Construction, visualisation et analyse de réseaux modélisant des systèmes réels

Bruno Pinaud

LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE

LaBRI

université de BORDEAUX

HABILITATION À DIRIGER DES RECHERCHES

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

par **Bruno Pinaud**

Construction, visualisation et analyse de réseaux modélisant des systèmes réels

---

Date de soutenance : 14 octobre 2019

Jury d'examen :

David Auber	Pr, LaBRI, Univ. de Bordeaux	Examinateur
Rachid Echahed ...	CR HDR, LIG CNRS, Grenbole ...	Rapporteur
Pascale Kuntz	Pr, LS2N, Univ. de Nantes	Examinatrice
Christine Largeron	Pr, Université de St-Etienne	Rapportrice
Guy Melançon	Pr, LaBRI, Univ. de Bordeaux	Référent
Mohamed Mosbah .	Pr, LaBRI, Bordeaux INP	Examinateur
Alexandru C. Téléa	Pr, University of Utrecht, Pays-Bas	Rapporteur
Gilles Venturini ...	Pr, LIFAT, Univ. de Tours	Examinateur

L'informatique doit être au service de chaque citoyen. Son développement doit s'opérer dans le cadre de la coopération internationale. Elle ne doit porter atteinte ni à l'identité humaine, ni aux droits de l'homme, ni à la vie privée, ni aux libertés individuelles ou publiques.

Article 1 (version initiale) – Loi informatique et libertés – 6 janvier 1978

Résumé

Recruté maître de conférences en septembre 2008, je suis membre de l'équipe BKB (Bench to Knowledge and Beyond) du Laboratoire Bordelais de Recherche en Informatique (LaBRI) et responsable du thème « Model Analysis to Knowledge » [1]. Le nom du thème résume parfaitement la thématique générale de mes travaux qui se situent dans ce que l'on appelle dorénavant la « science des données ». Cette science, en plein essor, consiste à utiliser de façon combinée et complémentaire des outils mathématiques, statistiques, de l'informatique et de la visualisation d'informations pour permettre à un expert d'un domaine d'application d'analyser ses données complexes et hétérogènes mais pas forcément volumineuses (les données tiennent en mémoire centrale d'un ordinateur moderne). L'objectif principal est d'améliorer sa compréhension des données et de leurs relations et ainsi potentiellement extraire des nouvelles connaissances intéressantes. Dans ce contexte mes travaux se focalisent globalement sur la modélisation, construction, stockage, visualisation et analyse de divers types de réseaux construits à l'aide de données fournies, le plus souvent, par des experts de ces données. Dans ce manuscrit, je présente des collaborations avec des géographes, juristes, bioinformaticiens, historiens, et pour finir avec des experts en sciences forensiques ou criminelles.

Dans mon cas, un réseau est défini informellement comme un ensemble d'éléments liés entre eux ou en interactions. Mathématiquement parlant, un réseau est modélisé par un graphe qu'il faut construire, manipuler et visualiser. Depuis que je suis au LaBRI, la majeure partie de mes travaux est consacrée à une mise en œuvre particulière du processus décrit précédemment : le développement d'une méthodologie basée sur la programmation à base de règles, aussi appelée réécriture de graphes, et de son implémentation dans une plateforme visuelle et interactive baptisée Porgy [2]. L'objectif est de construire un modèle d'un système complexe, d'effectuer des simulations variées et d'analyser les résultats obtenus le tout en utilisant la visualisation et en interaction permanente avec l'expert des données. J'ai aussi notamment contribué à des travaux sur la visualisation de graphes ou l'évaluation quantitative et qualitative de méthodes de visualisation.

La construction et/ou la manipulation et/ou la visualisation d'un réseau sont toujours présentes dans mes travaux et mes publications portent sur tout ou partie de ce processus. Je retiens particulièrement 7 articles dans des journaux internationaux, 6 articles dans des conférences internationales, 3 chapitres de livres internationaux, 3 publications à portée nationale (dont un prix du meilleur article académique en 2015). Ces contributions ont été pour la plupart rendues possible par la coordination de deux projets ANR (jeunes chercheurs et international avec le Luxembourg) et deux co-directions de thèse (la première soutenue en décembre 2017 et l'autre débutée en avril 2016). Ces travaux possèdent, bien sûr, une part importante de développement logiciels. Les travaux sur Porgy sont liés à des collaborations nationales et internationales initiées rapidement après ma nomination MCF en 2008.

1. Au LaBRI, une équipe est constituée de plusieurs thèmes.
2. http://porgy.labri.fr

Dans ce manuscrit, après une introduction qui détaille le positionnement scientifique et le contexte de l'ensemble de mes travaux et leurs contributions, je présente dans deux chapitres la plateforme Porgy et sa méthodologie de programmation à base de règles. Le premier porte sur la conception et le développement de l'ensemble de la méthodologie et de l'outil Porgy pour en faire une plateforme générique, c'est-à-dire non dédiée à un type de donnée particulière. Le deuxième chapitre présente différentes applications de Porgy afin de montrer que la plateforme et sa méthodologie associée possèdent bien les propriétés annoncées comme la généricité, la souplesse (applications sur des données bioinformatique ou bien des réseaux sociaux) et couvre l'ensemble des étapes d'un processus de science des données : modélisation, simulation et analyse.

Plus récemment, je participe à des projets en collaboration avec des experts de sciences humaines et sociales (géographie, histoire contemporaine, réseaux criminels, humanités numériques). Mon objectif premier reste sensiblement le même que pour les travaux sur Porgy : à partir de données complexes et hétérogènes fournies par les experts, il s'agit de (re)construire et analyser visuellement et interactivement des réseaux pour aider l'expert dans sa compréhension des données. Je présente dans un dernier chapitre des leçons tirées de ces projets notamment la difficulté de trouver un modèle générique de réseau suffisamment souple pour s'adapter à tous les types d'application et d'interactions. En particulier, un système complexe réel ne se modélise pas seulement avec un seul réseau mais plutôt un ensemble de réseaux liés entre eux. Une réponse semble être l'utilisation d'un modèle récemment publié (2014) baptisé les réseaux multicouches. Ce modèle permet de vraiment prendre en compte l'ensemble de la complexité du système à modéliser tout en restant facilement compréhensible et manipulable par les experts des données. La notion de couches amène des questions nouvelles puisque celles-ci deviennent un artefact mobilisable par l'analyse et la visualisation. Ces techniques d'analyse et de visualisation doivent donc évoluer pour être adaptées aux réseaux multicouches. Ce manuscrit termine ainsi par différentes perspectives de recherche autour des réseaux multicouches et de la confiance, si difficile à obtenir, des experts dans la visualisation.

Table des matières

Liste des figures

Chapitre 1

Introduction

The ultimate subject of the visualization research community is ***people****, not* ***pictures****.*
Jeffrey Heer (University of Washington, USA), Conférence Eurovis 2019

Sommaire

Recruté maître de conférences en septembre 2008, je suis membre de l'équipe BKB (Bench to Knowledge and Beyond) du Laboratoire Bordelais de Recherche en Informatique (LaBRI) et responsable du thème « Model Analysis to Knowledge » [1]. Le nom du thème résume parfaitement la thématique générale de mes travaux qui se situent dans ce que l'on appelle dorénavant la « science des données » tout en conservant une volonté de rester le plus possible donnée agnostique. C'est à dire que je souhaite conserver une part la plus importante possible de généricité dans mes travaux et donc ne pas être spécialisé sur un type de donnée particulière. Cette science des données, en plein essor, consiste à utiliser de façon combinée et complémentaire des outils mathématiques, statistiques, de l'informatique et de la visualisation d'informations pour permettre à un expert d'un domaine d'analyser ses données abstraites, complexes et hétérogènes (mais pas forcément très volumineuse, c'est à dire que les données tiennent en mémoire centrale d'un ordinateur moderne) pour en améliorer sa compréhension et ainsi potentiellement extraire des nouvelles connaissances intéressantes (Cao, 2017). Dans ce cadre, je m'intéresse à tout ce qui touche de prêt ou de loin à la modélisation, visualisation, analyse, stockage de ces données sous la forme de réseaux. Je reprends la définition de Brandes et Erlebach (2015) qui définissent un réseau informellement comme un ensemble d'éléments liés entre eux ou en interactions. Mathématiquement parlant, un réseau est modélisé par un graphe. Dit autrement, un réseau est un graphe avec une sémantique associée à ces éléments.

Dans mes travaux, les réseaux modélisent des données et problèmes issus du monde réel et sont destinés à être manipulés par les experts de ces données et problèmes. Un réseau peut être manipulé de façon combinatoire et algébrique, spécialité historique du LaBRI. Néanmoins, les interactions avec les experts nécessitent une approche plus intuitive et interactive telle que la visualisation. Mes contributions combinent ainsi plus ou moins la modélisation, le stockage,

1. Au LaBRI, une équipe est constituée de plusieurs thèmes.

l'analyse, la visualisation et l'interaction avec de tels réseaux avec comme objectifs de produire et être capable d'évaluer que les visualisations sont efficaces et pertinentes pour l'utilisateur expert des données et répondent ainsi à ses problèmes. Je décrit ci-dessous (section 1.1) tout d'abord comment mes travaux se positionnent dans le domaine de recherche de la visualisation plus précisément celui de la visualisation analytique qui combine analyses automatiques et visuelles en lien fort avec des interactions de l'utilisateur dans le but d'acquérir des connaissances nouvelles sur les données (Keim et al., 2010). Puis je présente les travaux, sur lesquels j'ai passé le plus de temps depuis ma nomination MCF, à savoir la mise au point d'une méthodologie et son implémentation dans la plateforme PORGY pour modéliser, visualiser et analyser des systèmes complexes à l'aide d'une modélisation à base de règles, aussi appelée réécriture de graphes (cf. section 1.2). PORGY comme nombre de mes autres travaux utilisent la plateforme de visualisation d'informations et de visualisation analytique TULIP (figure 1.1) développée créée par David Auber (Auber et al., 2014, 2016, 2017). Mes travaux en visualisation suivent globalement la mantra de Shneiderman (1996) décrite ci-dessous et utilisent de nombreux algorithmes classiquement employés dans la communauté de la visualisation (Munzner, 2014) et pour la plupart implémentés dans TULIP. Néanmoins pour capturer du mieux possible toute la complexité d'un système réel et modéliser au mieux les processus des experts, les méthodes et outils évoluent (van den Elzen et van Wijk, 2014; Luciani et al., 2019). Je présente donc pour cela les travaux menés plus récemment avec un modèle de réseaux, baptisé réseaux multicouches (Kivelä et al., 2014) et un nouveau processus qui tend à se répandre qui consiste à partir de données détaillées pour retrouver le contexte global associé (cf. section 1.3).

1.1 La visualisation analytique

L'augmentation constante des volumes de données manipulées et manipulable par les ordinateurs rend l'utilisation de la visualisation incontournable ne serait-ce que pour permettre à l'expert d'obtenir rapidement des intuitions sur ses données. Une des meilleures visualisations existantes et une des plus populaires (Ward et al., 2010) (que j'utilise régulièrement dans mes enseignements et présentations) est la carte réalisée par Charles Joseph Minard en 1869 (figure 1.2). Cette carte synthétise la progression de l'armée de Napoléon lors de la campagne de Russie à l'hiver 1812–1813. Cette carte se lit dans le sens de lecture habituel de gauche à droite et démarre de Pologne pour terminer à Moscou. Le trait du haut est le voyage aller, le trait noir du bas est le retour. L'épaisseur des traits est proportionnelle au nombre de soldats restant dans l'armée. Ces explications succinctes suffisent à faire comprendre globalement le désastre de cette campagne Napoléonienne et tout l'intérêt de la visualisation, à savoir gagner en efficacité dans l'analyse des données (Bertin, 1967). De plus, cette carte est encore plus riche. Sans sacrifier la lisibilité, elle montre plusieurs autres données dont le nombre de soldats (orthogonalement aux traits représentant l'armée), une courbe de température (en bas) alignée avec la visualisation du nombre de soldats, quelques noms de lieux et de rivières notamment la deuxième rivière en bas de l'image en partant de la gauche qui est la tristement célèbre Bérézina (l'épaisseur du trait est divisée presque par 3 [2]). De nos jours, les ordinateurs nous permettent « simplement » de passer largement à l'échelle en terme de volume et complexité des données, de trier et nettoyer les données de façons plus ou moins automatiques et bien sûr de rendre les visualisations interactives.

La visualisation se définit ainsi : *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition* (Card et al., 1999). Plus récemment, Ben Shneiderman

2. Pour l'anecdote, c'est l'origine de l'expression française « (C'est) la bérézina » : c'est la catastrophe, l'échec total (Le Grand Robert de la langue française, 2019).

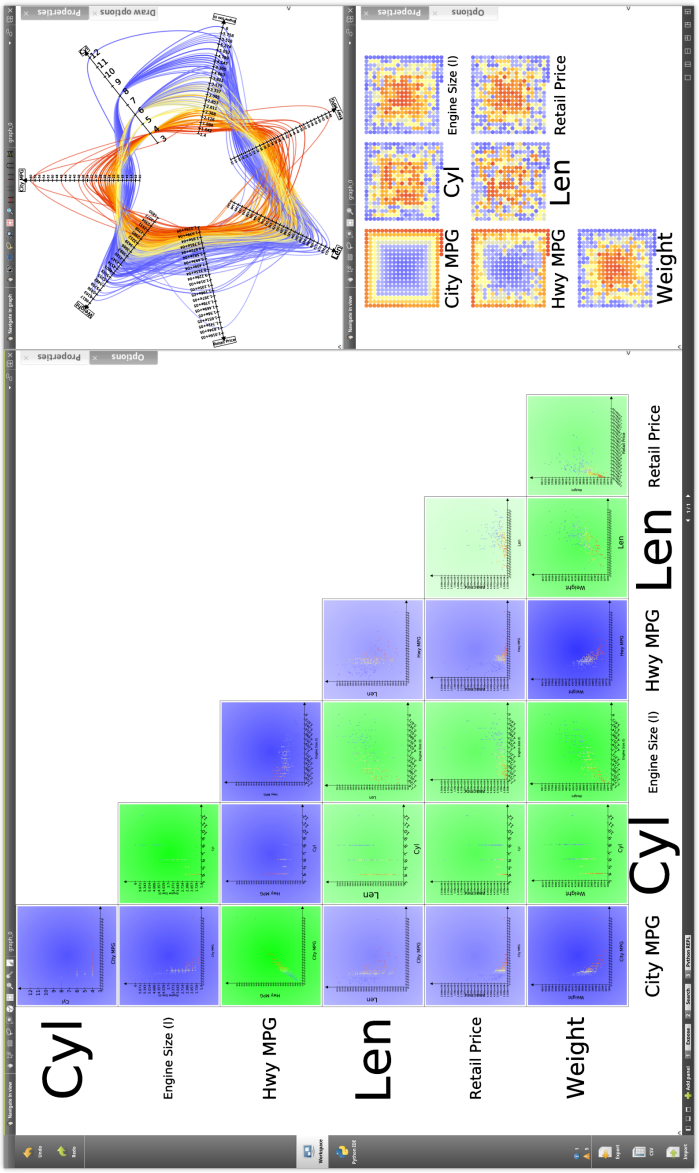


FIGURE 1.1 – Analyse d'un jeu de données multidimensionnelles sur des voitures (issu de Ward et al. 2010) avec TULIP. En haut à droite, une vue coordonnées parallèles où chaque axe est une dimension des données. Une voiture est ici représentée par une ligne. En dessous, une vue orientée pixels (chaque point est une voiture) est utilisée afin d'étudier très précisément les différences sur les corrélations entre deux variables. Dans cette vue les voitures sont ordonnées en suivant une spirale. Les voitures avec les valeurs les plus faibles pour chaque dimension sont au centre. La couleur des éléments est obtenue par une projection des valeurs de l'autonomie des véhicules en ville (City MPG). Dans la vue sur la gauche, une matrice de nuages de points (*scatterplot*) est utilisée pour rechercher visuellement les corrélations linéaires entre chaque couple de dimensions possibles. Les couleurs donnent une indication du cœfficient de corrélation linéaire (de -1 pour le bleu foncé à 1 pour le vert le plus foncé).

