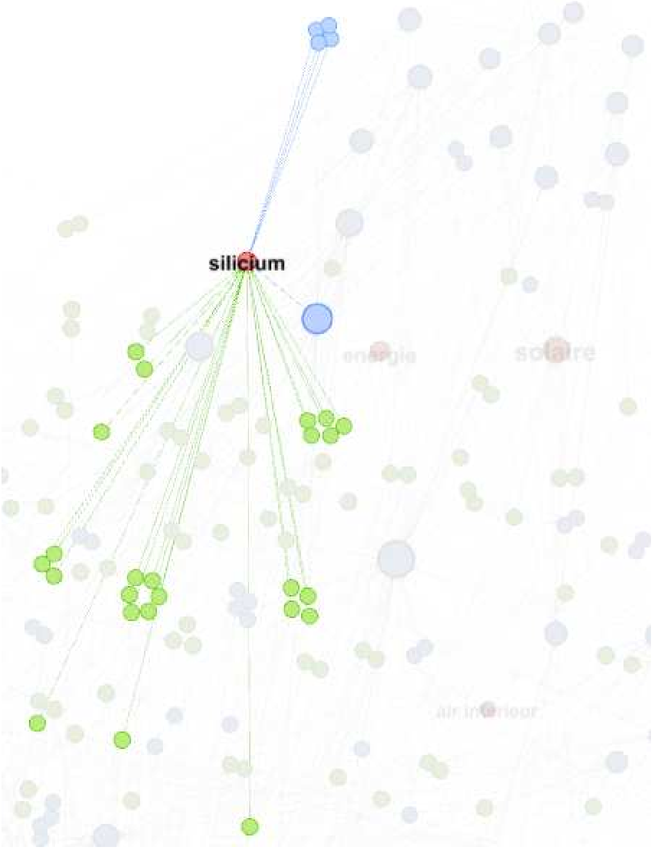


**Figure 87.** Visualization of the community labeled with the tag *silicium*.

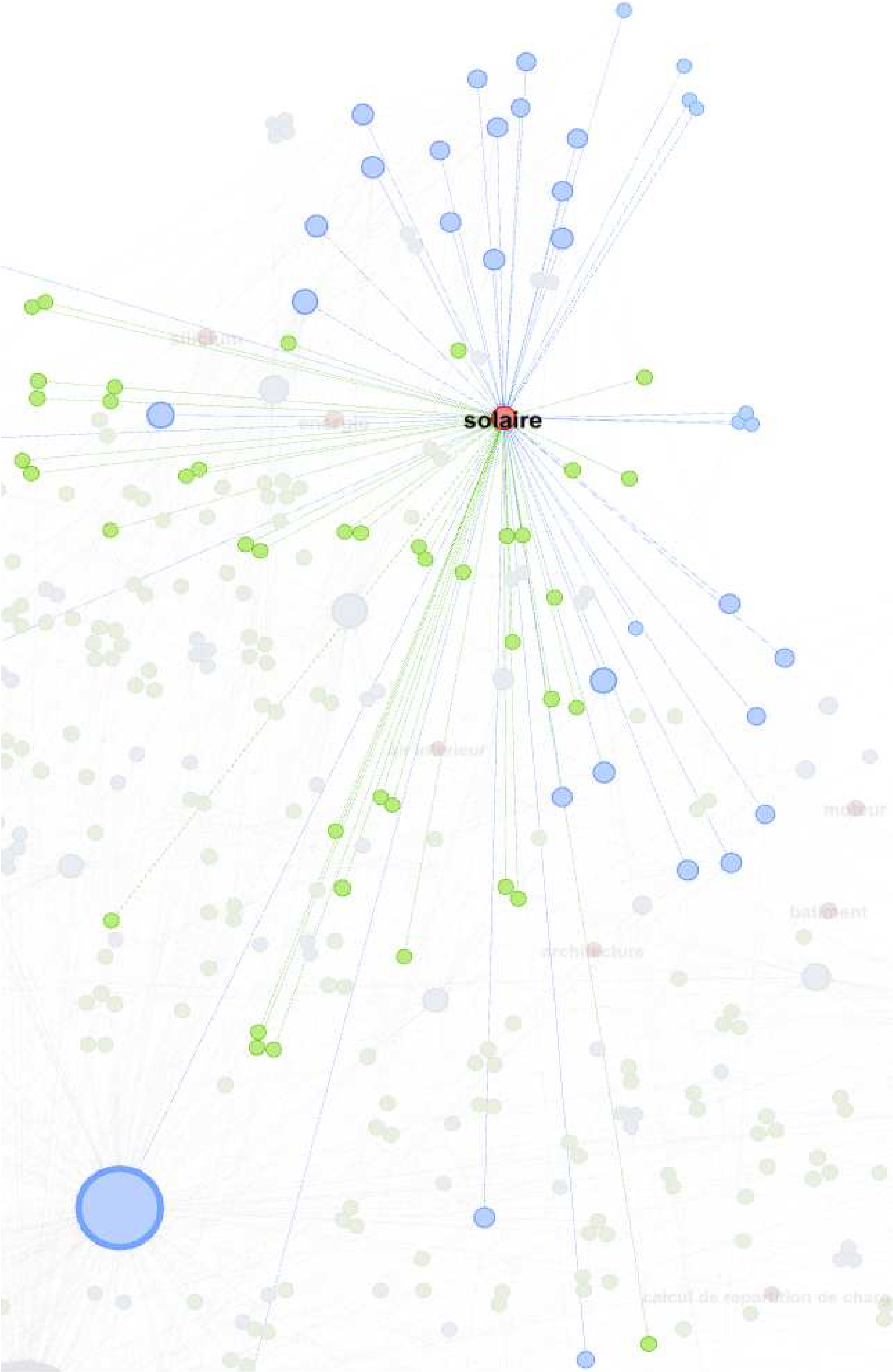


**Figure 88.** Visualization of the community labeled with the tag *solaire* (solar)

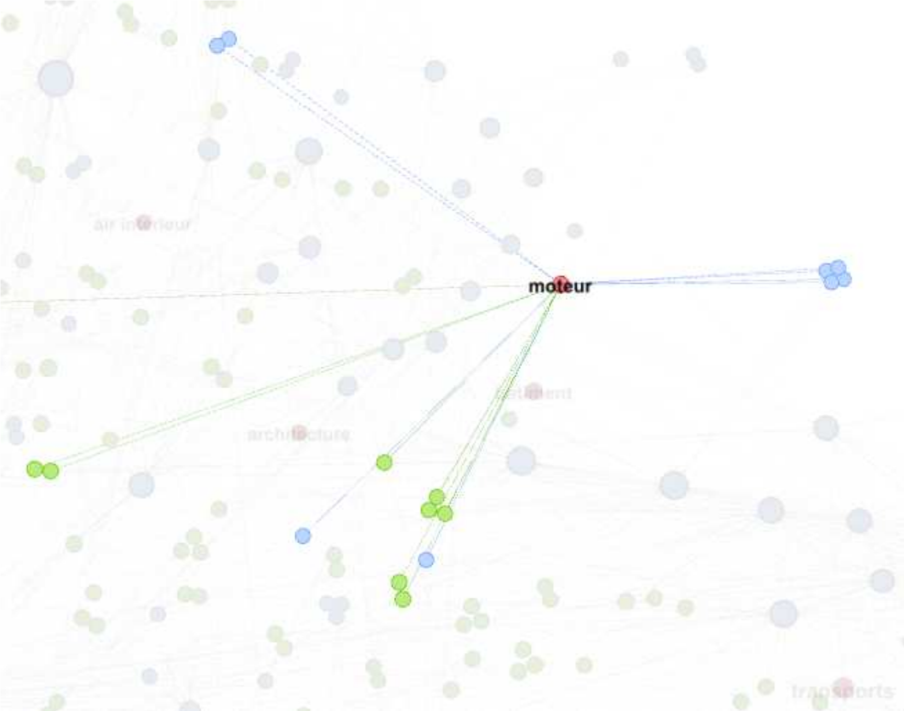


**Figure 89.** Visualization of the community labeled with the tag moteur (motor).

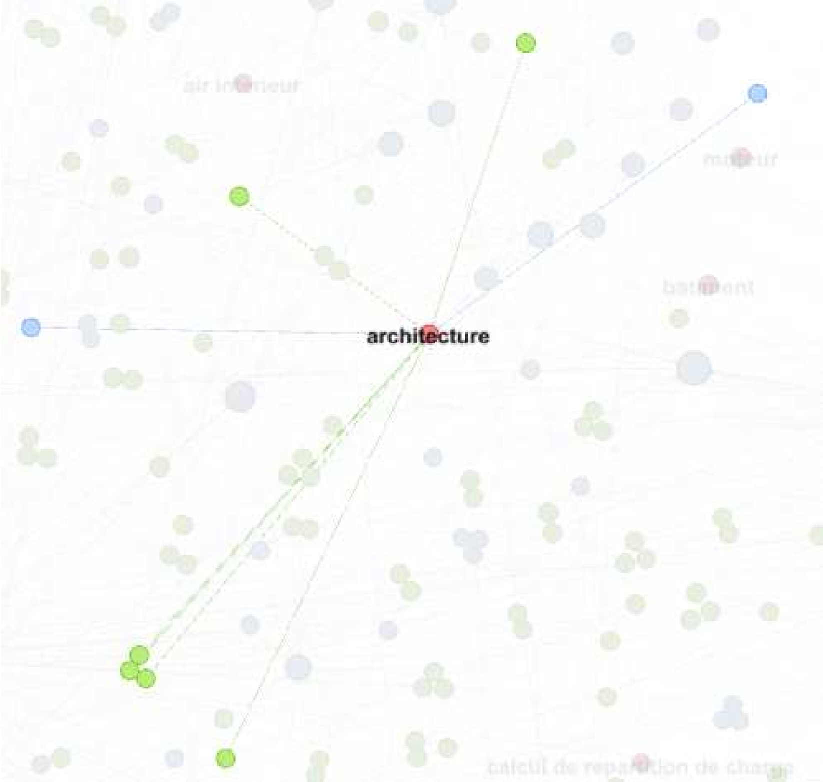


**Figure 90.** Visualization of the community labeled with the tag architecture.

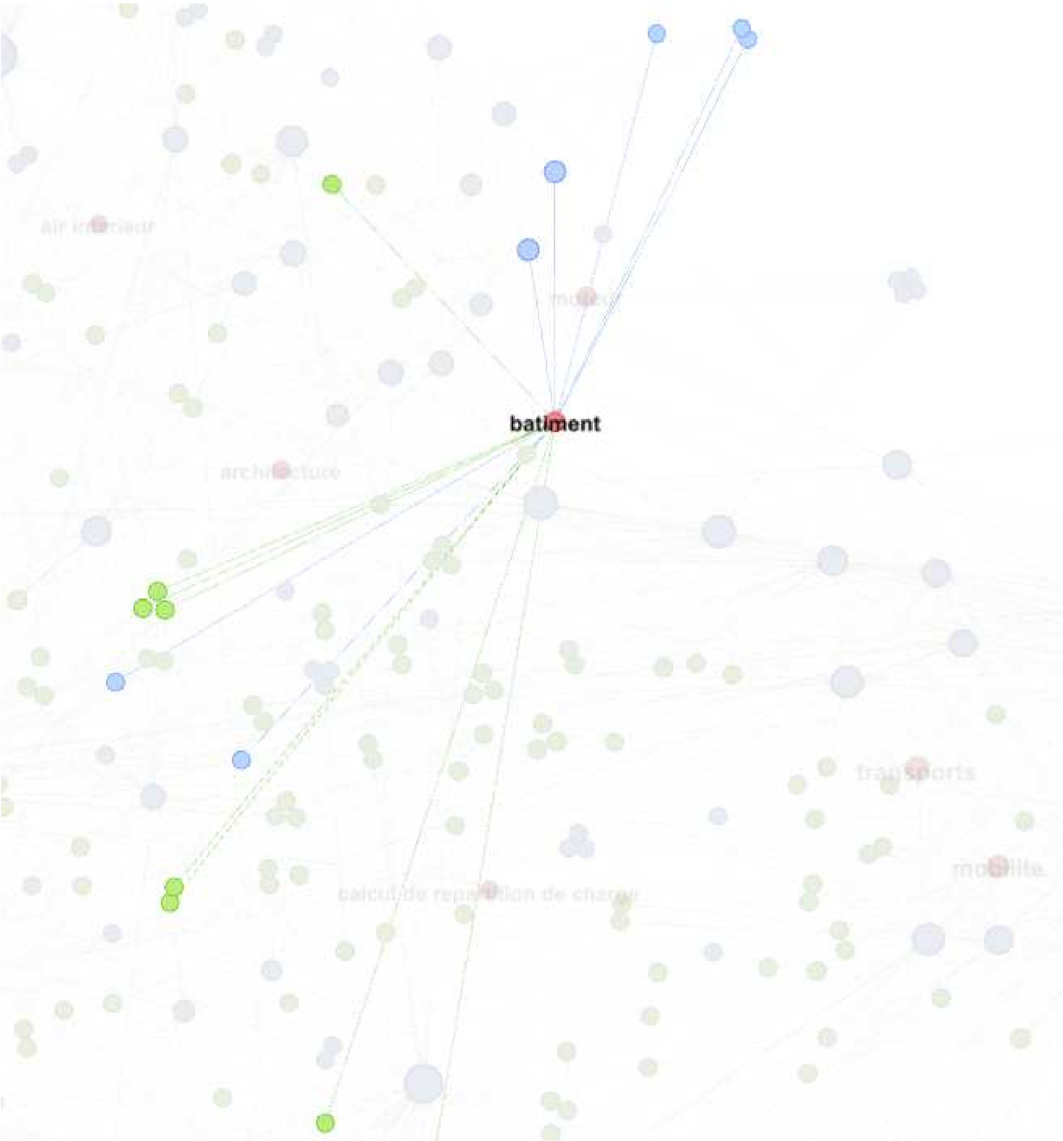


**Figure 91.** Visualization of the community labeled with the tag *batiment* (building).

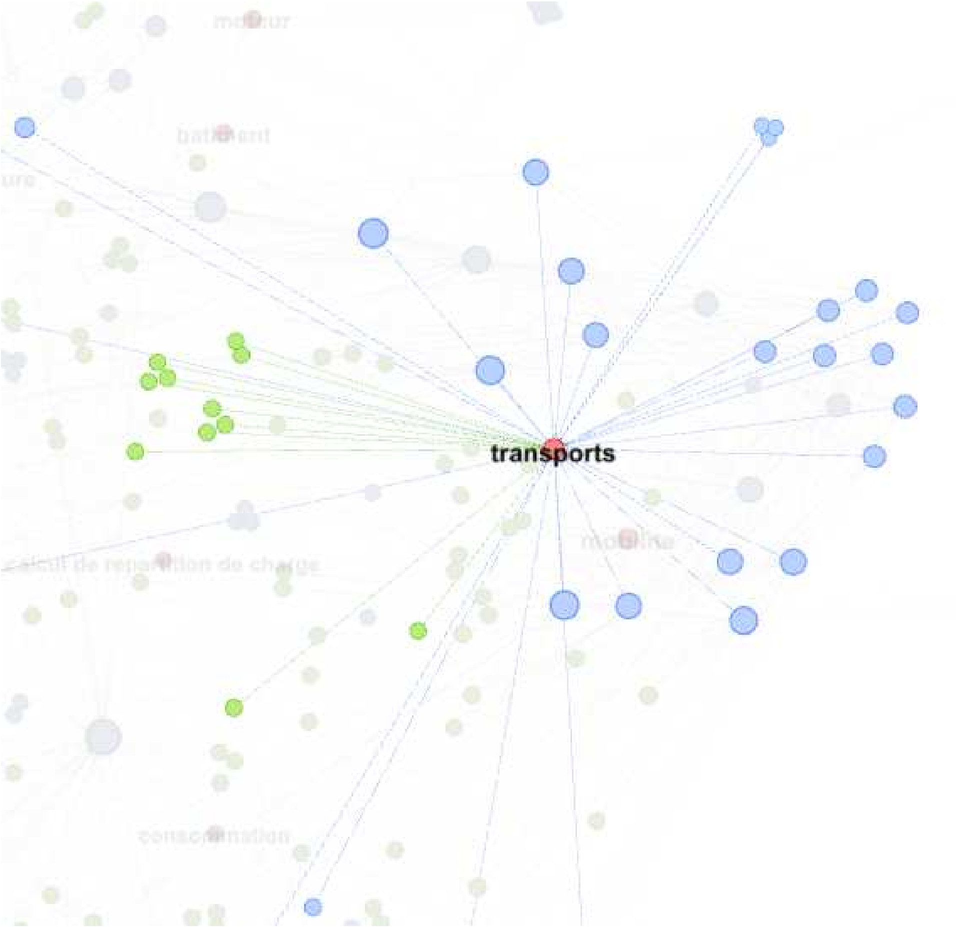