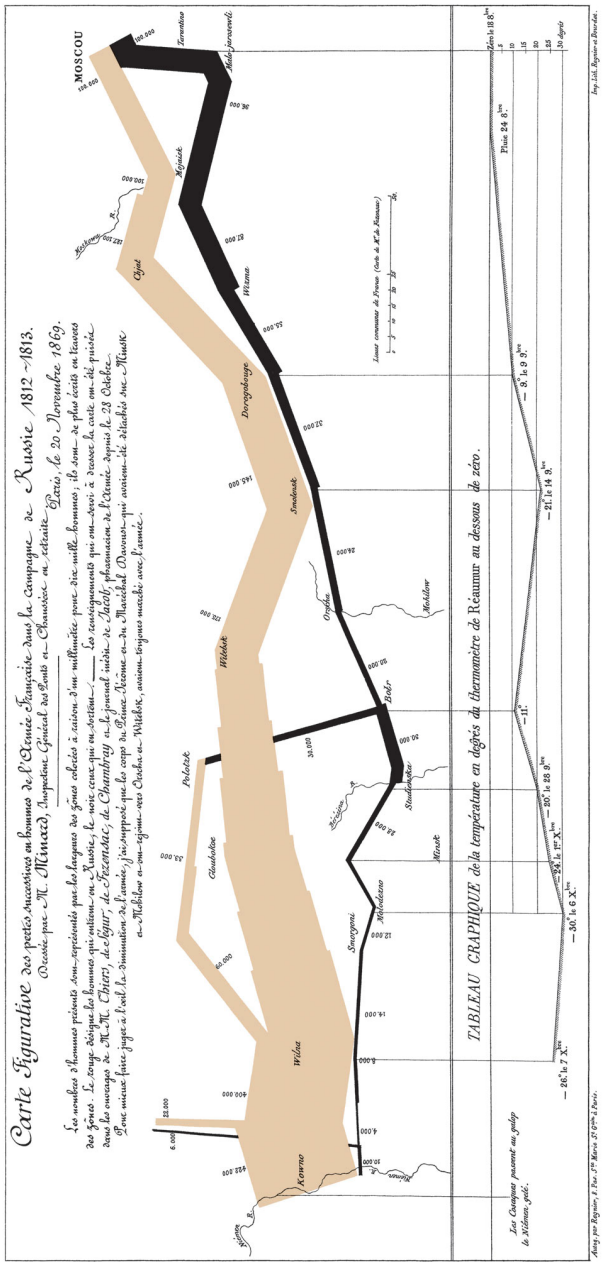


FIGURE 1.2 – Synthèse des pertes en hommes lors de la campagne de Russie de Napoléon (hiver 1812–1813). Carte réalisée par Charles Joseph Minard en 1869. Cette visualisation reste reconnue de nos jours comme une des meilleures visualisation d'informations existante (Ward et al., 2010).

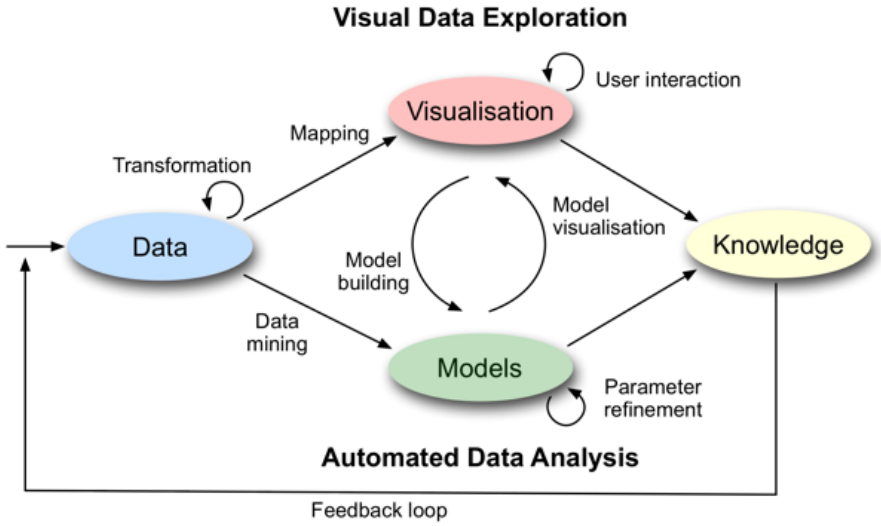


FIGURE 1.3 – Processus de visualisation analytique tel que décrit dans Keim et al. (2010).

(un précurseur dans le domaine) a déclaré (Hullman, 2019) : « ***The purpose of visualization is insight****, not pictures. By addressing meaningful problems and difficult decisions, we can help leaders and managers to be more effective.*» En effet, les visualisations sont avant tout destinées aux experts des données. Elles doivent donc rester accessibles, lisibles et compréhensibles. Un atout majeur d'une bonne visualisation fait que tout d'abord de nombreux problèmes informatiques et mathématiques mis en oeuvre pour produire et interagir avec les visualisations sont transparents pour l'expert et, de plus, certains problèmes du domaine de l'expert deviennent simples à résoudre grâce à une bonne visualisation. Néanmoins, il faut être capable d'évaluer quantitativement et qualitativement ces affirmations. Les travaux présentés dans ce manuscrit illustrent ces différents points.

Le processus suivi dans mes travaux est celui de la **visualisation analytique** initialement décrit dans Keim et al. (2008), repris dans Keim et al. (2010), et illustré par la figure 1.3. Je reprends la mantra de Keim qui explique parfaitement le processus : « *Analyse first – Show the Important – Zoom – Filter and analyse further – Details on Demand* ». Cette mantra est une évolution du traditionnel processus d'analyse visuelle de données énoncé par Shneiderman (1996) : « *Overview first, zoom and filter, then details on demand* ». Depuis 1996, les volumes et la complexité des données ont tellement évolué que des traitements automatiques sur les données préalables à la visualisation sont devenus indispensables. Dans mes travaux, les données, jugées pertinentes des experts du domaine d'application sont modélisées par des réseaux eux mêmes visualisés par des graphes. Il s'agit ensuite d'utiliser ou mettre au point les méthodes et outils visuels et interactifs pour que les experts puissent obtenir *in-fine* une meilleure connaissance et compréhension des données modélisées et des interactions entre elles. Mes contributions couvrent tout ou partie de ce processus illustré par les boîtes ovales de la figure 1.3 :

— Pour la partie « *Data* », leur transformation, nettoyage et analyse, je m'intéresse à tout type de données. Mes contributions utilisent notamment des données bioinformatiques (Andrei et al., 2019, 2011), des données géographiques (Noucher et al., 2016; Georis-Creuseveau et al., 2018), des réseaux sociaux de diverses origines (Vallet et al., 2015; Fernandez et al., 2018) dont des réseaux criminels (Lavaud-Legendre et al., 2017) ou encore des données économiques (Chinelo Ene et al., 2017; Ene et al., 2018). Ces données nécessitent le plus

souvent d'être numérisées, nettoyées, mises en forme, triées, agrégées avant de pouvoir être utilisées.

— Pour la partie « *Models* », j'ai beaucoup travaillé à la mise au point d'une méthodologie visuelle et interactive basée sur la réécriture de graphes et implémentée dans le logiciel Porgy (Pinaud et al., 2012; Fernández et al., 2019). Je présente ces travaux dans la section 1.2 suivante (et en détail dans les chapitres 2 et 3).

— Pour la partie « *Visualisation* », l'équipe que j'ai rejoint en 2008 avait déjà pour habitude d'implémenter l'ensemble de ses travaux dans la plateforme Tulip (figure 1.1) dont le point fort est son support de l'ensemble du processus de visualisation analytique. Je perpétue cette tradition en étant un contributeur et un ambassadeur actif de la plateforme, en particulier avec la plateforme interactive Porgy (voir ci-dessous) qui s'appuie sur Tulip. Dès les premiers travaux, j'ai piloté les premiers développements de Porgy en étant bien aidé par un ingénieur d'études (financement projet ANR jeunes chercheurs) et plus tard secondé par Jason Vallet pour ses travaux de thèse. L'imagination et l'inventivité des concepteurs de visualisations sont encore et toujours indispensables pour produire de nouvelles visualisations qui soient à la fois esthétiquement agréables à regarder tout en restant efficaces pour aider les experts des données à manipuler et analyser facilement leurs données. Ainsi, je m'intéresse à évaluer et comparer des techniques de visualisation pour différents types de données afin de s'assurer que leurs objectifs vers l'utilisateur final sont atteints (Archambault et al., 2011, 2010a,b; Sansen et al., 2015).

— Enfin, la partie « *Knowledge* » concerne les connaissances que l'expert peut acquérir soit en interagissant directement avec la visualisation et/ou en affinant la construction (surtout si cela est fait par des algorithmes) de son modèle de données grâce à la visualisation. L'ensemble de mes travaux illustre cette acquisition de nouvelles connaissances par l'expert. Il y a aussi en plus toujours une part de validation ou mieux d'évaluation des outils proposés aux experts des données ainsi que des leçons apprises.

1.2 Visualisation et modélisation à base de règles

Rapidement après mon arrivée au LaBRI en septembre 2008 est née une collaboration internationale (Prof. Maribel Fernandez, King's College London) grâce au programme des équipes associées Inria[3] sous l'impulsion de Hélène Kirchner (Inria) qui était venue nous trouver et collaborer avec nous pour l'occasion. L'idée de départ est relativement simple : en réécriture de termes et de graphes, qui est le domaine de recherche de Maribel et Hélène, les éléments qui participent à une réécriture sont représentés, la plupart du temps, dans les publications avec des graphes. Néanmoins, il n'existait, à notre connaissance et à ce moment là, pas de plateforme interactive utilisant la visualisation pour chaque étape du processus de réécriture, c'est à dire la modélisation, la simulation et l'analyse d'un système. Nous nous sommes donc lancés le défi d'en produire une. Rapidement, nos travaux ont consisté à mettre au point une méthodologie complète ainsi qu'une plateforme visuelle et interactive pour modéliser, simuler, et analyser le comportement d'un système à partir d'un ensemble de modifications locales effectuées par l'application de règles de réécriture. Cette technique de modélisation s'avère être utilisée même en dehors de sa communauté de recherche d'origine sachant qu'elle ne dit parfois pas son nom comme dans Eberle et Holder (2007) au sujet de la détection de fraude en utilisant des techniques basées sur des graphes ou dans Kejžar et al. (2008) qui présente plusieurs méthodes pour la construction de réseaux sociaux (cf. section 3.2 page 36 pour plus de détails sur ce deuxième exemple). Plus

3. J'étais membre à cette époque de l'équipe GRAVITE arrêtée en 2012.

récemment, la communauté de la visualisation s'intéresse aussi à la réécriture (Smith et al., 2012; Boutillier et al., 2018; Wenskovitch et al., 2014).

La réécriture de graphes, appelée aussi modélisation à base de règles, est un domaine de recherche avec de solides bases mathématiques (Courcelle, 1990; Löwe, 1993) mais peut néanmoins être expliquée relativement intuitivement. Il faut s'imaginer un jeu dans lequel des règles de la forme $A \to B$ sont appliquées sur un graphe G. Il s'agit de rechercher une image du graphe A dans un sous-graphe de G et de la remplacer par une image de B en reconnectant convenablement les éventuelles arêtes dont une extrémité est incidente à un sommet de A (pas d'arête pendante). Plus de détails et des représentations visuelles de ces différents éléments sont à retrouver dans la section 2.2 page 14.

La réécriture est un domaine de recherche que j'ai découvert avec cette collaboration. Notre objectif n'a jamais été de produire des résultats fondamentaux nouveaux dans ce domaine mais d'essayer de combiner réécriture et visualisation dans une approche générique et intégrée. Un challenge, que nous avons relevé avec Porgy, est de pouvoir visualiser et analyser l'évolution du système modélisé à n'importe quelle échelle et donc de conserver un historique de l'ensemble des simulations effectuées. Le chapitre suivant (chapitre 2) est une synthèse des travaux liés au développement de la plateforme Porgy et à la méthodologie associée. La figure 2.1 est une vue d'ensemble de Porgy utilisé pour la modélisation d'un réseau d'interactions de protéines. Cet exemple, utilisé depuis les prémisses des travaux sur Porgy (Andrei et al., 2011) et d'autres sont développés dans le chapitre 3 page 27. Ces exemples permettent de montrer les différentes contributions de la plateforme :

— la définition, l'implémentation et l'utilisation d'un modèle de graphes à ports dans lequel les sommets possèdent des ports sur lesquels sont connectés les arêtes. Ce modèle permet notamment de prendre en charge les réseaux biologiques qui nécessitent de pouvoir définir des attributs sur les connexions des arêtes sur les sommets ;

— la méthodologie de réécriture basée sur des graphes à ports et des règles de réécriture ;

— la définition et l'implémentation d'un langage pour piloter précisément l'application et le choix des règles ;

— L'utilisation de la visualisation à chaque étape du processus : définition des règles, du graphe initial, gestion de l'historique des réécritures, des simulations et de leur analyse. En effet, l'utilisation de la visualisation permet de rendre relativement facile des problèmes formellement complexes tels que le suivi des éléments pendant le processus de réécriture, la mise en avant de propriétés fondamentales comme la confluence ou encore le suivi des évolutions des paramètres du système.

1.3 Évolution des pratiques en visualisation et réseaux multicouches

Comme le font remarquer Luciani et al. (2019), les recherches en visualisations sont presque exclusivement basées sur la mantra de Shneiderman (1996) : *Overview First, Zoom and Filter, then Details on Demand*. Cette approche, échelle globale vers le détail, fonctionne bien sur des jeux de données de taille modérée quand la représentation globale des données est lisible donc compréhensible et manipulable dans des visualisations interactives. L'expert des données doit aussi avoir une bonne connaissance de ses données et de ses objectifs. Néanmoins, il s'avère que l'approche inverse, soit du détail vers le global, jusqu'à présent peu exploitée dans la communauté de visualisation (van Ham et Perer, 2009; van den Elzen et van Wijk, 2014) fonctionne bien mieux et est plus naturelle pour les experts notamment en Sciences Humaines et Sociales qui sont le

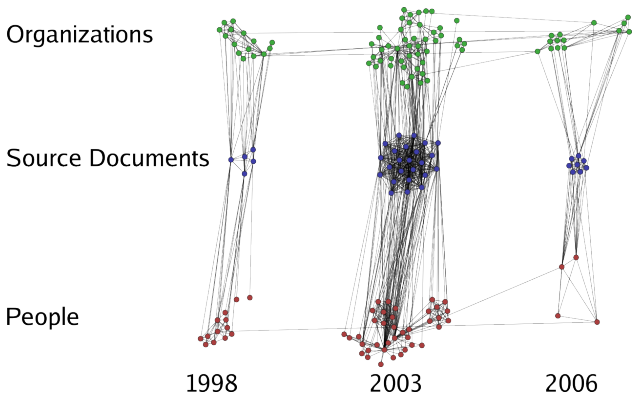


FIGURE 1.4 – Un exemple illustratif de réseau multicouche appliqué aux humanités numériques sur des données issues du projet BLIZAAR. Application sur le patrimoine culturel numérique. Plus formellement, c'est un réseau temporel de co-occurences entre des documents liés à l'histoire de la construction de l'Union Européenne, des personnes et organisations diverses citées dans les documents.

domaine d'application privilégié de mes autres travaux présentés dans ce manuscrit (chapitre 4 page 51). En effet, les experts des données ont souvent une connaissance incomplète de leurs données car trop volumineuses, trop complexes ou trop récentes ou alors une représentation globale des données n'est simplement pas envisageable pour les mêmes raisons. Il faut alors être capable d'explorer les données sans but initial parfaitement défini notamment pour valider le potentiel des données à répondre aux questions formulées par les experts. Généralement, les experts ont des questions ciblées et précises sur leur domaine que la prise en main des données (échelle du détail) permet, voire exige, de mettre en contexte (échelle globale). Le titre de Luciani et al. (2019) ouvre de nombreuses opportunités à développer : *Details-First, Show Context, Overview Last*. Je présente dans le chapitre 4 quelques travaux dans ce contexte.

De plus, l'utilisation d'un unique réseau ne suffit pas à modéliser convenablement toute la complexité d'un système réel. Pour bien prendre en compte cette complexité dans son ensemble, je présente différents travaux qui montrent que l'utilisation des réseaux multicouches (Kivelä et al., 2014) semblent être une bonne solution fédératrice et facile à appréhender entre les différents acteurs du processus de visualisation et les experts des données. Un exemple simple, issu du projet ANR BLIZAAR, est présenté figure 1.4. Je présente ce projet, que j'ai coordonné, et mes travaux en lien avec ces réseaux dans le chapitre 4 et des perspectives de recherche prometteuses sur ces réseaux dans le chapitre 5.

1.4 Synthèse et structure du manuscrit

Depuis ma nomination en tant que MCF, mes contributions scientifiques se synthétisent ainsi :

1. les travaux qui ont abouti à la plateforme de visualisation interactive PORGY pour la modélisation, simulation et analyse de systèmes basés sur la programmation à base de règles et en particulier :
   (a) la définition et justification de la méthodologie formelle basée sur des graphes à ports et un pilotage fin de l'application des règles ;
   (b) l'implémentation de cette méthodologie dans un système visuel et interactif, baptisé PORGY ;

   (c) l'utilisation de PORGY sur divers systèmes pour montrer la souplesse et la généricité de l'approche ;

2. des collaborations avec des experts de différents domaines (principalement en Sciences Humaines et Sociales, SHS) pour modéliser et visualiser leurs données issues du monde réel et répondre à leurs questions. Ces collaborations ont toutes un point commun qui est la nécessité de redéfinir à l'inverse les techniques classiques de visualisation : commencer par des détails (que l'expert maîtrise) pour produire des représentations plus globales et abstraites des données pour étudier le contexte autour des détails. Les réseaux multicouches semblent être un bon outil pour cela tant pour la modélisation que pour la visualisation ;
3. Des évaluations qualitatives et quantitatives de techniques de visualisations pour mesurer leur efficacité et leur pertinence pour des tâches données.

Dans ce manuscrit je présente mes travaux liés aux deux premiers points. Les chapitres 2 et 3 concernent les travaux liés à PORGY. Le chapitre 2 traite de la conception de la méthodologie de réécriture et la modélisation informatique de l'outil PORGY. Le chapitre 3 présente les principaux systèmes mis au point avec PORGY afin de montrer l'intérêt de notre méthodologie, sa souplesse et sa généricité. Le chapitre 4 présente les travaux effectués lors de différentes collaborations (appliquées à des données géographiques, des réseaux criminels ou encore sur le patrimoine culturel numérique) qui nous ont permis de mettre en avant l'intérêt d'une approche « détail vers global » et des réseaux multicouches. Je termine ce manuscrit (chapitre 5) par mes perspectives de recherche sur les réseaux multicouches, et comment améliorer la confiance des experts dans les visualisations réalisées avec leurs données et les éventuelles conclusions tirées des visualisations.

Chapitre 2

PORGY : plateforme visuelle et interactive pour la réécriture de graphes

Sommaire

Ce chapitre est dédié aux travaux que j'ai mené et encadré sur la définition et le développement de la plateforme visuelle et interactive de réécriture de graphes baptisée PORGY (figure 2.1). Son histoire a commencé à peine quelques semaines après mon recrutement au LaBRI (septembre 2008) et mon intégration à l'équipe projet mixte Inria/LaBRI nommée GRAVITE [1]. L'équipe avait été contactée par Hélène Kirchner (Inria) et Maribel Fernández (King's College London) pour la mise en place d'une collaboration autour de la modélisation, simulation et analyse de systèmes complexes à l'aide de la réécriture de graphes (aussi connu sous le nom de modélisation à base de règles), thématique dont elles sont expertes et qui de plus a une histoire au LaBRI (Courcelle, 1990).

Un **système de réécriture de graphes** se résume facilement par un jeu où des transformations régies par des règles sont appliquées sur un graphe initial jusqu'à ce que certaines conditions

1. Voir https://www.inria.fr/equipes/gravite. L'équipe a été arrêtée au 31/12/2012 à la demande de ses membres après une évaluation très positive.

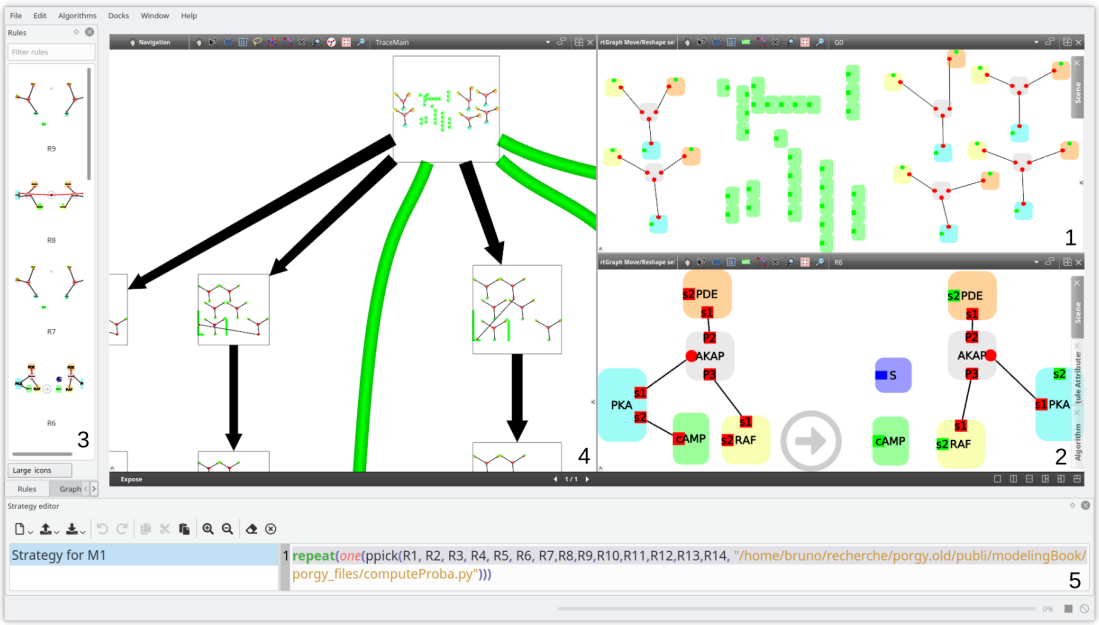
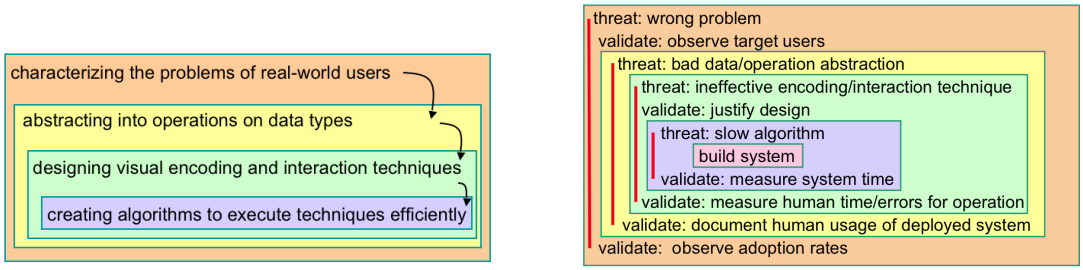


FIGURE 2.1 – Vue d'ensemble de PORGY sur un réseau biologique. Les différentes parties de la figure sont le **graphe à réécrire** qui représente un état du système (**1**), une **règle de réécriture** (**2**) et le catalogue de l'ensemble des règles disponibles (**3**), l'historique des opérations menées sur le système dans l'**arbre de dérivations** (**4**) et enfin l'éditeur de **stratégie** pour piloter les réécritures (**5**).

soient vérifiées. Une règle décrit un motif à identifier et à instancier dans un graphe et comment transformer ou faire évoluer ce motif. Dans la littérature de ce domaine, les auteurs utilisent souvent des formalismes graphiques (voir, par exemple, les réseaux d'interactions, Lafont, 1990). Ils sont plus faciles à comprendre, à expliquer et ils permettent de transmettre des intuitions sur le système bien plus facilement qu'un modèle algébrique. Néanmoins, en 2008, il n'existait, à notre connaissance, aucun environnement intégré interactif et visuel pour la modélisation, simulation et analyse d'un système de réécriture de graphes.

Je me suis alors fortement engagé dans cette collaboration qui s'est rapidement matérialisée par l'obtention d'un financement d'équipe associée Inria sur la période 01/2009–12/2011 (et dont je suis rapidement devenu le responsable scientifique). Cette période a permis d'établir les fondamentaux de PORGY, qui font l'objet de ce chapitre, à savoir le développement d'une plateforme basée sur la visualisation pour gérer l'ensemble des aspects relatifs à la modélisation, la simulation et l'analyse de systèmes complexes en utilisant le paradigme de la réécriture de graphes. Suite à l'équipe associée, les travaux sur PORGY ont continué grâce au projet ANR Jeunes Chercheurs EVIDEN (**E**xploration et **VI**sualisation de **D**onnées r**E**lationnelles dy**N**amiques, 12/2010–11/2014 [2]) dont j'étais le coordinateur [3]. Ce projet m'a notamment permis de recruter et d'encadrer plusieurs ingénieurs d'études ou stagiaires (de la première année de licence au Master 1) pour un temps de travail cumulé sur PORGY de plus de deux ans. Un de ces ingénieurs (Jason Vallet) a ensuite poursuivi en thèse (soutenue en décembre 2017, bourse ministérielle, co-

2. http://www.agence-nationale-recherche.fr/Projet-ANR-10-JCJC-0201
3. Ce projet a été co-écrit avec Romain Bourqui qui était investi au même niveau que moi.



(a) Le modèle imbriqué.

(b) Chaque niveau du modèle est associé à des mécanismes de validation et des risques identifiés.

FIGURE 2.2 – Le modèle imbriqué de Munzner tel que présenté dans Munzner (2009).

direction avec Guy Melançon) en développant encore plus la plateforme PORGY avec de nouvelles fonctionnalités et applications autour des réseaux sociaux. Cette thèse a reçu le prix de thèse de la société savante Extraction et Gestion des Connaissances (EGC) lors de l'édition 2019 de la conférence du même nom. Un article issu des travaux de cette thèse a reçu le prix du meilleur article académique lors de l'édition 2015 de cette même conférence (Vallet et al., 2015) [4]. Les travaux menés dans cette thèse sont décrits plus en détails dans le chapitre 3.

Comme expliqué dans Pinaud et al. (2012), le développement de PORGY a suivi le modèle imbriqué pour le développement de plateformes de visualisation publié par T. Munzner (Munzner, 2009) et reproduit figure 2.2. Cette méthodologie en 4 étapes se résume ainsi pour PORGY :

— tout commence par la définition d'un ensemble de questions et problèmes à résoudre exprimés par les experts des données aidés par les modélisateurs ;

— ensuite, ces problèmes sont traduits en exigences sur le système de réécriture, les tâches à effectuer ou les visualisations avec lesquelles interagir ;

— enfin, on peut en déduire les algorithmes et techniques de visualisation à assembler pour produire la plateforme ;

— pour finalement réaliser les développements nécessaires.

Bien que très contrôlée, cette façon de travailler permet d'arriver rapidement à un bon résultat qui satisfait vraiment les utilisateurs car chaque étape dispose de sa propre phase de validation après l'identification des risques associés. Je continue à utiliser cette méthodologie dès qu'il faut travailler avec des experts des données (cf. chapitre 4).

Ce chapitre synthétise le développement de la plateforme PORGY et présente les conclusions des travaux en suivant le modèle de Munzner. Après un rapide aperçu des travaux proches récents dans la section 2.1, la section 2.2 présente la formalisation et les visualisations associées aux différents éléments d'un système de réécriture. L'implémentation de PORGY qui est basée sur la plateforme TULIP est présentée dans le paragraphe 2.3. Les travaux présentés ci-dessous ont été principalement publiés dans Pinaud et al. (2012); Fernandez et al. (2014); Fernández et al. (2017, 2019); Andrei et al. (2019). De multiples références à ces articles sont faites dans la suite à chaque fois que le niveau de détail devient trop important. Des exemples de systèmes de réécriture développés avec PORGY sont présentés dans le chapitre 3.