**Figure 92.** Visualization of the community labeled with the tag *transports*.

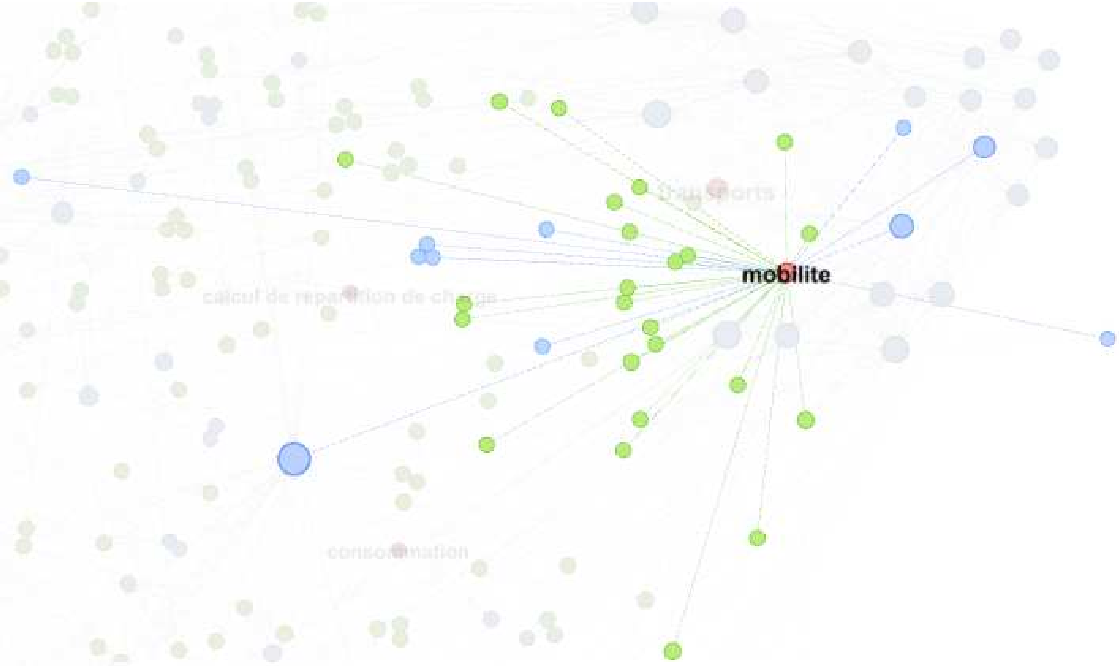


**Figure 93.** Visualization of the community labeled with the tag *mobilite* (mobility).

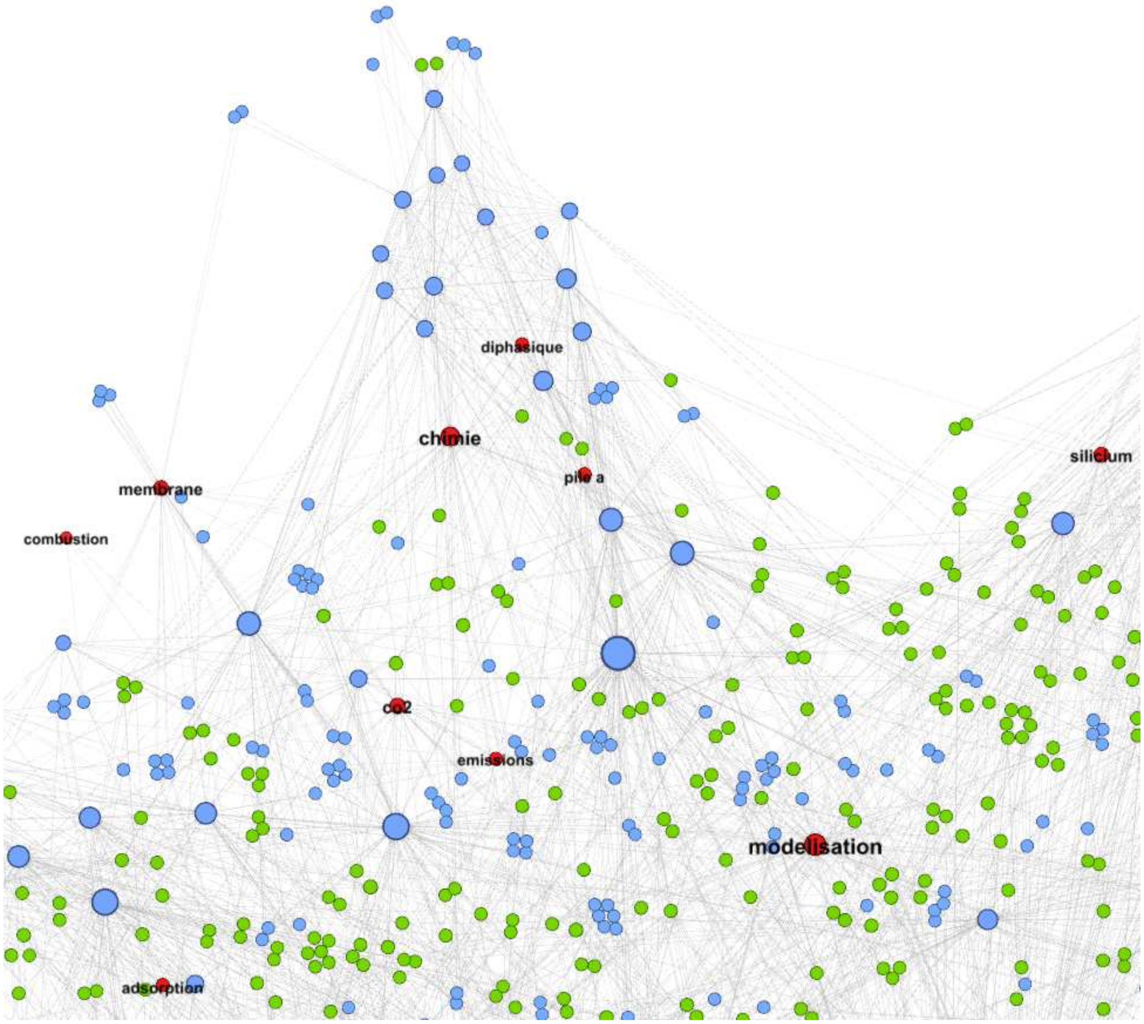


**Figure 94.** Visualization of the community labeled with the tags related to chemistry.

Figure 94 to Figure 97 represent communities related to chemistry. Figure 94 shows an overview of this area in which shared tags are also present in the area related to air pollution and energy (see Figure 67), such as *adsorption* (Figure 79) and *silicium* (Figure 87). Figure 95, Figure 96, and Figure 97 focus on the communities labeled with the tags *chimie* (chemistry), *pile* (battery) and *diphasique* (diphasic)