4. Voir la page dédiée au prix du site de l'association EGC : https://www.egc.asso.fr/category/manifestations/prix-egc.

2.1 État de l'art

Contrairement à PORGY qui se veut un outil générique avec un modèle suffisamment souple pour être facilement adapté à différents types de problèmes, les autres systèmes, que nous connaissons, sont dédiés et optimisés pour la simulation et l'analyse de réseaux biologiques. RuleBender (Smith et al., 2012) est dédié à la modélisation à base de règles pour des réseaux biochimiques et plus précisément sur la dynamique intracellulaire. Il est basé sur la plateforme BIONETGEN (Faeder et al., 2009) pour la description et simulation de systèmes biochimiques exprimés par un modèle à base de règles. Contrairement à PORGY qui offre une vue dédiée pour chaque composant du système, RuleBender montre en une seule vue agrégée, baptisée « contact map », le réseau à réécrire et les règles en explicitant les potentielles interactions au sein du réseau avec les règles. Plus récemment, Mosbie (Wenskovitch et al., 2014) a été mis au point par la même équipe que RuleBender pour comparer interactivement et visuellement des modèles biochimiques grâce à une généralisation de la « contact map » et une mesure de similarité entre cartes. Enfin, la récente plateforme Kappa (Boutillier et al., 2018) apporte un environnement visuel intégré au langage Kappa (Danos et al., 2007, 2012) et partage la même idée originale que PORGY à savoir que la modélisation est semblable à de la programmation, et donc peut se faire dans un environnement qui reprend l'apparence et les fonctionnalités des environnements de développements intégrés (IDE) bien connus en programmation. Cette plateforme est une collection d'outils qui mis bout à bout forment une méthodologie et un environnement intégré complet de simulation et d'analyse visuelle de modèles à base de règles. Néanmoins, le langage Kappa de description des règles est principalement algébrique à la différence de PORGY où chaque composant du système peut être édité et analysé visuellement. Dans le domaine du développement logiciels, Henshin Arendt et al. (2010) permet d'ajouter à l'environnement de développement Eclipse des capacités de transformation de modèles pour simplifier la refonte et l'évolution de logiciels basés sur le modèle EMF (Eclipse Modeling Framework). Plus globalement, ces systèmes sont dédiées pour un type de donnée et ne proposent pas une approche entièrement basée sur la visualisation pour chaque étape du processus de réécriture. Ils ne contiennent pas non plus de langage pour piloter finement l'application des règles comme dans PORGY.

Il existe, néanmoins, de nombreux langages pour le pilotage de l'aplication de règles. Je ne cite ici que quelques exemples qui nous ont fortement inspiré. Un état de l'art plus complet est disponible dans Fernández et al. (2019). Tout d'abord, GP (Plump, 2011, 2009) est un langage non-deterministe basé sur des règles. Il utilise des règles textuelles, des expressions conditionnelles et des boucles ainsi qu'un système « à la Prolog » (*backtracking*) pour naviguer dans l'historique d'application des règles sans pour autant l'afficher en permanence comme dans PORGY (figure 2.1). PROGRES (Schürr et al., 1997) permet à l'utilisateur de contrôler l'application des règles en utilisant des expressions non-déterministes, des expressions conditionnelles et des boucles. En plus des expressions classiques décrites précédemment, GROOVE (Rensink, 2003) ajoute des choix aléatoires et la possibilité de tester l'application d'une règle sans pour autant l'appliquer et des appels de fonctions. GrGen.NET (Geiß et al., 2006) gère quant à lui différentes stratégies d'exécutions. PORGY reprend toutes ces fonctionnalités en y ajoutant un pilotage de la position dans le graphe à laquelle appliquer (ou non) une règle.

2.2 Synthèse du modèle de données

Nos premiers travaux se sont concentrés sur la modélisation de réseaux biologiques et plus précisément les réseaux d'interactions de protéines. Ces réseaux imposent des contraintes sur les graphes car deux protéines se connectent par l'intermédiaire de sites de connexions spécifiques

que l'on modélise avec un modèle de graphe appelé **graphe à ports** (section 2.2.1). Avec ce modèle, on génère un **graphe initial à réécrire** ainsi que des **règles de réécriture** modélisées par deux sous graphes à ports interconnectés (cf. section 2.2.2). L'application d'une règle produit une **dérivation.** L'ensemble des dérivations effectuées est conservé dans une trace complète appelée **arbre de dérivations** (section 2.2.3). L'application des règles peut être pilotée par une **stratégie** incluant des expressions conditionnelles, des boucles ou encore des opérations probabilistes regroupées dans un langage conçu spécifiquement (section 2.2.4). La section 2 de Fernández et al. (2019) définit formellement et en détail l'ensemble des concepts nécessaires pour la réécriture de graphes à ports. Ci-dessous, je présente une synthèse des définitions des différents composants clés sus-cités en conservant une approche systémique proche des travaux effectués pour implémenter la plateforme PORGY.

2.2.1 Graphes à ports

Intuitivement, un graphe à ports, noté **p-graphe** dans la suite, est un graphe non orienté dans lequel les sommets possèdent des zones de connexions spécifiques, appelées **ports**. Les arêtes sont alors attachées exclusivement aux ports. Les petites zones rouges et vertes sur la partie 2 de la figure 2.1 sont les ports qui permettent aux sommets modélisant des protéines de se connecter entre eux. L'avantage de l'utilisation des graphes à ports à la place d'une définition plus standard où les sommets sont connectés directement par des arêtes est de pouvoir exprimer facilement des attributs spécifiques de la connexion d'une arête à un sommet. Cette spécificité est notamment nécessaire pour les réseaux biologiques ou la modélisation de stratégie de communication dans des réseaux informatiques (Andrei, 2008). Dans le domaine de la visualisation, des contraintes de connexions des arêtes aux sommets sous la forme de ports sont utilisées depuis longtemps dans des outils tels que VCG (Sander, 1995) et yFiles (`http://www.yworks.com`) ou bien dans le format de description de graphes GRAPHML (Brandes et al., 2002). On trouve aussi des algorithmes adaptés à la gestion des ports notamment pour dessiner des graphes orientés (Gansner et al., 1993) ou des diagrammes de flux (Spönemann et al., 2010).

Pour la suite de ce document, un p-graphe est défini par $G = (V, P, E, D)_{\mathcal{F}}$ avec :

- V : un ensemble fini de sommets avec $n, n_1, \ldots$ la liste des sommets ;
- P : un ensemble fini de ports avec $p, p_1, \ldots$ la liste des ports ;
- E : un ensemble fini d'arêtes qui sont des paires de ports (p_i, p_j) avec $e, e_1, \ldots$ la liste des arêtes ;
- D : un ensemble fini de tuples dont chaque élément est le nom d'une propriété associée à une valeur autorisée (par ex. $\{color := red, label := SHC\}$) dans G ;

ainsi qu'un ensemble $\mathcal{F}$ de fonctions dont $Attach(p)$ qui donne pour tout port p le sommet auquel il est rattaché et $Label : V \cup P \cup E \rightarrowtail D$ qui associe à chaque élément de $V \cup P \cup E$ un tuple propriétés/valeurs de D.

L'exemple de la figure 2.3 modélise l'addition de deux nombres naturels exprimés selon les axiomes de Peano sous la forme d'un réseau d'interactions (Lafont, 1990). Un tel réseau décrit des agents (représentés par les sommets) interconnectés par l'intermédiaire de ports. Chaque agent a un port principal (noté 'P' sur la figure) et un certain nombre de ports auxiliaires (notés avec des chiffres). Un nombre est modélisé par un agent qui représente le chiffre 0 et une suite d'agents successeurs 'S' qui équivalent à additionner 1 au nombre précédent. Le port noté '2' de l'agent d'addition ('+') donne le résultat de l'opération. D'autres exemples incluant notamment des propriétés définies par le modélisateur ainsi qu'une extension de ce modèle avec des arcs, donc pour des graphes orientés souvent nécessaires pour modéliser des réseaux sociaux, sont utilisés dans le chapitre 3.

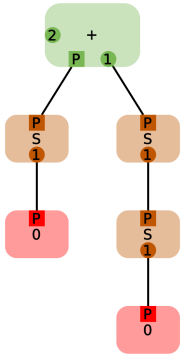


FIGURE 2.3 – Exemple de p-graphe pour modéliser une addition de deux nombres naturels sur la base des axiomes de Peano. Un nombre est modélisé par le chiffre 0 et une suite de successeurs. La figure modélise l'opération $2 + 1$. Ce graphe est à utiliser avec les règles de la figure 2.4.

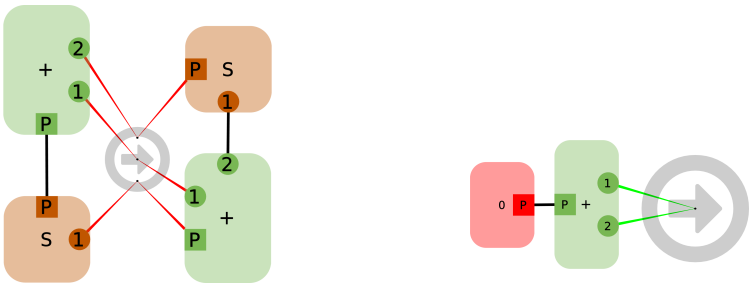
2.2.2 Règles de réécriture

Les règles de réécriture sont des représentations visuelles des transitions du système modélisé. Elles permettent une modélisation et surtout une visualisation simple et efficace du comportement du système contrairement aux représentations textuelles qui nécessitent plus d'efforts de la part de l'utilisateur. Une règle de réécriture, notée $L \to R$ est un p-graphe composé de (les définitions 4 à 6 de Fernández et al. (2019) proposent une formalisation détaillée) :

— Deux p-graphes disjoints notés L et R respectivement le membre gauche et le membre droit de la règle. Une réécriture consiste à instancier L dans un sous-graphe $g(L)$ du p-graphe G à réécrire et à le remplacer par $g(R)$, une instanciation de R dans G. $g(L)$ est alors un morphisme de L dans G formalisé par un ensemble de fonctions injectives de L dans G (images des ports, images des arêtes, sommets, etc.).

— Un nœud flèche ($\to$) équipé de ports dont les arêtes incidentes définissent les équivalences entre les sommets de L et de R. Ces équivalences sont nécessaires pour reconnecter les arêtes incidentes aux ports de $g(L)$ et qui ne font pas partie de $g(L)$ afin de ne pas laisser d'arête pendante dans G après réécriture. Ces équivalences permettent aussi de copier, adapter ou calculer les valeurs des propriétés des éléments de $g(L)$ dans $g(R)$ (par ex. le sommet doit devenir rouge après réécriture). L'utilisation de ports dans le nœud flèche au lieu de connecter des arêtes directement de L vers R simplifie fortement l'algorithme d'application d'une règle (cf. section 2.3.3).

La figure 2.4 présentent deux exemples de règles applicables sur le p-graphe de la figure 2.3. Intuitivement, l'idée est de vider une branche de l'agent '+' puis de le supprimer quand il ne reste plus qu'un '0' dans une branche (voir figure 2.5). La règle 2.4a qui modélise l'addition se résume ainsi $n + S(m) \to S(n + m)$. L'agent successeur qui était en paramètre de l'addition (port 'P') vient s'insérer à la sortie de l'addition (port '2'). Les agents connectés à la sortie de l'addition avant application de la règle seront reconnectés à la suite du port 'P' du nouvel agent 'S'. La suite de successeurs connectés au port 'P' de '+' contient alors un agent successeur en moins. Quand cette branche est vide de successeur, la règle 2.4b qui modélise une addition avec 0 est à utiliser. Elle supprime les agents '0' et '+'. Le port du nœud flèche et ses arêtes vertes incidentes indiquent qu'il faut connecter entre eux les images des ports de la règle situées aux extrémités opposées des images des arêtes incidentes aux ports '1' et '2' de l'addition. Il faut



(a) Interaction entre + et S. Cette règle peut s'exprimer ainsi $n + S(m) \rightarrow S(n + m)$.

(b) Addition avec 0.

FIGURE 2.4 – Règles pour l'addition de nombres naturels à utiliser avec le p-graphe de la figure 2.3.

donc relier la suite de successeurs connectés au port '1' de l'agent '+' à la suite de successeurs située sur le port de sortie '2'.

Pour l'application d'une règle, il est aussi possible de restreindre les endroits sur lesquels la règle peut s'appliquer. En effet, on peut souhaiter qu'un sommet particulier du graphe soit dans $g(L)$ ou au contraire surtout pas. Formellement, sur un p-graphe G, il est possible de définir deux sous-graphes P et Q qui représentent respectivement le graphe dont au moins un sommet doit faire partie de $g(L)$, c'est-à-dire $g(L) \cap P \neq \emptyset$ et à l'inverse Q est le graphe des éléments bannis qui ne peuvent pas être utilisés pour la réécriture $g(L) \cap Q = \emptyset$. En pratique Q est traité en priorité par rapport à P. Quand PORGY cherche un morphisme d'un membre gauche L_r d'une règle r dans un p–graphe G, il commence par chercher des équivalences des sommets de L_r dans $G \setminus Q$. Si un morphisme est trouvé, PORGY vérifie ensuite que $g(L) \cap P \neq \emptyset$. Ces mécanismes se sont avérés forts utiles pour les travaux sur les réseaux sociaux effectués dans le cadre de la thèse de Jason Vallet (chapitre 3).

2.2.3 Arbre de dérivations

Un atout de PORGY est la conservation de l'ensemble des tentatives, fructueuses ou non, d'applications de règles avec leurs paramètres (sous-graphes P et Q notamment) ainsi que des différents p-graphes générés. Cet historique sert ensuite à l'analyse du système (cf. chapitre suivant section 3.1) ou pour lancer de nouvelles applications de règles depuis n'importe quel p-graphe précédemment généré. Intuitivement, on modélise ainsi un arbre de dérivations dont les sommets sont les différentes évolutions du p-graphe initial et une arête représente l'application d'une règle. La figure 2.5 montre un extrait de l'application des règles présentées figure 2.4 appliquées sur l'exemple de la figure 2.3. La partie 4 de la figure 2.1 est un autre exemple. L'arbre de dérivations peut être vu comme un index visuel et interactif sur l'évolution du système. De récents travaux en visualisation qualifie une telle visualisation de visualisation multi-facettes (Hadlak et al., 2015, section 3.3.1), c'est-à-dire une visualisation qui montre plusieurs aspects des données visualisées en une seule visualisation. En plus de montrer les transitions dans le système et les différents états du p-graphe à réécrire, le dessin global de l'arbre de dérivations synthétise le temps qui passe (les différentes applications de règles) qui n'est évidemment pas linéaire dans le cas de la réécriture de graphe. L'application d'une règle est bien sûr non déterministe, une règle ne s'applique pas uniquement à un seul endroit sur le graphe à réécrire. Des branches parallèles dans

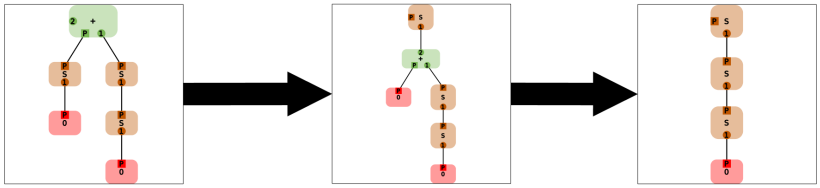


FIGURE 2.5 – Visualisation de deux étapes de réécritures du p-graphe de la figure 2.3 en utilisant les règles de la figure 2.4. Les arêtes correspondent à l'application d'une règle de réécriture. On commence par faire passer les successeurs d'une branche vers la sortie de l'agent '+' avec la règle de la figure 2.4a puis quand la branche est vide, c'est-à-dire qu'il ne reste qu'une addition avec '0', les deux agents sont supprimés à l'aide de la règle de la figure 2.4b et les listes de successeurs sont connectées pour former l'état final sur lequel aucune règle ne peut être appliquée.

l'arbre de dérivations peuvent ainsi être créées (Fig. 2.6, partie droite).

L'arbre de dérivations est un objet central du système de réécriture mais complexe. De nombreuses interactions sont disponibles dans PORGY ainsi que des visualisations dérivées (cf. chapitre suivant, section 3.1). Par exemple, au sujet du problème posé ci-dessus d'application d'une règle en plusieurs endroits, rien n'empêche d'obtenir au final deux p-graphes identiques en prenant des chemins différents dans l'arbre de dérivations. Cet arbre est donc dans le cas le plus général un graphe et les sommets identiques peuvent être regroupés pour faire apparaître des raccourcis ou des plus courts chemins comme publié dans Pinaud et al. (2012) et illustré figure 2.6.

2.2.4 Langage de stratégies

Les éléments présentés jusque ici sont encore insuffisants pour modéliser convenablement la dynamique d'évolution d'un système. Il manque un mécanisme de contrôle pour piloter l'application des règles que ce soit pour choisir la règle à appliquer et à quel endroit l'appliquer ou ne pas l'appliquer. Nous avons ainsi mis au point un langage de stratégie, pour répondre à ces objectifs. Le développement de ce langage a été un long processus depuis les premières itérations (Fernández et Namet, 2010; Andrei et al., 2011; Namet, 2011; Fernández et al., 2012) pour aboutir à une publication dans un journal international réputé (Fernández et al., 2019) qui présente une formalisation complète et détaillée de l'ensemble du langage en utilisant une sémantique opérationnelle à petit pas (Plotkin, 2004). Ce type de sémantique opérationnelle a l'avantage d'être très proche de l'implémentation réalisée. Le langage de stratégie de PORGY est inspiré de nombreux travaux dans ce domaine tels que les langages GP (Plump, 2009, 2011), GrGen (Geiß et al., 2006) ou GROOVE (Rensink, 2003) pour n'en citer que quelques uns. Néanmoins, le langage de stratégie de PORGY offre une caractéristique unique qui est le calcul de sous-graphes pour définir des positions où les règles peuvent ou pas s'appliquer.

Le tableau 2.1 présente la syntaxe abstraite complète du langage de stratégie structuré en quatre familles d'opérateurs. Tout d'abord l'application des règles (*Rules*), puis la définition des sous-graphes P et Q (*Positions*) qui peuvent être définis en fonction des valeurs des attributs des éléments (*Properties*) et enfin des compositions de stratégies (*Compositions)* pour obtenir des structures conditionnelles et des boucles. Une stratégie est générée à partir des règles de la grammaire en commençant par le non-terminal S. Une stratégie combine des applications de règles à partir du non-terminal A et met à jour les sous-graphes des endroits où appliquer la

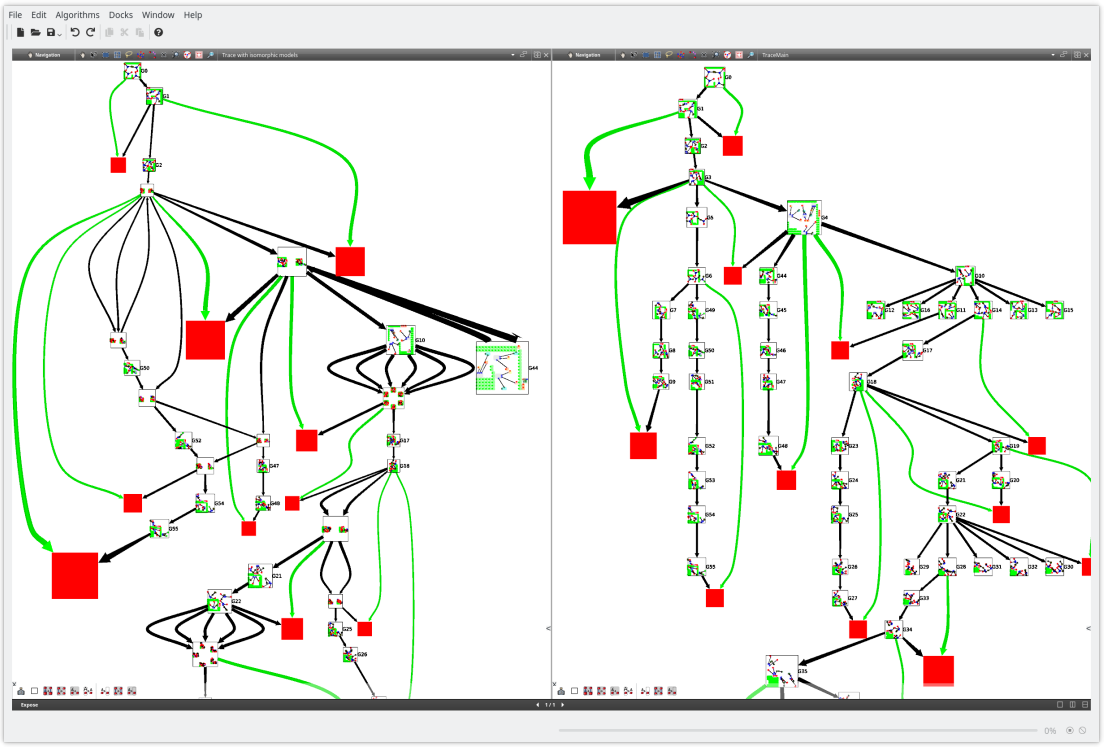


FIGURE 2.6 – Regroupement des sommets identiques (à gauche) d'un arbre de dérivations (à droite). Les sommets rouges représentent les tentatives infructueuses d'application de règles. Les arêtes vertes indiquent les états initiaux et finaux de l'exécution d'une stratégie.

Let L, R be port graphs; M, N subgraphs of R; W a subgraph of L;
$n, k \in \mathbb{N}$; $\pi_{i=1\ldots n} \in [0,1]$; $\sum_{i=1}^{n} \pi_i = 1$. Let *attribute* be an attribute;
e a valid value for it; *function* a computable function;
ComputedProbDist a probability distribution. $[x]$ means the item x is optional.

	(Probabilities)	Π	::=	$\{\pi_1, \ldots, \pi_n\} \mid \mathit{ComputedProbDist}$
Rules	**(Transformations)**	T	::=	$L_W \Rightarrow R_M^N \mid (T \parallel T)$
			$\mid$	$\texttt{ppick}(T_1, \ldots, T_n, \Pi)$
	(Applications)	A	::=	$\texttt{all}(T) \mid \texttt{one}(T)$
Positions	**(Focusing)**	F	::=	$\texttt{crtGraph} \mid \texttt{crtPos} \mid \texttt{crtBan}$
			$\mid$	$F \cup F \mid F \cap F \mid F \setminus F \mid (F) \mid \emptyset$
			$\mid$	$\texttt{ppick}(F_1, \ldots, F_n, \Pi)$
			$\mid$	$\texttt{property}(F, \mathit{Elem}[, \mathit{Expr}])$
			$\mid$	$\texttt{ngb}(F, \mathit{Elem}[, \mathit{Expr}])$
	(Determining)	D	::=	$\texttt{all}(F) \mid \texttt{one}(F)$
	(Updating)	U	::=	$\texttt{setPos}(D) \mid \texttt{setBan}(D) \mid \texttt{update}(\mathit{function})$
Properties	**(Properties)**	Elem	::=	$\texttt{node} \mid \texttt{edge} \mid \texttt{port}$
		Expr	::=	$\mathit{attribute}\ \mathit{Relop}\ e \mid \mathit{Expr}\&\&\mathit{Expr}$
		Relop	::=	$==\mid\neq\mid>\mid<\mid\geq\mid\leq\mid=\sim$
Compositions	**(Comparison)**	C	::=	$F = F \mid F \neq F \mid F \subset F$
			$\mid$	$\texttt{isEmpty}(F) \mid \texttt{match}(T)$
	(Strategies)	S	::=	$\mathsf{id} \mid \mathsf{fail} \mid A \mid U \mid C \mid S; S$
			$\mid$	$\texttt{if}(S)\texttt{then}(S)[\texttt{else}(S)]$
			$\mid$	$(S)\texttt{orelse}(S) \mid \texttt{repeat}(S)[(k)]$
			$\mid$	$\texttt{while}(S)\texttt{do}(S)[(k)] \mid \texttt{try}(S)$
			$\mid$	$\texttt{not}(S) \mid \texttt{ppick}(S_1, \ldots, S_n, \Pi)$

TABLE 2.1 – Syntaxe abstraite du langage de stratégie (extrait de Fernández et al., 2019).

règle (ou non) avec le non-terminal U en utilisant des expressions générées par F. La syntaxe concrète utilisable par les développeurs accompagnée d'exemples est disponible dans un rapport de recherche (Fernández et al., 2017). Le chapitre 3 décrit d'autres exemples d'usage.

2.3 Implémentation

Après avoir présenté de façon synthétique et illustré la démarche de combiner réécriture de graphes et visualisation dans la plateforme interactive Porgy, ses différents composants avec leur formalisation, nous allons maintenant aller voir sous le capot. Porgy est un ensemble d'une vingtaine de modules additionnels pour Tulip organisés autour de deux bibliothèques de codes partagées dont une couche d'abstraction du modèle de données de Tulip pour les graphes à ports. Porgy représente un peu plus de 30 000 lignes de code et utilise les mêmes technologies que Tulip à savoir le langage C++11, la bibliothèque Qt [5] pour les interfaces utilisateurs et leur gestion ainsi que plusieurs bibliothèques issues de Boost [6]. Ci-dessous, je présente synthétiquement Tulip et ensuite l'implémentation de Porgy ainsi que les algorithmes nécessaires à son bon fonctionnement.

2.3.1 La plateforme Tulip

Tulip est une plateforme de visualisation d'informations et de visualisation analytique. Initiée par David Auber en 2002, Tulip est un outil mature et reconnu mais qui reste destiné avant tout à des experts ou des développeurs expérimentés en visualisation. Un accompagnement (que j'effectue régulièrement pour différents publics dans différents contextes) pour prendre en main le logiciel reste indispensable. Tulip fait l'objet de publications régulières (Auber, 2004; Auber et al., 2014, 2016, 2017) ainsi que d'ajouts de fonctionnalités et de corrections par l'équipe de développement dont je suis un membre actif. Dans sa version actuelle (5.X), Tulip fournit des bibliothèques de programmation C++ et Python complète pour le développement de visualisations interactives pour des données relationnelles exprimées sous la forme de graphes. Le support de Python apporte une augmentation de l'interactivité de la plateforme grâce à la possibilité d'exécuter des scripts Python de quelques lignes à tout moment. Il est même possible d'appeler des scripts Python depuis du code C++ (fonctionnalité utilisée dans le langage de stratégie de Porgy).

Un graphe Tulip est composé de trois ensembles : un ensemble de sommets, un ensemble d'arêtes et un ensemble de propriétés dont certaines sont utilisées pour dessiner les graphes (par exemple, couleurs des éléments ou leur contour) ainsi que pour stocker les données de l'utilisateur. Tulip dispose aussi d'un vaste ensemble de modules additionnels pour notamment calculer diverses mesures sur les graphes (centralités, communautés, etc.), les dessiner, projeter des données sur la couleur des éléments ou leurs dimensions, et aussi pour réaliser diverses visualisations (par exemple des histogrammes bien utile pour Porgy, cf. chapitre 3) en complément du diagramme nœuds-liens traditionnel. La plateforme Tulip peut facilement être enrichie et personnalisée par l'ajout de modules additionnels écrits en C++ ou en Python.

Pour le développement de Porgy, la structure de données Tulip sous la forme d'une hiérarchie de sous-graphes ainsi que son mécanisme d'héritage de propriétés sur les sous-graphes sont fondamentaux pour limiter la consommation mémoire en ne dupliquant pas les sommets non modifiés par une étape de réécriture (plus de détails en 2.3.2). Porgy est simplement un ensemble de modules additionnels que ce soit pour l'interface utilisateur (appelé « Perspective »

5. https://www.qt.io
6. https://www.boost.org

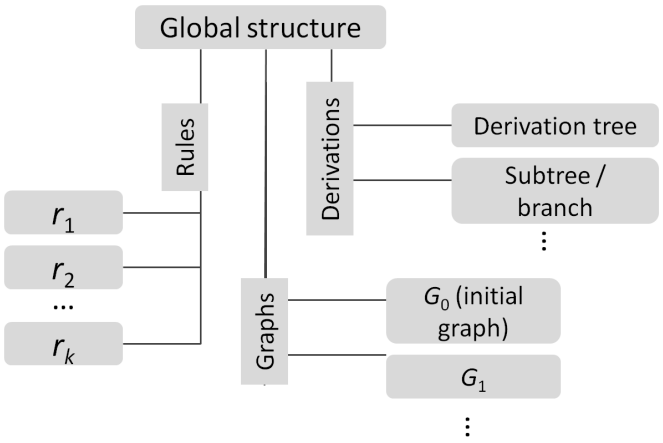


FIGURE 2.7 – Structure de données utilisée pour PORGY : une hiérarchie de sous-graphes TULIP qui permet de partager un maximum d'éléments sans les dupliquer.

dans le jargon TULIP, plus de détails en 2.3.5), les algorithmes de dessins de graphes adaptés aux p-graphes (appelé « LayoutAlgorithm » puisque l'objectif est de calculer des coordonnées pour les sommets des graphes) ou bien encore l'application d'une règle de réécriture qui est découpée en deux modules (plus de détails en 2.3.3), un pour chercher les morphismes du membre gauche, et le deuxième pour effectuer vraiment la réécriture à partir d'un morphisme choisi, et le langage de stratégie (cf. section 2.3.4) pour piloter les réécritures. PORGY utilise de nombreuses autres fonctionnalités de TULIP telles que la visualisation multi-facettes (Munzner, 2014; Hadlak et al., 2015) pour synchroniser de multiples visualisations entre elles (une modification sur une visualisation est immédiatement reportée sur les autres visualisations même si les éléments visualisés sont différents, figure 3.5 chapitre suivant) ou encore voir un sous-graphe dessiné dans un sommet comme pour l'arbre de dérivations (exemples des figures 2.5 et 2.6).

2.3.2 Structure de données et implémentation des p-graphes

PORGY nécessite une structure de données qui gère à la fois les propriétés visuelles basiques des graphes (par ex. la forme et la couleur des sommets) mais aussi des fonctionnalités avancées indispensables pour la construction et la manipulation de l'arbre de dérivations telles que la possibilité de ne pas dupliquer les éléments non modifiés par une réécriture lors de la création d'un nouvel état du p-graphe en cours de réécriture. Ce p-graphe à réécrire peut, de plus, être considéré comme un seul graphe dynamique qui évolue au gré des applications de règles. Néanmoins, un accès à chaque graphe dérivé du graphe original G_0 doit être préservé. Pour construire cet historique, les éléments communs entre les graphes (c'est à dire non modifié par une réécriture) doivent être partagés et non dupliqués. Par son mécanisme de sous-graphes et d'héritage de propriétés sur les éléments des sous-graphes, la structure de données de TULIP répond parfaitement à ces exigences. La figure 2.7 montre la structure de données utilisées qui est organisée en une structure globale (un graphe TULIP) qui contient l'ensemble des sommets et arêtes utilisés. Cette structure globale se décompose en trois entités spécialisées *Rules* pour les règles, *Graphs* pour les différents états du graphe à réécrire et *Derivations* pour les différents arbre de dérivations.