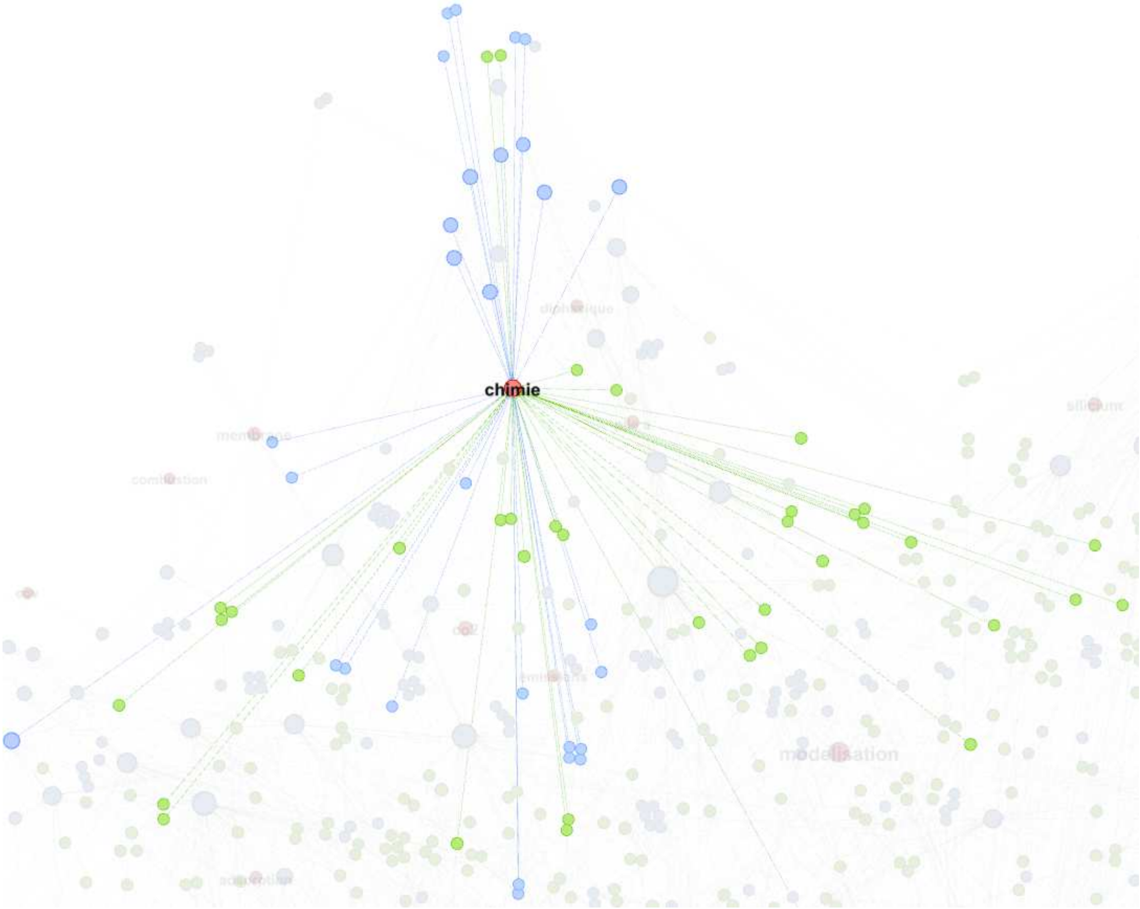


**Figure 95.** Visualization of the community labeled with the tag *chimie* (chemistry).

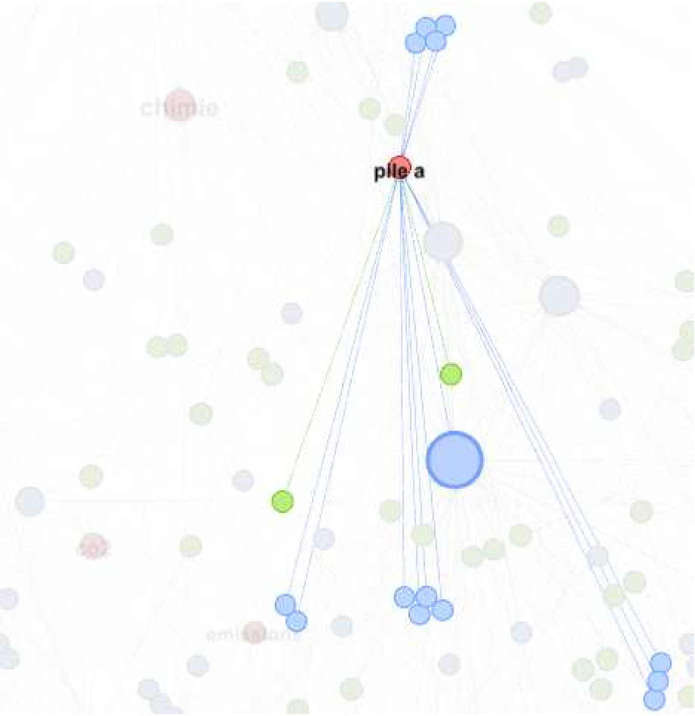


**Figure 96.** Visualization of the community labeled with the tag *pile a* (battery).

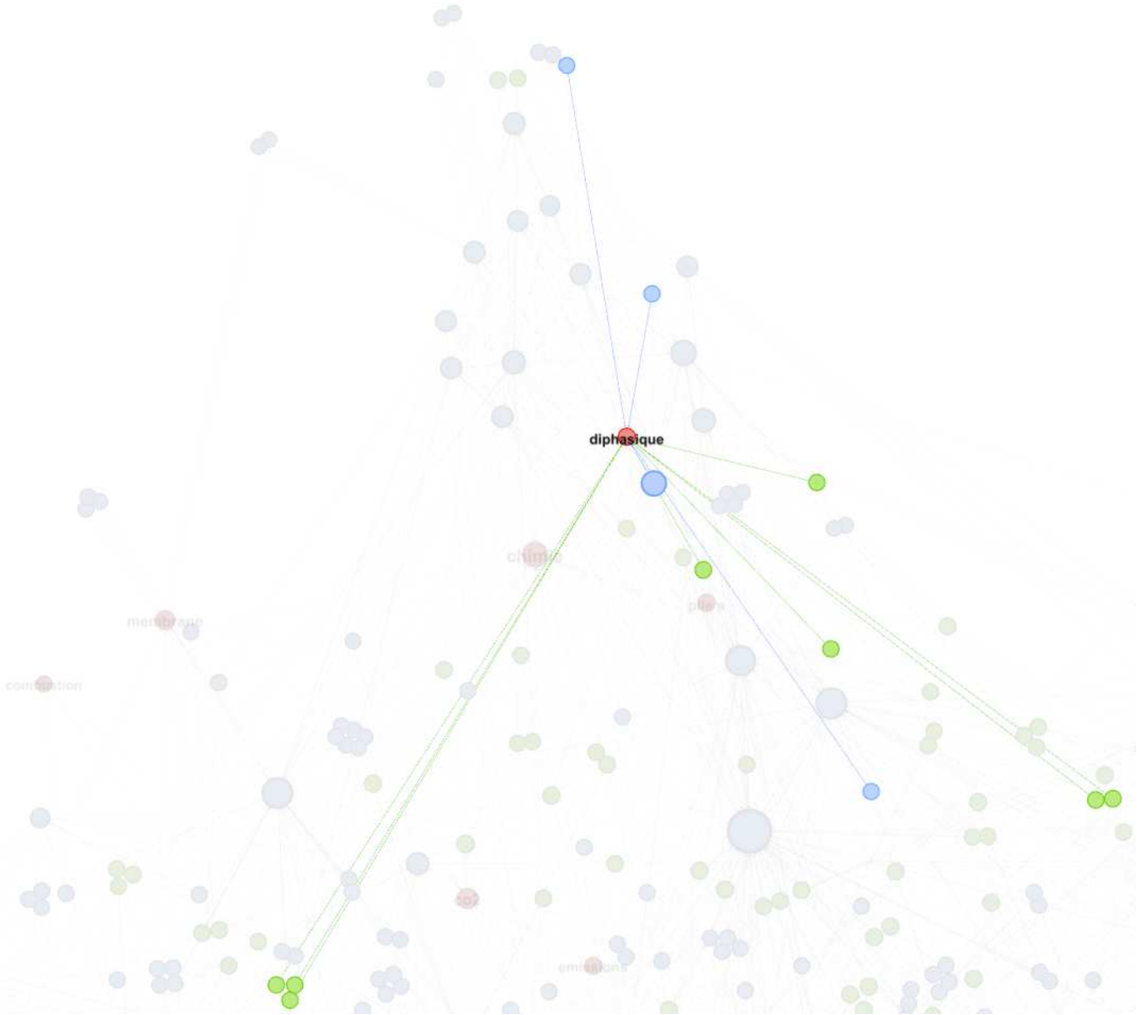


**Figure 97.** Visualization of the community labeled with the tag *diphasique* (diphasic).

## 6.4 Discussion

We could go further in exploiting semantic links between tags. (1) In [Limpens et al 2010] The ADEME's folksonomy was also enriched with `skos:related` and `skos:closeMatch` relations between tags, which exploitation should be investigated. For instance, the triple (*photovoltaic* `skos:related` *renewable energy)*, could be exploited to count one more occurrence of the tag *renewable energy* for each occurrence of the tag *photovoltaic*. (2) We can exploit other semantic relations between tags and use OWL entailments such as transitive properties. For instance, SKOS offers properties like `skos:transitiveNarrower` (notice that this transitive closure is indirectly performed by the iterative propagation of SemTagP); this could give better grouping of tags but perhaps produce too broad generalizations. Semantic statements like *energy* `skos:transitiveNarrower` *renewable energy* and *renewable energy* `skos:transitiveNarrower` *photovoltaic* could be exploited to count one occurrence of the tag *energy* for each occurrence of the tag *photovoltaic*. (3) The ontological primitives used to type the links between actors can describe different intensity of relationships. Consequently, when we choose to propagate tags through different properties, we could give more weight to tags propagated through given properties. For instance, in a working environment, tags used by `rel:worksWith` neighbors could be weighted twice more than tags used by `rel:colleagueOf` neighbours. (4) The tree formed by the `skos:narrower` relations of the dataset that we used for our experiment have a depth of 3, with most of the branches with a depth of 1 or 2. It could be interesting to test the propagation with a deeper tree of `skos:narrower` relations between tags. A deeper `skos:narrower` tree has intermediary tags between very broad tags and very narrow tags, which could help the propagation stop before labeling communities with too broad tags with less human control. (5) The algorithm may generate disconnected communities labeled with the same tag. This could be a way to detect structural holes [Burt 1992], namely communities that would benefits of exchanging but that cannot. (6) Finally, the current algorithm propagates only one tag per actor, an interesting extension would be to allow several tags to be propagated, which would also allow detect overlapping communities.

## 6.5 Partial Conclusion

SemTagP is a novel community detection algorithm that takes benefits of the semantics of RDF descriptions of social networks in order to reveal its communities and to meaningfully label their activities. To our knowledge, this is the first community detection that both detects and meaningfully labels communities. Based on a semantic propagation of tags, SemTagP turns large folksonomies into a subset of significant tags identifying and characterizing communities. The introduction of semantics in the RAK label propagation algorithm offered to handle not only the link structure of social graphs but also the semantic of the tags used by its actors. The label propagation mechanism was designed to exploits the social network link structure and trap labels in dense group of nodes. The assignation of tags, instead of random labels, improves the propagation with the shared vocabulary used to annotate the resources of the network. The exploitation of semantic relations between tags, inferred from the flat folksonomy, improves the propagation with the shared knowledge emerging from the social tagging process.

We tested this algorithm on the social network emerging from the Ph.D. theses funded by the ADEME agency, which enabled us to detect and characterize the distribution of its agents and activities. We compared the quality of the partition obtained with 4 different types of propagations: RAK, TagP, SemTagP and a controlled SemTagP. The controlled SemTagP outperformed the results of the 3 others algorithms, highlighting that the introduction of both the tags and the semantics between tags offers a significant improvement to the RAK algorithm.

# 7. Perspectives and Applications

In this thesis, we focused on the technological issues related to analyzing semantic representations of online social networks. However in order to leverage the social experience of a business intelligence that is conducted with emergent social software platforms, we need to go further in the analysis and to turn its results into functionalities. In this chapter, we discuss about some open issues that have to be tackled and that I would like to address in future researches.

In the first section, we present the challenge of considering temporal data in the social graph and during the analysis. Then, we present some perspectives to conduct analyses that scale to very large networks. Finally we review some human science literature that could help designing smart social functionalities that exploit the results of a semantic social network analysis

## 7.1 Temporal Social Network

In this thesis we have proposed a method to analyze social networks by considering their semantic representations. In particular, we focused our algorithms and experiments on the exploitation of the ontological primitives that were used to type and structure social networks. However, real social networks are very dynamic structures that can be considered as permanently evolving objects. An analysis that considers the temporal dimension of a social network could provide valuable information about its characteristics and give clues about its past and future evolutions. For instance, the date of interactions between actors could be used to refine community detection by focusing on recent activities. Furthermore, understanding how an actor became well positioned in a network can help anticipating and detecting the development of others important positions.

### 7.1.1 Temporal Semantic Network Model

Considering temporal information in network analysis implies to capture such information into the social network representation. [Kostakos 2009] defined a model of temporal graphs with "a graph representation that can retain rich temporal information". Each actor is represented by a set of nodes, one set for each social contact in time (e.g. send or receive an email at time $t_i$) with a weighted edge that links 2 consecutive contacts and encode the duration between them, and unweighted edges that link nodes of 2 actors engaged in a same contact. [Tang et al 2009] proposed a richer and simpler representation of temporal graphs based on "a sequence of time windows, where for each window we consider a snapshot of the network state at that time interval", which is formalized as follow:

**Definition 62. Temporal graph**: let a network trace starting at $t_{min}$ and ending at $t_{max}$, $R^s_{i,j}$ a contact between *i* and *j* at time *s*, and *w* a time range, a temporal graph $G^w(t_{min}, t_{max})$ is a sequence of graphs $Gt_{min}$, $Gt_{min+w}$,…, $Gt_{max}$, with Gt=(V, E) such that $(i, j) \in V$ if and only if there exists $R^s_{i,j}$ with $t \leq s \leq t + w$.

This last definition of temporal graphs can be easily adapted to RDF representations of social networks. [Carroll et al 2005] extended the syntax and semantics of RDF to cover the definition of named graphs, which enables us to identify and describe RDF graphs. This extension of RDF is being discussed for the next version of RDF. Consequently, RDF descriptions could be decomposed into named graphs described with temporal data, each one would contain RDF descriptions related to a given time window.

Some ontological models define temporal primitives that enable us to directly insert temporal data into the RDF descriptions. In particular, the Dublin core ontology provides the property `dcterms:created` to describe the date and time when a resource was created, which is reused by many others ontologies. For instance, SIOC (described in chapter 4) recommends using this property to describe the date of creation of online resources. However, even if online contributions contain valuable social networking data (e.g. emerging interactions and affiliations), many social data represent relationships between people and are typed with ontological primitives that do not contain any temporal data (e.g. `foaf:knows`, `rel:worksWith`, etc.).

If these previous solutions (named graphs and ontological primitives) enable us to integrate temporal data into RDF graphs, they are not necessarily adapted to fully obtain a temporal semantic graph. Temporal data can encode a large range of information. For instance, a temporal data can represent the date or the duration of an action as well as the date or the validity of an RDF description. Consequently, different perspectives and issues have to be considered to design a temporal semantic graph model.

### 7.1.2 Analysis of Temporal Semantic Social Network

Once integrated into the RDF graph, temporal data are valuable in the analysis. [Kostakos 2009] and [Tang et al 2009] have adapted classical social network analysis metrics to their own definition of temporal graphs. In particular they focused on the analysis of paths in such graphs, and have shown how the consideration of temporal data modifies the perception of distances between actors of a social network. For instance, with the model of [Kostakos 2009], if Peter has sent a mail to Jack at time *t*, and Jack has then sent a mail to Paul at time *t+1*, then the distance between Peter and Paul is of 1 in the chosen time unit, instead of a sequence of relations. Consequently, metrics that are based on the notion of path in graphs are directly impacted in temporal graphs. [Tang et al 2009] adapted different graph theory metrics to their model and highlighted how the granularity of the chosen time window can impact the results and the interpretations of an analysis.

In addition, the previous approaches only consider the temporal dimension of the social network, and they do not integrate any semantic information in their analysis like the classical network analysis method based on graph theory (see chapter 3). A temporal social network represented in RDF would enable us to not only consider the temporal dimension of a social network, but also the semantics used to typed and structure social links between actors by extending the method that we develop in this thesis.

How a temporal network analysis should be conducted on RDF descriptions of social network? How such analysis could take benefits of both the semantics and temporal data of RDF graphs?

## 7.2 Large Scale Network Analysis

In chapter 5, we conducted an experiment with a semantic social network composed of 60k nodes and millions of typed edges (representing both declared relationships and interactions). Simple metrics like the degree or the components were efficiently computed when considering the semantics of the network. However more complex queries involving path computation were so time consuming that we chose to limit the number of projections on the graph.  Considering the semantics of social graphs in the analysis is time and space consuming, which is one of the main drawbacks of our approach to scale to very large networks. More generally, computing complex metrics of network analysis is time consuming and experiments reported in chapter 3 show that handling very large networks need other approaches: (1) identifying computation techniques that are iterative, parallelizable or distributed, or (2) identifying approximations and heuristics that can be used with necessary conditions to obtain good quality results.

### 7.2.1 Iterative, Parallelizable or Distributed Algorithms

Incremental, parallelizable and distributed algorithms offer a more efficient use of computational resources as well as the possibility to dispose of more powerful ones.

First, incremental algorithms should compute network metrics once and then update them by considering graph modifications instead of computing new results from the complete new graph. Such approach would require iterative definitions of network analysis metrics as well as their implementations on semantic graphs.

Then, parallelizing the computation of network metrics would enable the execution of different tasks at the same time on many processing devices and put back together each output to get the final result. For instance, the clustering coefficient of a network can be computed as the sum of local values (see chapter 3), which could be computed in parallel and then summed.