L'entité *Graphs* est initialisée avec un graphe G_0 qui contient l'état initial du système de réécriture. Chaque application d'une règle sur un graphe G_i revient à créer un nouveau sous-graphe G_{i+1} dont le parent est *Graphs*. Ainsi, les éléments communs entre deux sous-graphes sont

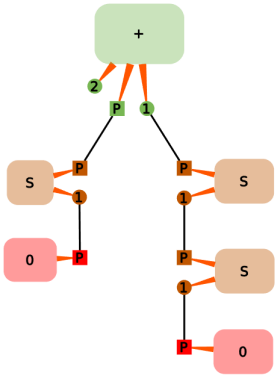


FIGURE 2.8 – Graphe à ports de la figure 2.3 montrant le détail de la structure sous-jacente dûe à TULIP. Chaque sommet central est connecté à ses ports par des arêtes (ici en orange). Cette structure est transparente pour l'utilisateur.

partagés depuis *Graphs* et non dupliqués afin de préserver la mémoire et l'efficacité du système. L'entité *Derivations* contient un arbre de dérivations principal *Derivation tree* dont des branches peuvent être isolées dans d'autres sous graphes pour des besoins particuliers. Les sommets de l'arbre de dérivations font directement références aux sous-graphes de *Graphs* pour visualiser le graphe en cours de réécriture à l'intérieur du sommet de l'arbre de dérivations (figure 2.5). Un autre avantage de ce partage de données est de pouvoir facilement suivre des éléments dans l'arbre de dérivations. La sélection d'un sommet n dans un sous-graphe de *Graphs* est immédiatement visible dans l'ensemble des sous-graphes où n est présent ainsi que sur les sommets de l'arbre de dérivations (figure 3.4 chapitre suivant).

La gestion des graphes à ports et plus précisément la gestion des ports est effectuée très simplement en utilisant un sommet central connecté en étoile avec d'autres sommets qui sont les ports (figure 2.8). Au travers de l'environnement graphique PORGY, la manipulation directe des centres, ports ou arêtes qui les relient n'est pas possible pour maintenir la cohérence du modèle (une arête ne peut relier que des ports).

2.3.3 Moteur de réécriture

Le moteur de réécriture tire aussi avantage de la hiérarchie de sous-graphes présentée précédemment. Une application de règle sur un graphe G commence par rechercher et conserver dans des sous-graphes de G les morphismes de son membre gauche. Puis, à partir d'un morphisme, la réécriture est effectuée et sauvegardée dans un nouveau sous-graphe de *Graphs*. Un nouveau sommet est alors ajouté dans l'arbre de dérivations pour représenter ce nouveau sous-graphe. Ces deux étapes sont chacune gérées par un module additionnel dédié succinctement détaillé dans les deux paragraphes suivants. La description formelle complète d'une étape de réécriture est à retrouver dans Fernández et al. (2019).

Recherche des morphismes du membre gauche. Le problème est de trouver des morphismes $g(L)$ du membre gauche L d'une règle, donc d'un graphe vers un sous-graphe. Ce problème est plus connu sous le nom de recherche d'isomorphisme graphe/sous-graphes. Nous utilisons simplement une adaptation de l'algorithme original de Ullmann (Ullman, 1976) afin de gérer les différentes propriétés à vérifier entre le membre gauche et le graphe ainsi que les parties

du graphe autorisées (P) ou interdites (Q) à la réécriture. Globalement, la complexité n'est pas gênante car les membres gauches des règles sont souvent petits par rapport à la taille des graphes à réécrire. En revanche, le nombre de morphismes trouvé peut être relativement important. Dans la structure de données, un morphisme de membre gauche $g(L)$ d'une règle r sur G se traduit par la création d'un sous-graphe de G. Ce nouveau sous-graphe contient seulement les images des éléments du membre gauche de la règle avec une propriété associée à chaque élément qui indique de quel sommet du membre gauche l'élément est image.

Application de la réécriture. À partir du sous-graphe d'un morphisme $g(L)$ dans G, un nouveau sous-graphe G' de *Graphs* clone de G est ajouté. Une copie du membre droit de la règle $g(R)$ est ajoutée dans G'. Les arêtes représentées en rouge dans les règles (Fig. 2.4a) indiquent les équivalences entre les ports de L et R. Il est ainsi possible de reconnecter les extrémités des arêtes incidentes à un port de $g(L)$ et de recopier ou calculer les valeurs des propriétés associées aux ports de $g(L)$. Enfin, le morphisme du membre gauche de la règle est supprimée de G' pour obtenir le graphe final. L'arbre de dérivations est alors mis à jour en ajoutant un nouveau sommet qui fait référence à G' ainsi qu'une arête entre les sommets qui représentent G et G'.

2.3.4 Langage de stratégies

L'exécution des stratégies s'effectue aussi grâce à un module additionnel Tulip qui se présente comme un interpréteur. En entrée, ce module a besoin d'un graphe et d'une stratégie. En sortie, le module effectue les opérations bas-niveaux nécessaires sur l'ensemble de la hiérarchie de sous-graphes. J'ai écrit cet interpréteur à l'aide de la librairie Spirit (sous-librairie Qi pour être précis) de Boost qui est dédiée à l'écriture d'analyseurs lexicaux et syntaxiques. Grâce à Spirit la grammaire présentée dans le tableau 2.1 est implémentée exactement comme nous l'avons présentée dans Fernández et al. (2019) grâce à la sémantique opérationnelle à petits pas. Le nombre de lignes de code dédié de ce module additionnel reste limité malgré la richesse du langage (∼2300 lignes de C++).

2.3.5 Interface utilisateur

L'interface utilisateur est elle aussi construite sur des modules additionnels. Le principe est classique à savoir Modèle-Vue-Contrôleur (MVC) bien connu en génie logiciel. Un modèle (M) de données (la hiérarchie de sous-graphes) est visualisé (V) de différentes façons et est interrogé ou mis à jours par l'intermédiaire de contrôleurs (C). Chaque composant du système de réécriture (graphe à réécrire, règles et arbre de dérivations) est visualisé grâce à une « vue » (jargon Tulip) dédiée. Chaque vue possède un ensemble d'interactions dédiées afin d'interroger ou modifier le modèle de données. L'ensemble de ses actions est coordonné par une couche métier dédiée, appelée « perspective », qui est en fait tout simplement l'interface utilisateur (figure 2.1). Grâce à la perspective et au modèle de données Tulip, l'application d'une règle ou d'une stratégie se fait très intuitivement par glisser-déposer sur les sommets de l'arbre de dérivations (figure 2.9).

2.4 Synthèse du chapitre

Ce chapitre est dédié à la présentation des travaux et développements qui ont abouti à la plateforme Porgy notamment la méthodologie visuelle et interactive mise au point pour la modélisation, simulation et analyse de systèmes complexes exprimés à l'aide du paradigme de la réécriture de graphes. L'idée directrice est de permettre l'étude du comportement global d'un système complexe à partir d'un ensemble de modifications locales contrôlées et pilotées. J'ai

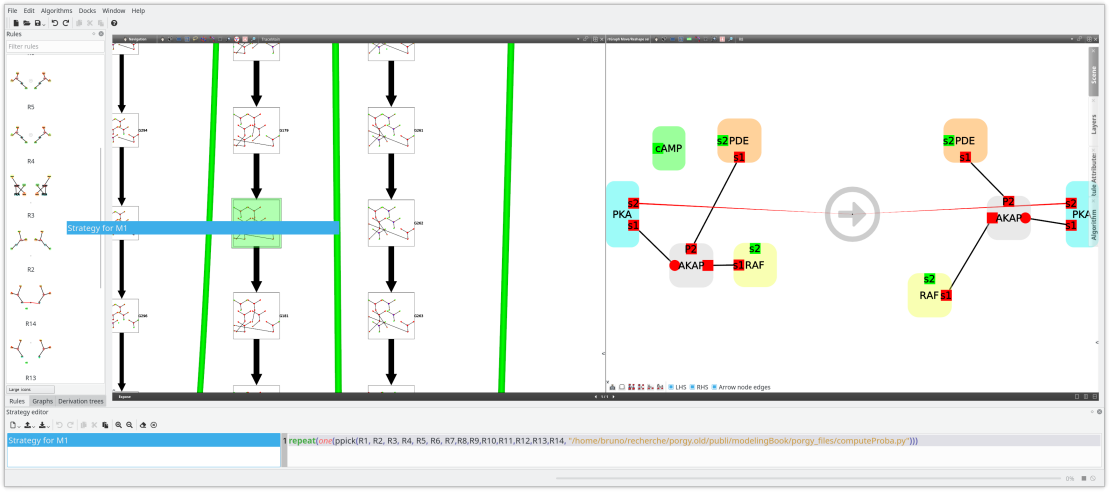


FIGURE 2.9 – Glisser-Déposer d'une stratégie sur un sommet de l'arbre de dérivations pour lancer son exécution. Le fond vert du sommet indique simplement que la stratégie sera appliquée sur le sommet.

présenté et illustré ces travaux, initiés dès mon arrivée au LaBRI, à savoir la formalisation et l'implémentation de PORGY.

Un système de réécriture de graphes exprimé avec PORGY nécessite trois composantes :

— une modélisation de l'état initial du système décrit par un graphe à ports ;

— une description des étapes d'évolution du système par des règles de réécriture pour exprimer l'ensemble des modifications locales qui peuvent intervenir ;

— une couche de contrôle sous la forme d'une stratégie décrite par un langage spécifiquement développé pour définir une simulation du système : où appliquer (ou ne pas appliquer) les règles dans le graphe à réécrire, quelles règles ou combinaisons de règles appliquer et les conditions d'arrêt de la simulation.

Les résultats (positifs ou négatifs) des simulations sont rassemblés dans un historique visuel et interactif appelé arbre de dérivations. Cet arbre peut ensuite être analysé. Des exemples d'analyses sont présentés dans le chapitre suivant. Les systèmes biologiques sont de loin les plus utilisés dans les outils proches de PORGY que nous connaissons. Néanmoins, PORGY a été développé pour être suffisamment souple et générique pour modéliser un très grand nombre de systèmes variés. Le chapitre suivant présente plusieurs applications de PORGY notamment sur des réseaux biologiques ou sociaux afin de montrer en particulier toute la souplesse et la généricité du modèle de PORGY. Ce chapitre inclut aussi un bilan des perspectives et des travaux restant à effectuer sur la plateforme PORGY.

Chapitre 3

Applications de PORGY

Sommaire

Le chapitre précédent décrit les travaux formels et les développements qui ont conduit à la plateforme PORGY et à sa méthodologie associée combinant réécriture de graphes et visualisation pour la modélisation, la simulation et l'analyse de systèmes complexes. Cette méthodologie se résume simplement : à partir d'un réseau exprimé sous la forme d'un graphe qui modélise l'état initial d'un système, d'un ensemble d'opérations locales qui décrivent la dynamique de ce système sous la forme de règles de réécriture ainsi qu'une stratégie pour piloter l'application des règles, on souhaite visualiser et étudier le comportement global du système modélisé. La modélisation complète du système (état initial, règles et stratégie) est appelée un système de réécriture. Dans ce chapitre, je présente et illustre différents systèmes de réécriture que nous avons mis au point pour avant tout démontrer différents usages de PORGY :

1. la généricité et la souplesse en travaillant sur divers types de données et en produisant des systèmes de réécriture paramétrables ;
2. la possibilité de montrer des propriétés importantes des systèmes modélisés comme leur terminaison ou confluence ;
3. l'utilisation de la visualisation pour produire et permettre à l'utilisateur d'interagir avec les graphes à différentes échelles du système de réécriture ;

4. l'utilisation de la réécriture comme un dénominateur commun pour comparer intrinsèquement des systèmes,
5. et ainsi faire émerger les briques de base de ces systèmes pour produire de nouveaux systèmes combinant ces briques avec des propriétés nouvelles à l'image du fonctionnement d'un algorithme génétique (référence à mes travaux de doctorat (Kuntz et al., 2006)).

L'avantage 1 est illustré par la présentation de plusieurs exemples basés sur des systèmes biologiques et des réseaux sociaux. Les travaux sur l'exemple biologique présentés en section 3.1 illustrent l'avantage 2. Cet exemple est utilisé depuis nos premières publications (Andrei et al., 2011; Pinaud et al., 2012; Fernandez et al., 2014) et repris complètement dans un chapitre de livre (Andrei et al., 2019). Les avantages 3 et 4 font écho à de nombreuses problématiques dont celles liées à la construction et l'analyse de réseaux sociaux et leur dynamique (Newman et al., 2006; Brandes et Wagner, 2004). Cette thématique de recherche est en pleine expansion grâce notamment aux importants volumes de données disponibles de nos jours et à la multiplication des réseaux facilement accessibles sur le web (Facebook, Twitter, LinkedIn, ...). Parmi les nombreuses problématiques existantes autour de l'analyse des réseaux sociaux, nous nous sommes intéressés :

— à la construction de réseaux en imitant des interactions réellement observées entre des acteurs humains ;

— aux phénomènes de *propagation* d'information reconnu comme un des problèmes les plus importants dans les réseaux sociaux (Golbeck, 2013) dans lequel des utilisateurs propagent consciemment une information vers leurs voisins et ainsi de suite (par ex. annonce d'un évènement, diffusion de rumeurs) ;

— aux phénomènes de *diffusion* qui consistent à diffuser automatiquement des informations à travers le réseau.

Les phénomènes de propagation et de diffusion sont souvent exprimés par des modèles mathématiques abstraits (Kempe et al., 2003; Goyal et al., 2010; Giakkoupis et al., 2015) qui reposent comme la réécriture sur le principe d'actions locales (les individus communiquent uniquement avec leurs voisins directs) qui ont des conséquences sur l'ensemble du réseau (Est-ce que tous les individus ont reçu l'information ? Est-ce que l'information se propage bien et rapidement à l'ensemble des acteurs du réseau ?). Les communications entre individus peuvent donc s'exprimer par des règles de réécriture. Au delà de la contribution de modéliser les phénomènes de propagation et dissémination avec la réécriture de graphes, l'objectif visé est d'aboutir à une méthode pour étudier des modèles et les comparer pour, par exemple, comprendre où se situent les différences entre eux. Ces travaux sont issus de la co-direction avec Guy Melançon de la thèse de Jason Vallet menée entre novembre 2013 et décembre 2017 (Vallet, 2017) grâce à une bourse ministérielle et la collaboration sans faille avec Maribel Fernández (King's College London) et Hélène Kirchner (Inria). Les prémisses de ces travaux ont été présentés à la première conférence européenne sur les réseaux sociaux (Vallet et al., 2014). Cette conférence nous a permis de rencontrer V. Batagelj et ses travaux sur la génération de réseaux aléatoires (Kejžar et al., 2008) à partir d'une modélisation des interactions observées au sein de groupes de personnes. Suite à cette rencontre, nous avons repris et adapté les travaux de Batagelj en proposant un modèle de génération de graphes petits mondes (section 3.2). La section 3.3 présente ensuite la modélisation de deux modèles de propagation les plus souvent rencontrés dans la littérature ainsi qu'un modèle de diffusion puis son hybridation avec un des modèles de propagation présenté précédemment. Ces travaux ont nécessité un important travail de développement (pour moi et le doctorant) pour rendre la plateforme PORGY la plus générique possible : gestion des graphes orientés (ou non), lors de la recherche d'un membre gauche, vérification qu'une arête n'existe pas, possibilité de filtrer sur l'ensemble des éléments et propriétés du graphe, lors de la phase de réécriture, possibilité de

calculer dynamiquement une nouvelle valeur pour les propriétés des éléments réécrits. Les détails de ces travaux de développement ne sont pas présentés dans ce manuscrit. Des versions préliminaires de ces travaux sur les réseaux sociaux ont été publiées (Vallet et al., 2015; Fernández et al., 2016) pour au final aboutir à une publication qui reprend l'ensemble des résultats dans un journal international (Fernandez et al., 2018). À noter que l'article (Vallet et al., 2015) a reçu le prix du meilleur article académique lors de la conférence Extraction et Gestion des Connaissances (EGC) 2015. La thèse de Jason a reçu le prix EGC de la meilleure thèse lors de l'édition 2019 de la conférence.