Finally the ability to implement existing algorithms on distributed architecture would enable to exploit more powerful computational resources in order to analyze very large graphs. For instance, virtualization solutions[44] enable us to automate the distribution of algorithms on several processing devices.

### 7.2.2 Approximation and Heuristics

In chapter 5, when we limited the number of projections of queries we only obtained a sampling of the total amount of possible graph projections, from which we approximated the results of the computed metrics. However, in this case, the quality of the approximation was highly related to the index of the semantic engine, from which the graph projections are performed. Further investigations have to be conducted to evaluate and improve the quality of such approximations. In classical graph theory, different solutions have been investigated to approximate network analysis metrics. Some proposed sampling algorithms that enable to obtain good approximations of the computed metrics. Others defined heuristics, generally based on social network characteristics (e.g. small world property), to reduce initial problems to less complex ones.

In particular, different sampling algorithms have been investigated to compute the betweenness centrality [Brandes & Pitch 2007] [Bader et al 2007] [Geisberg et al 2008].

[44] http://en.wikipedia.org/wiki/Hardware_virtualization

The main issues tackled by these algorithms are to select a sampling of nodes that is representative of the whole graph, and to limit the bias of this selection on the final results (e.g. centralities of nodes that are adjacent of selected ones tend to be overestimated). The best results were obtained with a sampling performed with a random selection of nodes.

Recently, [Bonneau et al 2009] have conducted a social network analysis on a sample of the Facebook social network that was built from a crawl of the public profiles. The authors analyzed this sample and were able to accurately approximate the degrees and betweenness centralities of nodes, find shortest paths between users, and detect communities. This experiment shows the efficiency of analysing a sample in order to understand the organization of a network.

In our case, the limitation of the number of projections could be considered as a sampling, which quality could be improved by (1) adapting the querying of the RDF graph to analyze a random sampling, and/or (2) adapting the indexing of the semantic graph engine to propose sampling features in queries.

We started investigating a sampling method to compute the betweenness centrality on RDF graphs by adapting the algorithm of [Brandes & Pich 2007]. We process as follow:

1. select all the actor of the network involved in a given type of relation:

```
select ?actors where{
  {?x param[rel] ?y} UNION {?y param[rel] ?x}
}
```

2. select a random sample of actors among the results of this query.
3. Adapting the shortest path query of chapter 5, iteratively ask for the shortest paths starting and ending from each of the actors of the selected sample, and sum for each intermediary actor the corresponding partial betweenness.
4. Actors having the highest sums of partial betweenness have the highest betweenness centralities.

Our first investigations based on probabilistic methods confirm the relevancy of this approach, but a deeper analysis is required to assess its quality [Fedou 2009].

Step 1 and 2 enable us to extract the graph sample. However, the sampling of the actors could probably be an option of the semantic engine, and for instance we could apply directly the queries that ask for shortest paths with an option that enable a random sampling. Recently, SPARQL 1.1 introduces the `Sample` function that returns an arbitrary value from the set passed to it. This function could probably be used to define queries that perform a random sampling with a limited number of results.

## 7.3 Functionalities and Applications

*« It's better to organize information according to how people use it, rather than what department owns it » Jackob Nielsen*

In this thesis we designed the SemSNA ontology (described in chapter 5) that enables us to enrich social data with the graph characteristics of their corresponding social networks. Thus, we are able to structure the overwhelming flow of social data that are produced during a social business intelligence cycle. We now need to go beyond these technological advances, and develop functionalities and intelligent agents that leverage the experience of the consumers of these data.

Human scientists have investigated how people and organizations should manage their relationships and their locations in a social network. In particular they developed a network theory of social capital, which is defined by [Lin 2008] as follow:

**Definition 63. Social Capital:** "resources embedded in one's social network, resources that can be accessed and mobilized through ties in the network" [Lin 2008].

"The premise behind the notion of social capital is rather simple and straightforward: investment in social relations with expected returns" [Lin 1999]. Helping users of emergent social software to better access and enrich their social capital should be an objective of functionalities that are aimed to improve the experience of a social business intelligence. In particular, one should be able to understand the global structure of its social network, to mobilize the relevant resources that are accessible, and to develop relationships that enrich this capital.

### 7.3.1 Detecting and Highlighting Strategic Relationships

In an enterprise 2.0, [McAfee 2009] classifies the relationships of a person as follow:

- Strong links represent relationships that are frequently activated with a significant substance; typically people working in the same team are linked by strong links and interact regularly on working subjects.
- Weak links represent relationships without significant meaning; typically a simple acquaintance in a working place between people working separately.
- Potential links represent absent relationships but a social proximity; typically collaborators of collaborators are likely to become linked.
- Absent relationships are not likely to appear unless randomly, namely the remainder of the social network.

Human scientists have widely discussed the benefits of strong and weak links. On one hand dense strong links and closure of network may increase the sharing of resources and strengthen trust in a group [Coleman 1988]. On the other hand, weak links and sparse networks may facilitate access to more varied resources and sources of information [Granovetter 1973] [Burt 1992]. Generally, social networks are composed of dense groups separated by structural holes and bridged by few weak links: people in either side of a structural hole circulate in different flows of information [Burt 2001]. The development of new relationships could strengthen the density of a group or connect different groups; the resources that are available through potential links are considered as potential social capital [Lin 2008].

### 7.3.2 Managing Relationships and Strengthening Networking Positions

We have seen that the diverse relationships of a person provide different opportunities that have to be highlighted. In [Lin 2008], the author argues that the benefit of these opportunities depend on the purpose of actors' actions.

"For expressive action, the purpose is to maintain and preserve existing resources (e.g., to highlight production, to foster collaboration, or to strengthen community cohesion). The network strategy for expressive action is easily understood: to bind with others who share similar resources, who are sympathetic to one's needs to preserve resources, who are prepared to provide support or help" [Lin 2008]. Consequently, in order to reinforce the sharing of resources and receive support for propagating information it is important to maintain and develop a dense core of strong relationships.

"For instrumental action, the purpose is to obtain additional or new resources". Burt and Lin both argue that an actor cannot find new social capital in its inner circle and should access it through structural holes. [Lin 2008] argues it as follow: "where additional and better resources are needed, binding and bonding relations may not be sufficient. Accessing better social capital may require extending one's reaching beyond inner circles – bridging through weaker ties or non-redundant ties (e.g. structural holes)." [Burt 2004] argues that people that are close to structural holes have an informational benefit, consequently for instrumental action it is interesting to develop relationships with people that have a low clustering coefficient and a high betweenness centrality.

People do not necessarily know all the benefits provided by their network position and relationships, nor how to efficiently develop it. They should be assisted in it, with for instance ergonomic social information about their contacts, smart social notifications or the social context of search results.

### 7.3.3 Accessible, Mobilized and Potential Social Capital

Web based social tools have put at the disposal of their users a wild range of resources, shared by their own social network or socially distant people. People could interact with this content in different ways, search, consult, contribute, promote, etc. However, due to the huge amount of available resources, a wild range of interesting resources will not be mobilized, while less relevant ones will be. It is important to help users better select and collect these resources. In particular, users should know the resources that were published by their direct contact (i.e. their social capital), by socially close actors (i.e. potential social capital), and by socially distant actors. Moreover this content should be also augmented with data about the network position of related actors, which could provide insight on the redundancy or the novelty of its information.

In addition, it is also important to differentiate accessible capital and mobilized capital: "accessed social capital as well as actual use of social capital should be both measured and closely examined" [Lin 2008]. Differentiating access and mobilization is very pertinent in online platforms to help the users reuse yet mobilized resources and discover new ones. In social web applications, different elements highlight the mobilization of a resource such as a click on a link, references in posts, bookmarks, tagging actions, comments, etc. Consequently a user can be linked by many data to its mobilized social capital. Inversely, all the resources his contacts have made available to their network represent accessible social capital, which he is not even aware of the existence. Highlighting these resources is necessary to help users search and navigate in this consequent amount of unexploited resources.

# 8. Conclusion

This thesis proposes an approach to help analyzing the characteristics of the heterogeneous social networks that emerge from the use of web-based social applications. These researches are grounded in a social business intelligence scenario that integrates Web 2.0 tools for advanced collaborations, and Semantic Web technologies for data interoperability and information processing. In this scenario, the need for understanding emergent social organizations arises when the benefits of online collaboration is hindered by the frequent lost of relevant information in overwhelming flows of blinking social signals. Our contribution leverages Social Network Analysis with Semantic Web frameworks for analyzing and structuring semantically captured social data. We go beyond the mining of the flat link structure of social graphs by integrating a semantic processing of the network typing and the shared knowledge that emerges from online activities.

## 8.1 Contributions

The contributions of this dissertation can be exploited (1) to bring online social data to ontology-based representations, (2) to conduct a social network analysis that takes advantage of the rich semantics of such representations, and (3) to semantically detect and label communities of online social networks and social tagging activities.

### 8.1.1 Leveraging Online Social Data to Ontology-based Representations

In chapter 4, we presented how Semantic Web technologies, along with already existing ontologies, can be used to build, exchange, and query directed typed graph representations of online social data. Building on top of RDF (to perform triple descriptions of URI named resources), OWL and RDFS (to define ontologies and positively constrain the use of their primitives), different ontologies have been designed to semantically describe and link online social networks. FOAF is a base vocabulary for describing people, their attributes, their acquaintances and their online account. RELATIONSHIP and SIOC extend FOAF in order to precisely describe relationships between people and their online activities. SCOT and others ontologies propose vocabularies of social tagging, and tags as well as topics of web publications can be then semantically structured with the lightweight semantic primitives of SKOS.

While most online social data are still only accessible in XML, JSON or are trapped into relational databases, we have shown in chapter 5 how CORESE could be used to wrap, link and open these data with ontology-based metadata in RDF. CORESE SPARQL extensions offer to query heterogeneous data sources with SPARQL and build RDF on the fly. Once structured into RDF, we showed how a range of pre-processing can be applied to add types or relations whenever they detect a pattern, such as `construct` SPARQL queries, ontological constraint reasoning, or also proprietary rules engines.