3.1 Analyse d'un système biologique

Dès le commencement de nos travaux en 2009, PORGY est utilisé sur des systèmes biologiques plus précisément des réseaux d'interactions de protéines. L'exemple utilisé peut se résumer ainsi de mon point de vue de néophyte du domaine : à partir de différentes protéines dont les interactions sont modélisées par différentes règles (la figure 2.1 du chapitre précédent en montre quelques unes), nous souhaitons étudier des voies métaboliques (simulées par des applications successives de différentes règles) qui jouent un rôle important dans le développement de la cellule associée et sa résistance aux médicaments. Nous souhaitons vérifier les hypothèses des biologistes, à savoir que le fonctionnement normal du système revient à dégrader des protéines en d'autres protéines et que la vitesse de dégradation est liée à des interconnexions spécifiques préalables avec d'autres protéines. Selon les biologistes, la création des protéines au fur et à mesure de l'évolution du système dans le temps doit prendre la forme d'un escalier (on suit particulièrement une protéine nommée SA). Je ne suis pas spécialiste des réseaux biologiques, donc je ne vais pas rentrer plus en avant dans les détails du fonctionnement du réseau. Le lecteur intéressé pourra se référer à Andrei et al. (2019). Je ne présente pas non plus la modélisation du réseau mais l'analyse par la visualisation analytique des résultats de simulations de deux versions de ce système baptisées M_1 et M_2 à partir d'un même graphe initial. La différence entre les versions est la présence de plus d'interconnexions spécifiques initiales entre les protéines dans M_1 que dans M_2. Le modèle M_1 est donc composé de plus de règles que M_2 (14 contre 7) afin de gérer convenablement toutes les interactions liées à la présence de ces interconnexions supplémentaires. Si notre modélisation est correcte, la pente de l'escalier doit être plus raide dans M_1 (donc dégradation plus rapide, la concentration de protéine SA doit augmenter plus vite) que dans M_2.

Le code des stratégies associé à M_1 et M_2 est équivalent (le code associé à M_1 est donné dans la stratégie 3.1). Une répétition a lieu autant de fois que possible (`repeat()`) d'un choix probabiliste (`ppick()`) entre les règles données par les biologistes (14 règles pour M_1 notées $R_1, R_2, \ldots, R_{14}$). Dès qu'une règle est choisie, une seule application de cette règle est effectuée parmi l'ensemble des applications possibles (`one()`). Pour cet exemple, les probabilités d'application des règles ne sont pas données *a priori* (cette possibilité existe néanmoins dans le langage) car les probabilités évoluent dynamiquement au fur et à mesure de l'évolution du système (plus une règle peut s'appliquer sur le système plus sa probabilité d'application sera forte). Les probabilités sont en fait calculées grâce à un programme écrit en Python (« *ComputeProba.py* ») et exécuté à chaque appel de l'opérateur de choix probabiliste `ppick()`.

Une fois les simulations terminées, leur analyse s'effectue conformément au processus traditionnel de visualisation d'information synthétisé par Shneiderman (1996) : « *Overview first, zoom and filter, then details on demand* ». À partir d'une vue d'ensemble des données, l'utilisateur peut effectuer des zooms et des filtrages par différents moyens afin de pouvoir afficher plus de détails sur les données sélectionnées et ainsi augmenter sa compréhension du système modélisé.

Stratégie 3.1 : Stratégie associée au modèle M_1 pour l'appel des 14 règles du modèle.

```
repeat(
  one(
    ppick(R1, R2, R3, R4, R5, R6, R7, R8,
      R9, R10, R11, R12, R13, R14,"ComputeProba.py")
  )
)
```

3.1.1 Vue d'ensemble

Tout d'abord à la fin de l'exécution de la stratégie qui marque la fin de la simulation, l'arbre de dérivations peut être visualisé en intégralité grâce à PORGY. Les figures 3.1 et 3.2 montrent les arbres de dérivations générés par 10 répétitions des stratégies associées aux modèles M_1 et M_2 à partir du même état initial. Les arêtes vertes connectent l'état initial du système et l'état final atteint à la fin de la stratégie. Ces visualisations sont évidemment difficiles à analyser en l'état. On remarque néanmoins l'aspect stochastique des stratégies puisque les différentes branches qui symbolisent les différentes simulations ont un nombre de sommets représentant les états intermédiaires du système de réécriture différents. De plus, quelques interactions issues de TULIP peuvent s'avérer bien pratique telle que la loupe illustrée figure 3.2.

Au lieu de regarder l'ensemble d'un arbre de dérivations, il est possible de se concentrer sur une branche et la visualiser soit à l'aide de vignettes soit d'une animation, les deux étant complémentaires (Archambault et al., 2011). Les figures 3.3 et 3.4 montrent chacune un extrait d'une branche de l'arbre associé à M_1. On peut notamment voir distinctement où s'applique une règle et les changements qu'elle implique (figure 3.3) ou encore visualiser quand un sommet est créé ou modifié par une réécriture (figure 3.4).

Après avoir visualisé notre système en prenant un point de vue large, nous pouvons maintenant nous concentrer sur les détails de son évolution et particulièrement la protéine **SA**. PORGY fournit une large palette d'outils de visualisation analytique pour étudier le système et comparer les résultats des exécutions des stratégies associées aux modèles M_1 et M_2.

3.1.2 Zoom et filtrage

L'évolution de la concentration de la protéine **SA** que souhaite vérifier les biologistes est à mesurer le long des branches de l'arbre. Le temps qui passe est marqué ici par la profondeur de la branche. Après avoir sélectionné une branche de l'arbre de dérivations, PORGY permet d'isoler cette branche pour compter les différents types de sommets présents. Une courbe d'évolution du nombre de protéine **SA** pour les modèles M_1 (figure 3.5) et M_2 (figure 3.6) est ainsi produite. On remarque immédiatement que les courbes ont la forme en escalier attendue par les biologistes mais que, en revanche, la vitesse d'évolution est différente (attention, les figures ne sont pas à la même échelle). Ces résultats sont ceux attendus par les biologistes.

Grâce à TULIP les sommets de ces courbes sont les sommets des arbres de dérivations associés et donc des états du système. La synchronisation automatique et transparente de l'ensemble des visualisations permet de sélectionner des sommets sur la courbe d'évolution, et cette sélection est immédiatement reportée sur l'arbre de dérivations (les sommets bleus de la figure 3.5). Ces états intéressants du système pour l'expert peuvent alors être étudiés plus précisément.

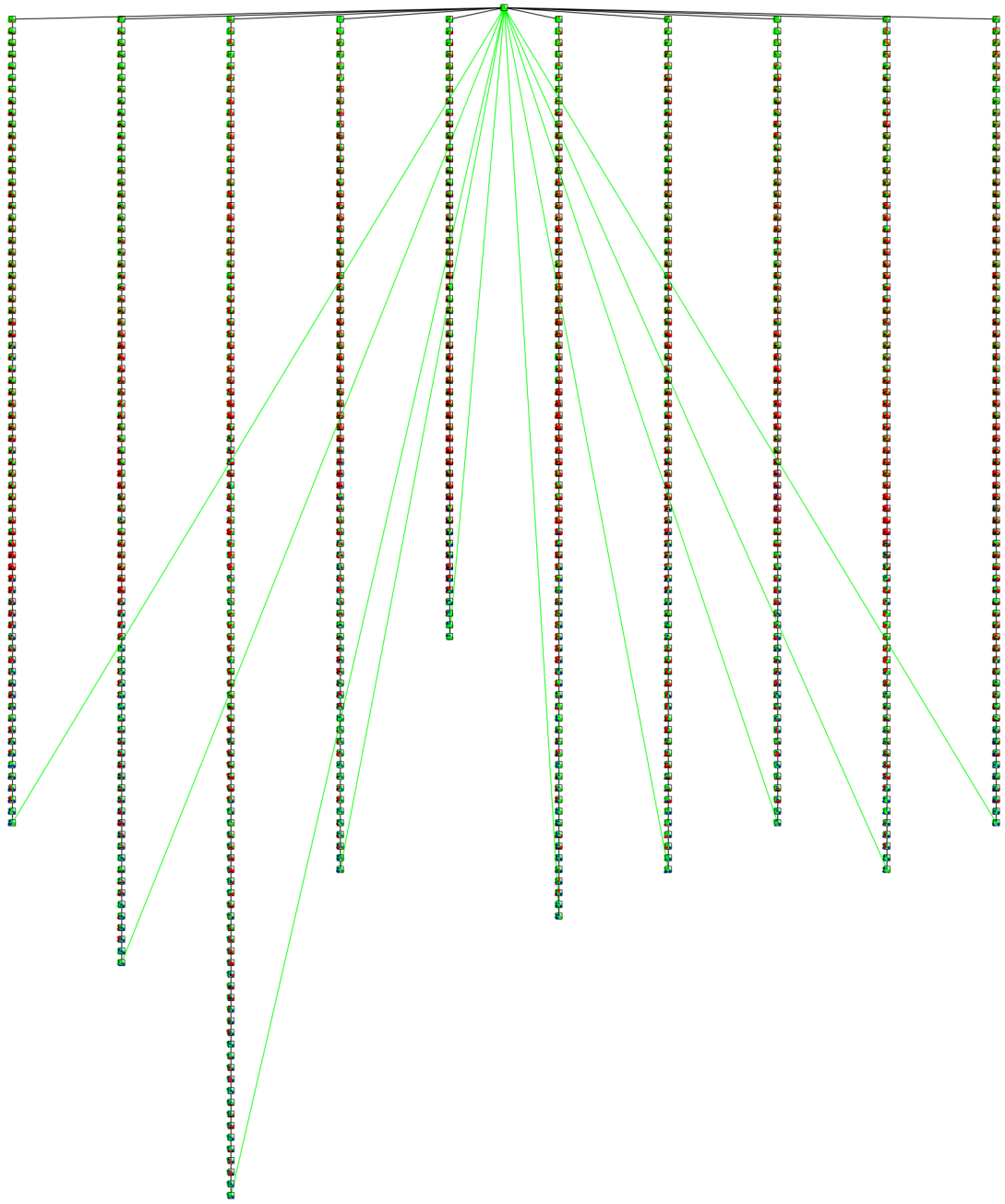


FIGURE 3.1 – Arbre de dérivations associé au modèle M_1 pour 10 répétitions de la stratégie 3.1. La branche la plus longue possède 100 étapes intermédiaires. La profondeur des branches est différente car les séquences d'application de règles sont calculées aléatoirement par la stratégie. Une arête verte connecte l'état initial utilisé pour démarrer la stratégie à l'état final associé.

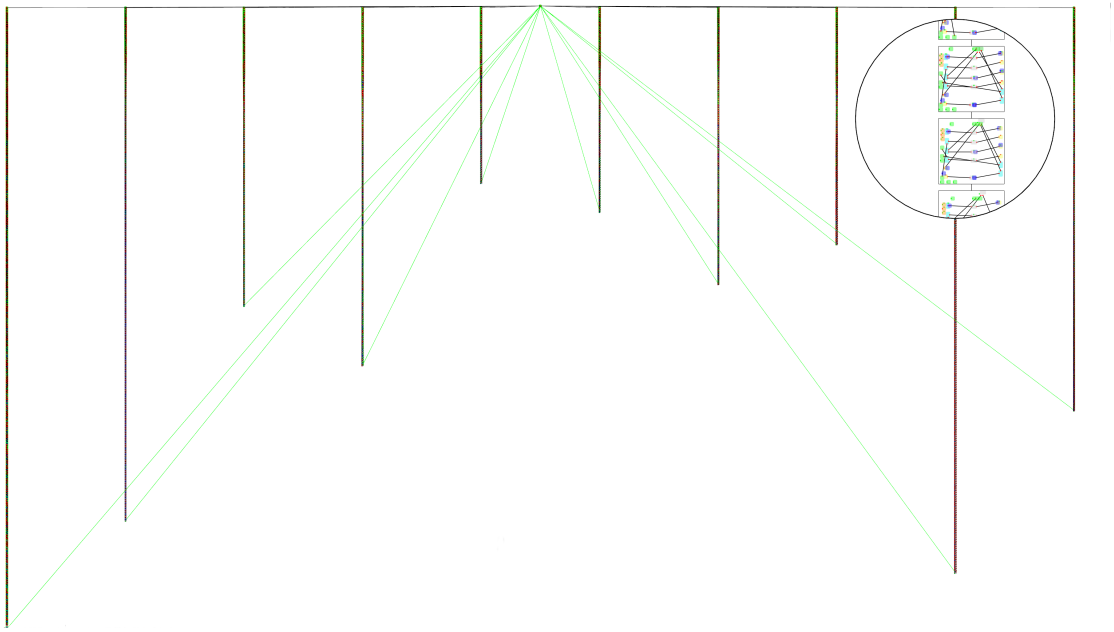


FIGURE 3.2 – Arbre de dérivations associé au modèle M_2. La stratégie a été répétée 10 fois à partir de l'état initial. La branche la plus longue possède 343 états intermédiaires. Un agrandissement de quelques états d'une branche (interaction disponible dans PORGY) est réalisé en haut à droite de la figure. La profondeur des branches est différentes à cause de l'usage de probabilités dans la stratégie afin d'éviter d'avoir deux fois la même séquence de règles pendant les simulations. Une arête verte connecte l'état initial utilisé pour démarrer la stratégie à l'état final associé.