### 8.1.2 Extending Social Network Analysis to Ontology-based Representations

In chapter 5, we extended social network analysis to ontology-based representations of online profiles, relationships, activities and emergent vocabularies. We extended social network analysis operators in order to parameterize them with the ontological primitives used to type the nodes and the links of their RDF representation. These semantically extended operators allow conducting rich analyses of social networks and handling the diversity of interactions and relationships with parameterized social metrics. We implemented these operators with SPARQL queries, which offer a native handling of subsumption relations in an engine like CORESE while classical SNA ignores the semantics of RDF graphs. This SPARQL operationalization of parameterized operators allows us to adjust the granularity of the analysis of relations. In addition, we are able to automatically focus on different sub-graphs while still working on the same initial graph. The SPARQL queries that we defined and tested implement the parameterized definitions of several classical SNA operators including the extraction of the degrees, the geodesics, the diameter, or the components. Once computed, the characteristics of the analyzed social networks can be used to enrich RDF descriptions of social data, using the SemSNA ontology that defines different SNA metrics. SemSNA includes primitives that describe (1) the context of the analysis, (2) strategic positions, and (3) the global network structure. This analytic based enrichment of social data enable us to more efficiently manage the life cycle of an analysis, and to offer advanced social functionalities that takes benefits of the emergent organization of the network.

This SNA method benefits of the evolutionary approach of Semantic Web technologies. Thus, new queries that compute new operators can be defined at anytime and SemSNA can be readily extended with appropriate semantics. However, several queries are based on SPARQL extension mechanisms of CORESE; some were already implemented in the engine and others were specifically designed or adapted for these researches (e.g.

the `group by any` clause used to query components was initially introduced for this specific need). The integration of many of these extensions into SPARQL1.1 (e.g. property paths, `group by` clause, aggregating functions) enables the reuse (or adaptation) of most of the presented queries with other semantic engines.

### 8.1.3 Semantic Community Detection and Labelling

Building on top of our results on semantic social network analysis, we proposed in chapter 6 an original approach that attempts to open a new perspective to community detection. We designed a community detection algorithm, SemTagP, that takes benefits of the semantic RDF descriptions of social networking and social tagging data, in order to not only detect communities but also to meaningfully label their activities. Extending the RAK algorithm [Raghavan et al 2007] that detects communities by random label propagation, SemTagP takes benefits of the linking structure of the network with a propagation mechanism and takes also advantage of the emerging semantics of social tagging with an ontology based labeling. First, the assignation of tags, instead of random labels, improves the propagation with the shared vocabulary used to annotate the resources of the network and offered a meaningful labeling of the communities. Then, the consideration of the inferred semantics between tags refines the labeling process and improves the propagation with the shared knowledge of the network. Finally, we introduced a human control option to refine the partitioning and avoid too broad generalizations between tags, which improved the results in the experiment that we conducted.

## 8.2 Publications

The researches that we conducted during my Master degree and the contributions proposed in this Ph.D. thesis led to 10 publications in workshops, conferences, books and a journal.

Our results on Semantic Wikis, to which I contributed during my Master degree, led to several publications in a French national conference [Buffa et al 2007], in the major journal on Semantic Web [Buffa et al 2008a], and in a book chapter [Buffa et al 2008b]

The first publications of my thesis were aimed at positioning our researches according to existing literature and other works conducted by different partners. In [Erétéo et al 2008] we proposed a state of the art on social network analysis and its application on a Social Semantic Web, and we positioned our contribution to this domain. Then, we published a position paper that highlights the articulation of different works conducted in our research group to analyze and capture collective intelligence from online interactions [Erétéo et al 2009a]. Finally, we published two papers that ground our researches and collaborations in the social business intelligence scenario of the ISICIL project [Gandon et al 2009] [Leitzelman et al 2009].

In the last publications, we published our method and different results we obtained to conduct a social network analysis with Semantic Web Frameworks in major venues. In

[Erétéo et al 2009b], we present our Semantic Web stack of tools to enhance social network analysis with semantics, we show how to parametrize and operationalize different centrality measures with SPARQL, and we describe the core primitives of SemSNA. In [Erétéo et al 2009c], we provide formal definitions in SPARQL of several SNA operators parameterized by ontological primitives, we extend SemSNA ontology, and we present the semantic social network analysis that we conducted on a real online social network. Finally we presented the complementary aspects of our results to semantically analyze social networks and those of [Limpens 2010] to semantically structure folksonomies, for Studying Virtual Communities [Erétéo et al 2011].

# 9. References

[Adamic & Adar 2003] Adamic, L. A., Adar E.: Friends ans Neighbors on the web. Social Networks, vol 25, p211-230. (2003)

[Adida 2008] B. Adida : hGRDDL: Bridging micorformats and RDFa. Special Issue of the Journal of Web Semantics on Semantic Web and Web 2.0, Volume 6, Edited by Mark Greaves and Peter Mika, Elsevier, p 61-69. (2008)

[Alkhateeb et al 2007] Alkhateeb, F., Baget, J.F., Euzenat, J.: RDF with Regular Expressions INRIA RR-6191, http://hal.inria.fr/inria-00144922/en. (2007)

[Angles & Gutierrez 2008] Angles, R., Gutierrez, C.: The Expressive Power of SPARQL. In Proceedings of the 7th International Semantic Web Conference, Karlsruhe, Germany. (2008)

[Anyanwu et al., 2007] Anyanwu, M., Maduko, A., Sheth, A.: SPARQL2L: Towards Support for Subgraph Extraction Queries in RDF Databases, Proc. WWW2007. (2007)

[Bader & Madduri 2006] D. A. Bader, K. Madduri: Parallel algorithms for evaluating centrality in real-world networks. ICPP2006. (2006)

[Bader et al 2007] Bader, D. A., Kintali, S., Madduri, K., Mihail, M.: Approximating betweenness centrality. WAW2007. (2007)

[Baget et al 2008]Baget, J-F., Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F., Giboin A., Gutierrez, A., Leclère, M., Mugnier, M.-L., Thomopoulos, R.: Griwes: Generic Model and Preliminary Specifications for a Graph-Based Knowledge Representation Toolkit. Proc. ICCS'2008. (2008)

[Barabasi & Albert 1999] Barabasi, A., Albert, R.: Emergence of scaling in random networks. Science, 286:509-512. (1999)

[Barber 2007] Barber, M.J.: Modularity and Community Detection in Bipartite Network. Phys. Rev. E, 76, 036106. (2007)

[Bavelas 1948] Bavelas, A.: A mathematical model for group structures. Journal of Human Organization 7, p: 16-30. (1948)

[Berge 1985] Berge, C.: Graphs and Hypergraphs. Elsevier Science Ltd. (1985)

[Berners-Lee et al 2001] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American. (2001)

[Bolshakova & Azuaje 2003] Bolshakova, N., Azuaje, F.: Cluster validation techniques for genome expression data. Signal processing, 83:825-833. (2003)

[Bonacich 1987] Bonacich, P.: Power and centrality: A family of measures. American Journal of Sociology, 92, 1170-1182. (1987)

[Bonneau et al 2009] J. Bonneau, J. Anderson, F. Stajano, R. Anderson: Eight Friends are Enough: Social Graph Approximation Via Public Listings. SocialNets 2009: The Second ACM Workshop on Social Network System. (2009)

[Borgatti 2005] Borgatti, S. P.: Centrality and network flow. Social networks 27 p: 55-71. (2005)

[Bothorel & Bouklit 2008] Bothorel, C., Bouklit, M.: An algorithm for detecting communities in folksonomy hypergraphs. 8th International Conference on Innovative Internet Community Systems I2CS 2008, Schoelcher, Martinique Sponsored by IEEE. (2008)

[Brandes 2001] Brandes, U.: A faster algorithm for betweenness centrality. J. Math. Socio 25(2): 163-177. (2001)

[Brandes & Pich 2007] Brandes, U., Pich, C.: Centrality estimation in large networks. Journal of Bifurcation and Chaos in Applied Sciences and Engineering 17(7): 2303-2318. (2007)

[Brandes 2008] Brandes, U.: On variants of shortest-path betweenness centrality and their generic computation. Social Networks 30 (2): 136-145. (2008)

[Breslin et al 2005] Breslin, J.G., Harth, A., Bojars, U., Decker, S.: Towards Semantically-Interlinked OnlineCommunities. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532,pp. 500–514. Springer, Heidelberg. (2005)

[Brickley & Miller 2004] Brickley, D., Miller, L.: FOAF Vocabulary Specification. FOAF Project http://xmlns.com/foaf/0.1/, last access: January 2011. (2004)

[Bron & Kerbosch 1973] Bron, C., Kerbosch, J.: Finding all cliques of an undirected graph. ACM 16, p: 575- 577. (1973)

[Buffa et al 2007] Buffa, M., Gandon, F., **Erétéo, G.** : Wikis et Web Sémantique, Conference IC 2007, p49-60. (2007)

[Buffa 2008] Buffa, M.: Du Web aux wikis : une histoire des outils collaboratifs. http://interstices.info, online journal, issue of 23/05/08. (2008)

[Buffa et al 2008a] Buffa, M., Gandon, F., **Erétéo, G.**, Sander, P., Faron, C. : SweetWiki: A semantic wiki, Special Issue of the Journal of Web Semantics on Semantic Web and Web 2.0, Volume 6, Issue 1, February 2008, Edited by Mark Greaves and Peter Mika, Elsevier, Pages 84-97. (2008)

[Buffa et al 2008b] M. Buffa, F. Gandon, **Erétéo, G.**: A Wiki on The Semantic Web, book chapter in Emerging Technologies for Semantic Work Environments: Techniques, Methods and Applications, Editors: Jorg Rech, Bjorn Decker, Eric Ras, Information Science Reference, ISBN: 9781599048772, p115-137. (2008)

[Buffa & Faron-Zucker 2010] Buffa, M. & Faron-Zucker, C. : Gestion sémantique des droits d'accès au contenu : l'ontologie AMO. In Proceeding of EGC2010. (2010)

[Burt 1992] Burt, R. S.: *Structural holes. The Social Structure of Competition*, Cambridge, Harvard University Press. (1992)

[Burt 2001] Burt, R. S.: Structural Holes versus Network Closure as Social Capital. N. Lin, K. Cook, R. S. Burt: Social Capital: Theory and research. Aldine de Gruyter: 31-56. (2001)

[Burt 2004] Burt, R. S.: Structural Holes and Good Ideas. American Journal of Sociology 100(2): 339-399. (2004)

[Carroll et al 2005] Carroll, J., Bizer, C., Hayes, P. Stickler, P.: Named graphs, provenance and trust. In the Proceedings of the 14th International Conference on World Wide Web, WWW2005. (2005)

[Cattuto et al 2008] Cattuto, C., Benz, D., Hotho, A., Stumme, G.: Semantic Grounding of Tag Relatedness in Social Bookmarking Systems. 7th International Semantic Web Conference. (2008)

[Chen et al 2009] Chen, J., Zaiane, O. R., Goebel, R. : Detecting Communities in Social Networks using Max-Min Modularity, SIAM International Conference on Data Mining (SDM'09), Sparks, Nevada, USA, April 30- May 2. (2009)

[Cohn & Marriott 1958] Cohn, B.S., Marriott, M.: Networks and centers of integration in Indian civilization. Journal of social Research 1, p:1-9. (1958)

[Coleman 1988] Coleman, J. S.: Social capital in the creation of human capital. The American journal of sociology, Vol 94, Supplement: Organizations and Institutions: Sociological and Economic Approaches to the Analysis of Social Structure. (1988)

[Conein 2004] Conein, B.: Communautés épistémiques et réseaux cognitifs: coopération et cognition distribuée. Revue D'Economie Politique 113, 141-159. (2004)

[Corby et al 2004] Corby, C., Dieng-Kuntz, R., Faron-Zucker, C.: querying the semantic web with the corese search engine. ECAI/PAIS2004. (2004)

[Corby 2008] Corby, O: Web, Graphs & Semantics, ICCS'2008. (2008)

[Corby et al 2009] Corby, O., Kefi-Khelif, L., Cherfi, H., Gandon, F., Khelif, K.: Querying the Semantic Web of Data using SPARQL, RDF and XML. INRIA Research Report n°6847. (2009)

[Corby & Faron-Zucker 2010] Corby, O., Faron-Zucker, C.: The KGRAM Abstract Machine for Knowledge Graph Querying. IEEE/WIC/ACM Int. Conference, September 2010, Toronto, Canada. (2010)

[Danon et al 2005] Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A. : Comparing community structure identification. Journal of statical Mechanics: Theory and Experiment, 2005(09):P09008. (2005)

[Djidjev 2008] Djidjev, H.N.: A scalable multilevel algorithm for graph clustering and community structure detection. In: Aiello, W., Broder, A., Janssen, J., Milios, E.E. (eds.) WAW 2006. LNCS, vol. 4936, pp. 117–128. Springer, Heidelberg. (2008)

[Donetti & Munoz 2004] Donetti, L., Munoz, M. A.: Detecting communities: a new systematic and efficient algorithm. Journal of statical mechanics, 2004(10):10012. (2004)

[Dongen 2000] Dongen, S. V.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht. (2000)

[Dunbar 1998] Dunbar, R., I., M.. The social brain hypothesis. Evolutionary Anthropology, 6, 178-90. (1998)

[Ereteo et al 2008] **Erétéo, G.**, Buffa, M., Gandon., F., Grohan, P., Leitzelman, M., Sander, P.: A State of the Art on Social Network Analysis and its Applications on a Semantic Web, SDoW2008, workshop at the 7th International Semantic Web Conference. (2008)

[Ereteo et al 2009a] **Erétéo, G.,** Buffa, M., Gandon., F., Leitzelman, M., Limpens, F.: Leveraging Social data with Semantics, W3C Workshop on the Future of Social Networking, Barcelona. (2009)

[Ereteo et al 2009b] **Erétéo, G.**, F. Gandon., O. Corby, M. Buffa: Semantic Social Network Analysis. Web Science. (2009)

[Ereteo et al 2009c] **Erétéo, G.**, Buffa, F. Gandon, O. Corby: Analysis of a Real Online Social Network Using Semantic Web Frameworks", 8th Int. Sem. Web Conf., ISWC'2009. (2009)

[Ereteo et al 2011] **Erétéo, G.**, Limpens, F., Buffa, M., Gandon, F., Corby, O., Leitzelman, M., Sander, P.: Handbook of Research on Methods and Techniques for Studying Virtual Communities, chapter Semantic Social Network Analysis, a Concrete Case. IGI Global. (2011)

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

[Everett & Borgatti 2005] Everett, M. G., Borgatti, S. P.: Ego network betweenness. Social Networks 1, 215-239. (2005)

[Facebook blog 2010] Facebook blog post http://blog.facebook.com/blog.php?post=409753352130, consulted in September 2010. (2010)

[Facebook press 2010] Facebook press http://www.facebook.com/press, consulted in September 2010. (2010)

[Fedou 2009] Fedou, M.: Modèle probabiliste pour une approximation d'indices de l'analyse des réseaux sociaux. intership report, Supélec. (2009)

[Finin et al 2005] Finin, T., Ding, L., Zou, L.: Social networking on the semantic web. Learning organization journal 5 (12): 418-435. (2005)

[Flom et al 2004] Flom, P. L., Friedman, S. R., Strauss, S., Neaigus, A.: A new measure of linkage between two sub-networks. Connections 26(1): 62-70. (2004)

[Freeman 1977] Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40, p:35-41. (1977)

[Freeman 1979] Freeman, L.C.: Centrality in social networks: Conceptual Clarification. Social Networks. 1, 215-239. (1979)

[Freeman & Borgatti 1991] Freeman, L. C., Borgatti, S. P.: Centrality in valued graphs: A mesure of betweenness based on network flow. Social Networks 13: 141-154. (1991)

[Fortunato et al 2004] Fortunato, S., Latora, V., Marchiori, M.: Method to find community structures based on information centrality. Phys. Rev. E 70(5): 056104. (2004)

[Gandon et al 2009] Gandon, F., Abdessalem, T., Buffa, M., Bugeaud, F., Comos, S., Corby, O., Delaforge, N., **Ereteo, G.**, Giboin, A., Grohan, P., Herledan, F., Le Meur, V., Leitzelman, M., Leloup, B., Limpens, F., Merle, A., Soulier, E. : ISICIL: Information Semantic Integration through Communities of Intelligence online, Proc. International Workshop on Web Intelligence and Virtual Enterprise (WIVE), 10th IFIP Working Conference on Virtual Enterprises ( PRO-VE'09), Thessaloniki, Greece. (2009)

[Garrison 1960] Garrison, W. L.: Connectivity of the interstate highway system. Proc. Reg. Science Association 6, p:121-137. (1960)

[Geisberg et al 2008] Geisberg, R., Sanders, P., Scultes, D.: Better approximation of betweenness centrality. ALENEX08. (2008)

[Girvan & Newman 2002] Girvan, M., Newman, M. E. J.: Community structure in social and biological networks. PNAS 99 (12): 7821-7826. (2002)

[Girvan & Newman 2004] Girvan, M., Newman, M. E. J.: Finding and evaluating community structure in networks. Phys. Rev. E, 69:026113. (2004)

[Golbeck et al 2003] Golbeck, J., Parsia, B., Hendler, J.: Trust network on the semantic web. Proceedings of cooperative information agents. (2003)

[Golbeck & Rothstein 2008] Goldbeck, J., Rothstein, M.: Linking social Networks on the web with FOAF. Proceedings of the twenty-third conference on artificial intelligence, AAA08. (2008)

[Gregory 2009] Gregory, S., Finding overlapping communities in networks by label propagation. http://arxiv.org/abs/0910.5516. (2009)

[Gruber 2005] Gruber, T.: Ontology of folksonomy: A mash-up of apples and oranges. In Conference on Metadata and Semantics Research MTSR2005. (2005)

[Gruber 2009] Gruber, T.: Ontology. In Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems. Springer, Heidelberg. (2009)

[Gustafsson et al 2006] Gustafsson, M., Hörnquist M., Lombardi, A.: Comparison and validation of community structures in complex networks. Phys 367: 559-576. (2006)

[Harary & Norman 1953] Harary, F., Norman, R.Z.: Graph Theory as a Mathematical Model m Social Science. Ann Arbor: University of Michigan, Institute for Social Research. (1953)

[Hawthorne 1851] Hawthorne, N.: The House of the Seven Gables. Wildside Press. (1851)

[Hendler & Golbeck 2008] Hendler, J., Goldbeck, J.: Metcalfe's law, web 2.0 and the Semantic Web. Journal of Web semantic 6(1): 14-20. (2008)

[Henri & Pudelko 2003] Henri, F., Pudelko, B.: Understanding and analyzing activity and learning in virtual communities. Journal of Computer Assisted Learning 19, 474-487. (2003)

[Holme et al 2002] Holme, P., Kim, B. J., Yoon, C. N., Han, S. K.: Attack vulnerability of complex networks, Phys. Rev. E 65, 056109. (2002)

[Hotho et al 2006] Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In The Semantic Web: Research and Applica- tions, volume 4011 of Lecture Notes in Computer Science, p. 411–426, Heidelberg: Springer. (2006)

[Jin et al 2007] Jin, Y., Matsuo, Y., Ishizuka, M. : Extracting a Social Network among Entities by Web mining. ESWC 2007. (2007)

[Kautz et al 1997] Kautz, H., Selman, B., Shah, M.: The hidden Web. AI magazine, Vol. 18, No. 2, pp. 27-35. (1997)

[Khare & Celik 2006] Khare, R., Celik, T.: Microformats: a pragmatic path to the Semantic Web. Proceedings of the 15th international conference on World Wide Web. (2006)

[Kim et al 2007] Kim, H., Yang, S., Song, S., Breslin, J. G., Kim, H.: Tag Mediated Society with SCOT Ontology. ISWC2007. (2007)

[Kleinfeld 2002] Kleinfeld, J. S., Could I be a big world after all? The "six degrees of separation" myth http://www.uaf.edu/northern/big_world.html. (2002)

[Knerr 2007] Knerr, T.: Tagging Ontology – Towards a Common Ontology for Folksonomies. http://tagont.googlecode.com/files/TagOntPaper.pdf. (2007)

[Knuth 1993] Knuth, D. E.: The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA. (1993)

[Kochut & Janik 2007] Kochut, K. J., Janik, M.: SPARQLeR: Extended SPARQL for Semantic Association Discovery, Proc. European Semantic Web Conference, ESWC'2007, Innsbruck, Austria. (2007)

[Kwak et al 2010] Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a Social Network or a News Media? In Proceedings of the 19th World Wide Web Conference, Raleigh, USA. (2010)

[Latora & Marchiori 2004] Latora, V., Marchiori, M.: A measure of centrality based on the network efficiency. Phy 9(6): 188. (2004)

[Leavitt 1951] Leavitt, H.J.: Some effects of communication patterns on group performances. Journal of Abnormal and Social Psychology 46, p:38-50. (1951)

[Leicht & Newman 2008] Leicht, E. A., Newman, M. E. J.: Community structure in directed networks, Phys. Rev.Lett. 100, 118703. (2008)

[Leitzelman et al 2009] Leitzelman, M., **Erétéo, G.**, Grohan, P., Herledan F., Buffa, M., Gandon., F.: Leveraging Social data with Semantics, W3C Workshop on the Future of Social Networking, Barcelona. (2009)

[Leskovec & Horvitz 2008] Leskovec, J. Horvitz, E., Planetary-scale views on a large instant-messaging network. In Proceeding of the 17th World Wide Web Conference, WWW2008, Beijing, China. (2008)

[Leskovec et al 2010] Leskovec, J., Lang, K. J., Mahoney, M. W.: Empirical Comparison of Algorithms for Network Community Detection. In proceeding of the 19th International World Wide Web Conference, WWW2010, Madrid, Spain. (2010)

|Limpens et al 2008] Limpens, F., Gandon, F., Buffa, M.: Bridging Ontologies and Folksonomies to Leverage Knowledge Sharing on the Social Web: a Brief Survey", Social Software Engineering & Applications (SoSEA), L'Aquila, Italy. (2008)

[Limpens et al 2010] Limpens, F., Gandon, F., Buffa, M., "Helping online communities to semantically enrich folksonomies", In proceedings of Web Science 2010, WebSci10. (2010)

[Limpens 2010] Limpens, F.: Multi-points of view semantic enrichment of folksonomies. Ph.D. thesis. (2010)

[Lin 2008] Lin, N.. A network theory of social capital. In D.Castiglione, J.W. van Deth, and G. Wolleb; editors, Handbook on Social Capital. Oxford University Press. (2008)

[Liu & Murata 2009] Liu, X., Murata, T.: Community Detection in Large-Scale Bipartite Networks, Proc. of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. (2009)

[Mariolis 1975] Mariolis, P.: Interlocking directorates and control of corporations: The theory of bank control, Social Science Quarterly 56, 425-439. (1975)

[Markines et al 2009]] Markines, B., Cattuto, C., Menczer, F., Benz D., Hotho A. & Stumme, G.: Evaluating similarity measures for emergent semantics of social tagging. In 18th International World Wide Web Conference, p. 641–641. (2009)

[Matsuo et al 2006] Matsuo, Y., Hamasaki, M., Takeda, H., Nishimura, T., Hasida K., Ishizuka, M. : POLYPHONET: An advanced social network extraction system. In proceedings WWW2006. (2006)

[McAfee 2009] McAffe, A., Enterprise 2.0: new collaborative tools for your organization's toughest challenges, Harvard Business Press. (2009)

[Mika 2005a] Mika, P.: Ontologies are us: A unified model of social networks and semantics., in The Semantic Web. Proceedings of the 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, volume 3729 of Lecture Notes in Computer Science, p. 522–536: Springer. (2005)

[Mika 2005b] Mika, P.: Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 3, No. 2-3., pp. 211-223. (2005)

[Milgram 1967] Milgram. S.: The Small World Problem. Psychology Today, 1(1): 61 – 67. (1967)

[Moody 2001] Moody, J.: Race, school integration, and friendship segregation in America, Am. J. Sociol. 107, 679-716. (2001)

[Moreno 1933] Moreno, J.L. Emotions mapped by new geography, New York Times. (1933)

[Newman 2001] Newman, M. E. J.: Scientific collaboration networks. Shortests paths weighted networks, and centrality. Phys Rev 64: 016132. (2001)

[Newman 2003a] Newman, M. E. J.: The structure and function of complex networks. SIAM Review 45, 167-256. (2003)

[Newman 2003b] Newman, M. E. J.: A measure of betweeness centrality based on random walks. Cond-mat/0309045. (2003)

[Newman 2004a] Newman, M. E. J.: Fast algorithm for detecting community in networks. Phys. Rev. E 69, 066133. (2004)

[Newman 2004b] Newman, M. E. J.: Detecting community structure in networks. European Physical Journal B 38: 321-330. (2004)

[Newman et al 2005] Newman, R., Ayers, D., Russell, S.: Tag ontology. Retrieved June 14, 2008, from http://www.holygoat.co.uk/owl/redwood/0.1/tags/. (2005)

[Newman 2006a] Newman, M. E. J.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104. (2006)

[Newman 2006b] Newman, M. E. J., Modularity and community structure in networks. Proceedings of the National Academy of Science USA, 103:8577–8582. (2006)

[Newman 2008] Leicht, E. A., Newman, M. E. J.: Community structure in directed networks, Phys. Rev. Lett. 100, 118703. (2008)

[Nicosia et al 2009] Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M.: Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech., P03024. (2009)

[Nieminem 1973] Nieminem, J.: On Centrality in a directed graph. Journal of Social Science Research 2, p: 371-378. (1973)

[Nieminem 1974] Nieminem, J.: On Centrality in a graph. Scandinavian Journal of Psychology 15:322-336. (1974)

[O'Reilly 2005] O'Reilly, T.: http://oreilly.com/web2/archive/what-is-web-20.html. (2005)

[Passant et al 2009] Passant, A., Laublet, P., Breslin, J.G., Decker, S.: SemSLATES: Improving Enterprise 2.0 Information Systems. The 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2009), Washington, DC, USA. (2009)

[Paolillo & Wright 2006] Paolillo, J. C., Wright, E.: Social Network Analysis on the Semantic Web: Techniques and Challenges for Visualizing FOAF, in Book Visualizing the semantic WebXml-based Internet And Information. (2006)

[Pissard 2008] Pissard, N.: Etude des interactions sociales mediates: methodologies, algrithmes, services". Ph.D. thesis. (2008)

[Passant et al 2008] Passant, A., Laublet, P.: Meaning Of A Tag: A Collaborative Approach to Bridge the Gap Between Tagging and Linked Data. LDOW2008. (2008)

[Pitts 1965] Pitts, F.R.: A graph theoric appraoch to historical geography. The Professional Geographer 17, p :15-20 .

[Pérez et al 2008] Pérez, J., Arenas, M., Gutierrez, C.: nSPARQL: A Navigational Language for SPARQL. In the procedings of ISWC'2008. (2008)

[Pons et al 2005] Pons, P., Latapy, M.: Computing communities in large networks using random walks. ISCIS2005. (2005)

[Radicchi et al 2004] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D. : Defining and identifying communities in networks. Proceedings of national Academy sciences USA 101, p: 2658-2663. (2004)

[Raghavan et al 2007] Raghavan, R.N., Albert, R., Kumara, S.: Near Linear Time Algorithm to Detect Community Structures in Large Scale Network. Phys. Rev. E, 76, 036106. (2007)

[Rattigan et al 2006] Rattigan, M. J., Maier, M., Jensen, D. : Using structure indices for efficient approximation of network properties. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 357-366. (2006)

[Rattigan et al 2007] Rattigan, M. J., Maier, M., Jensen, D.: Graph clustering with network structure indices. International Conference on Machine Learning. (2007)

[Rowe & Ciravegna 2010] Rowe, M., Ciravegna, F.: Disambiguating Identity Web References using Web 2.0 Data and Semanticss. The Journal of Web Semantics. (2010)

[San Martin & Gutierrez 2009] San Martin, M., C., Gutierrez: Representing, Querying and Transforming Social Networks with RDF / SPARQL. ESWC09. (2009)

[Santos et al 2006] Santos, E., Pan, L., Arendt, D., Pittkin M.: An Effective Anytime Anywhere Parallel Approach for Centrality Measurements in Social Network Analysis. IEEE2006. (2006)

[Scott 2000] Scott, J.: Social network analysis, a handbook. Deuxième edition, Edition Sage. (2000)

[Shaw 1954] Shaw, M.E.: Group structure and the behavior of individuals in small groups. Journal of Psychology 38, p139-149. (1954)

[Shimbel 1953] Shimbel, A.: Structural parameters of communication networks. Bulletin of Mathematical Bio-physics 15, p:501-507. (1953)

[Shirky 2008] Shirky, C.: Here Comes Everybody. Penguin Press. (2008)

[Smith 2008] Smith, M. S.: Social Capital in Online Communities, proceeding of the 2nd PhD workshop on Information and knowledge management. (2008)

[Story et al 2009] Story, H., Harbulot, B., Jacobi, I., Jones, M., FOAF+SSL: RESTful Authentication for the Social W. In Proc. of the First Workshop on Trust and Privacy on the Social and Semantic Web, Heraklion, Greece. (2009)

[Tifous 2007] Tifous, A., Ghali, A. E., Dieng-Kuntz, R., Giboin, A., Evangelou, C., Vidou, G.: An Ontology for Supporting a Community of Pratice. K-CAP'07. (2007)

[Tyler et al 2003] Tyler, J. R., Wilkinson, D. M., Huberman, B. A.: Email as spectroscopy: automated discovery of community structure within organizations. International Conference on Communities and Technologies p 81-96, Deventer, The Netherlands. (2003)

[UCINET 2002] Borgatti, S.P., Everett, M.G., Freeman, L.C.: Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies. (2002)

[Vidou et al 2006] Vidou, G., Dieng-Kuntz, R., El Ghali, A., Evangelou, C., Giboin, A., Tifous, A., Jacquemart, S.: Towards an Ontology for Knowledge Management in Communities of Practice. (2006)

[Wasserman & Faust 1994] Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press. (1994)

[Watts & Strogatz 1998] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks". Nature 393 (6684): 409–10. (1998)

[Weitzner 2007] Weitzner, D. J.: Whose Name Is It, Anyway? Decentralized Identity Systems on the Web. IEEE Internet Computing, vol. 11, no. 4, pp. 72-76, doi:10.1109/MIC.2007.95. (2007)

[Wellman 2001] Wellman, B.: Computer Networks As Social Networks. Science 293, 2031-34 (2001).

[Wenger 1998] Wenger, E.: Communities of Practice: Learning as a Social System Thinker (1998)

[White & Borgatti 1994] White, D. R., Borgatti, S. P.: Betweenness centrality measures for directed graphs. Social Networks 16, p 335 – 346. (1994)

[Wilkinson & Huberman 2003] Wilkinson, D. M., Huberman, B. A.: A method for finding communities of related genes. In proceedings of the national Academy of sciences. (2003)

[Wu 2004] Wu, F., Huberman, B.A.: Finding communities in linear time: a physics approach. HP Labs. (2004)

[Xu et al 2007] Xu, X., Yuruk, N., Feng Z., Schweiger, T. A. J.: SCAN: a Structural Clustering Algorithm for Networks. In KDD, pages 824.833. (2007)

[Zachary 1977] Zachary, W. W.: An Information Flow Model for Conflict and Fission in Small Groups, Journal of Anthropological Research, Vol. 33, No. 4, pp. 452-473. (1977)

[Zhou & Lipowsky 2004] Zhou, H., Lipowsky, R. : Network browniam motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities. International conference on computational science, p: 1062-1069. (2004)

# 10. Table of Figures

# 11. Table of Definitions