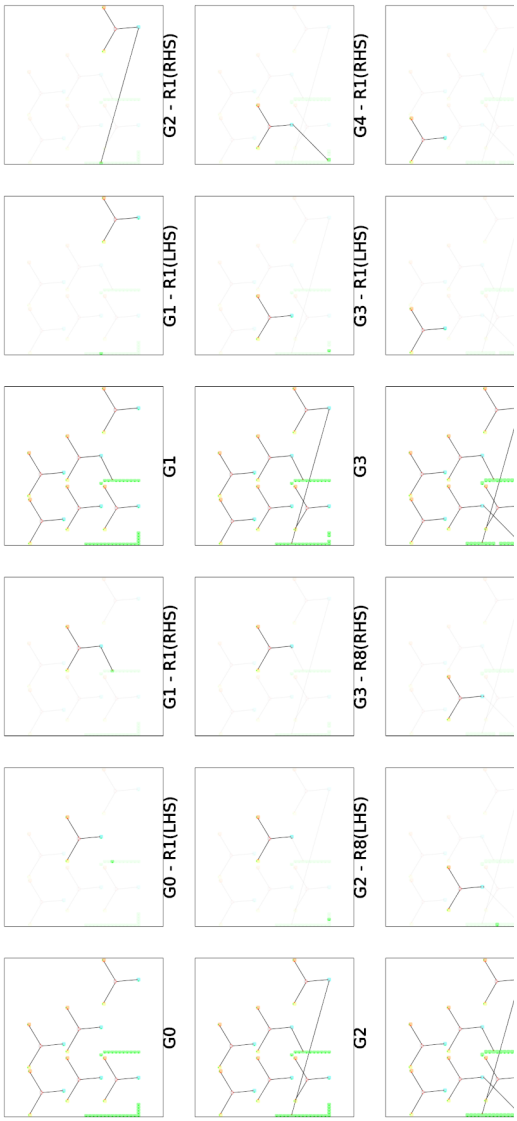


FIGURE 3.3 – Vue en format vignettes d'une dérivation du modèle M_1 (une branche de l'arbre). La première vignette (G_0) est le graphe initial, la suivante montre le morphisme du membre gauche de la règle utilisée (ici **R1**), la suivante montre à son tour les modifications réalisées par la réécriture et enfin le graphe G_1 complet et ainsi de suite.

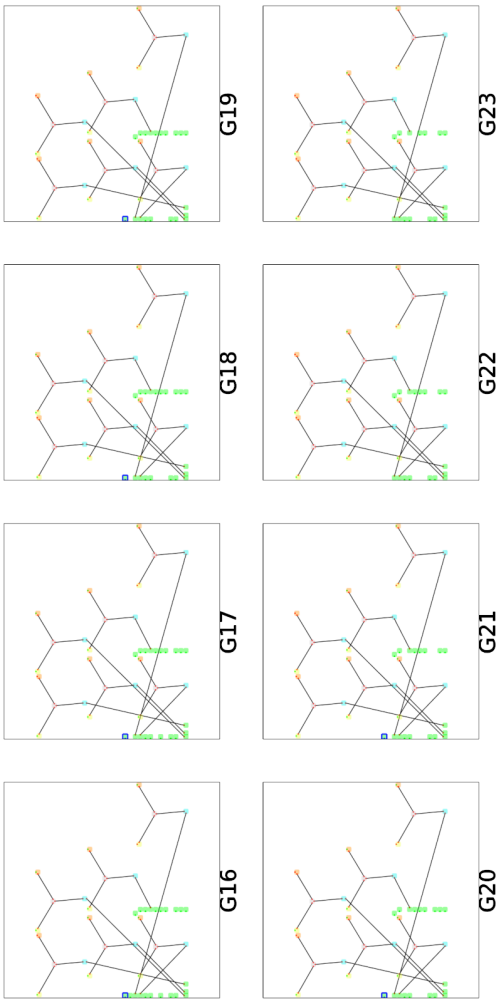


FIGURE 3.4 – Suite de la figure 3.3 sans les détails des applications des règles. A noter que de G_{16} à G_{21} un sommet non modifié est sélectionné (carré bleu sur la gauche de chaque vignette). Ce sommet n'est plus sélectionné sur G_{22}. On peut en déduire immédiatement que ce sommet a été modifié par la réécriture qui a permis de créer G_{22}.

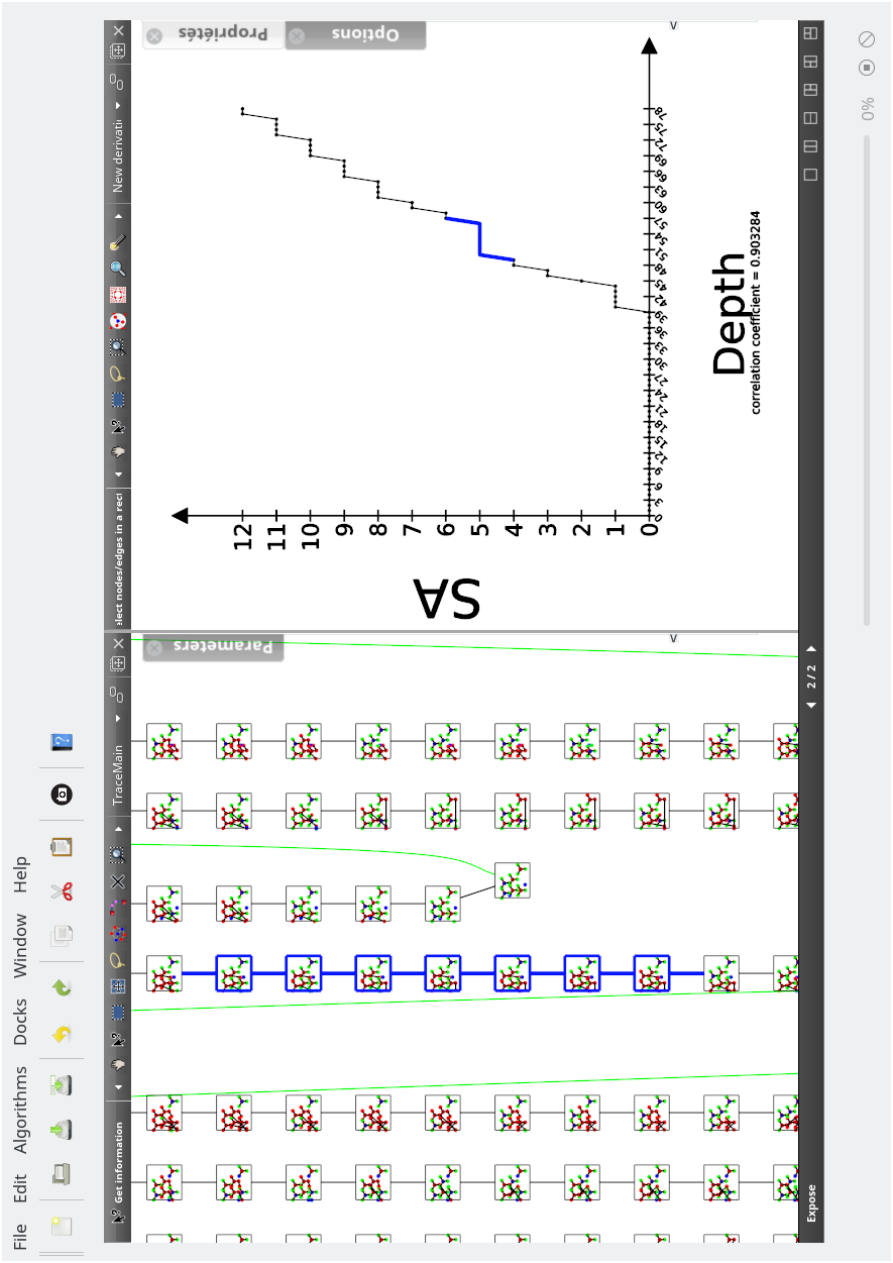


FIGURE 3.5 – Zoom sur l'arbre de dérivations de la figure 3.1 (partie gauche) et visualisation de l'évolution du nombre de protéines **SA** (partie droite) pour M_1. Les points sur la courbe sont les sommets de la branche de l'arbre de dérivations. La sélection de sommets sur la courbe est automatiquement reportée sur l'arbre de dérivations.

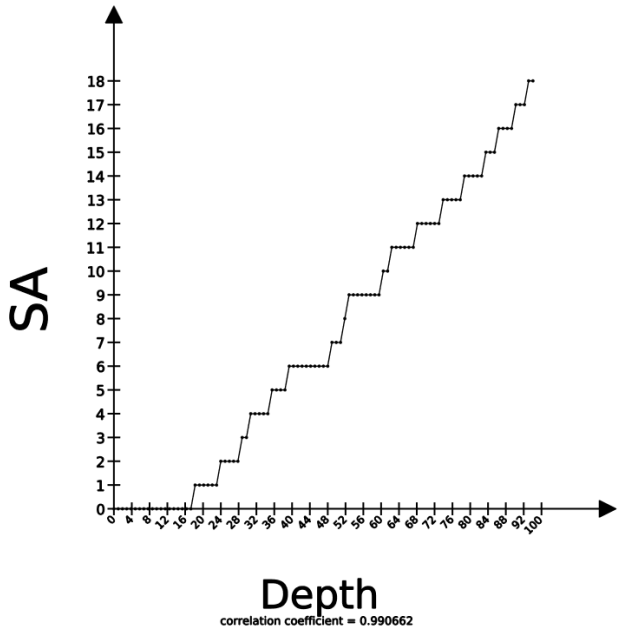


FIGURE 3.6 – Évolution de SA au fur et à mesure des réécritures pour M_2.

3.1.3 Détails à la demande

Après avoir identifié des états d'intérêts du système, il est possible de zoomer plus fortement sur les sommets de l'arbre de dérivations pour voir le graphe contenu dans chaque sommet plus en détails. Il est, par exemple, possible de faire simplement passer le pointeur de la souris au-dessus d'une arête noire qui indique une application de règle. Les sommets et arêtes supprimés, créés ou modifiés par l'application de la règle sont mis en valeur (figure 3.7). Les sommets modifiés par la réécriture ont un aspect normal. Les sommets et arêtes non modifiés deviennent transparents. On voit ainsi immédiatement les modifications engendrées par l'application de la règle.

3.2 Génération de réseaux aléatoires en simulant des interactions entre personnes

L'utilisation de générateurs de réseaux aléatoires pour valider des algorithmes ou estimer des statistiques est une pratique courante. En revanche, l'implémentation de ces générateurs est un problème complexe notamment si on souhaite obtenir des propriétés particulières ou s'assurer de l'aspect réellement aléatoire du réseau généré (Melançon et Philippe, 2004; Sallaberry et al., 2013). En utilisant comme inspiration les travaux présentés dans Kejžar et al. (2008) (qui fait de la réécriture sans le mentionner) et les règles décrites dans cet article, notre objectif est de générer des réseaux dont les liens simulent des interactions entre personnes comme dans la vraie vie. Dans un article fondateur sur l'analyse de réseaux sociaux, toujours très cité, Borgatti et al. (2009) font état de quatre types de liens dans un réseau social : liens de similarités (par ex. même lieu, même âge), liens de relations (par ex. ami de, superviseur de, aime, déteste), liens d'interactions (par ex. parle à, aide, préviens) et liens de flux (par ex. informe). De plus, les réseaux générés

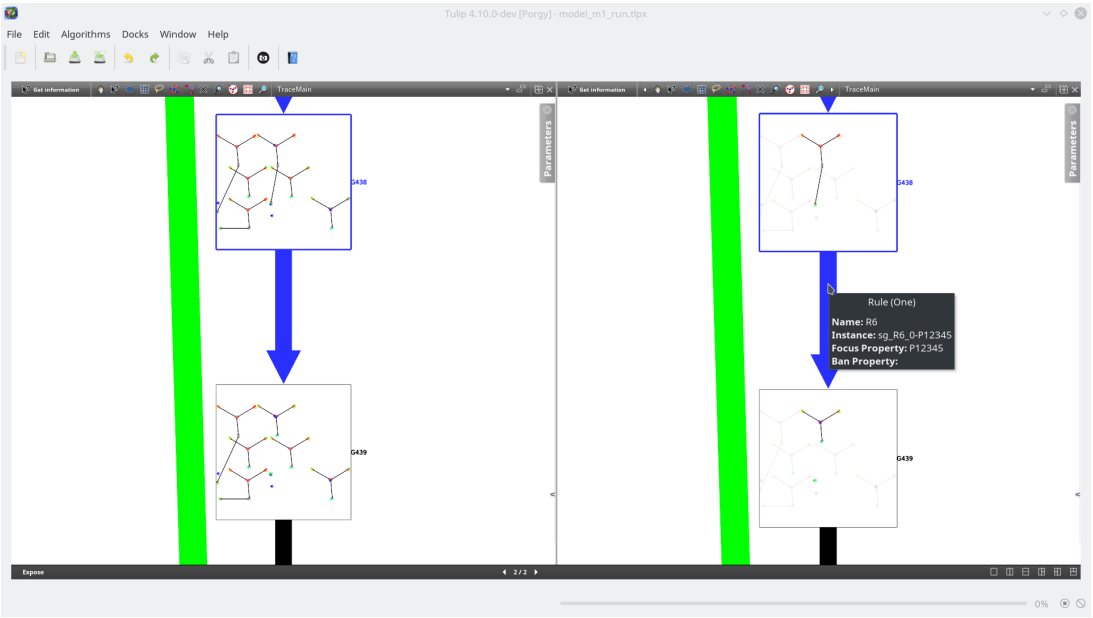


FIGURE 3.7 – À partir d'une vue normale (à gauche), il est possible de faire passer le pointeur de la souris au dessus d'une arête de l'arbre de dérivations afin de pouvoir visualiser directement l'endroit où s'applique la règle et les modifications engendrées. Des informations sur l'environnement d'exécution de la règle sont aussi affichées.

sont dit « petit monde ». Ils ont ainsi un petit diamètre (distance moyenne entre chaque paire de sommets courte) et un coefficient d'agglomération (clustering) moyen élevé caractérisé par une tendance des sommets à se connecter prioritairement avec leurs voisins pour créer des groupes de sommets fortement connectés entre eux appelés *communautés* (Watts et Strogatz, 1998). Ces propriétés sont pertinentes pour réaliser ensuite une étude de modèles de propagation et de diffusion d'informations (sections suivantes) sur de tels réseaux car ces propriétés ont un impact fort sur une diffusion de l'information rapidement à l'ensemble du réseau.

3.2.1 Modélisation des interactions

À partir d'un sommet, la génération s'effectue en trois étapes. Tout d'abord, de nouveaux utilisateurs rejoignent le réseau. Ils sont connectés aux utilisateurs les ayant invités (figure 3.8). Ces nouveaux utilisateurs découvrent ensuite le réseau, et bien souvent de nouvelles connaissances, ce qui créent de nouvelles connexions (figure 3.9). Enfin, les utilisateurs peuvent, par exemple, rencontrer les amis de leurs amis et ainsi créer de nouvelles connexions (figure 3.10). Soit trois utilisateurs A, B et C. Pour cette dernière étape, on utilise :

1. la transitivité (en haut à gauche) où A qui influence B influence aussi C (l'artiste préféré de mon artiste favori a de grandes chances de faire aussi parti de mes artistes favoris) ;
2. les rencontres liées à une influence commune (en haut à droite) où B et C sont influencés par A et ont donc probablement des centres d'intérêts communs, ou bien ;
3. le contraire (en bas) où B est influencé par A et C.

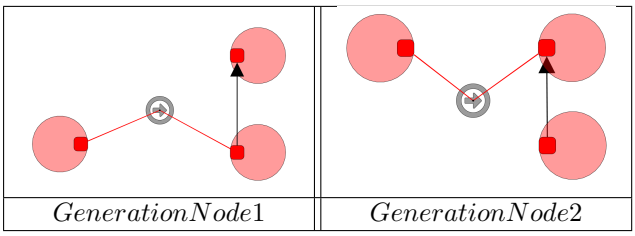


FIGURE 3.8 – Génération de nouveaux sommets dans chaque direction d'arête (invitation d'une nouvelle personne).

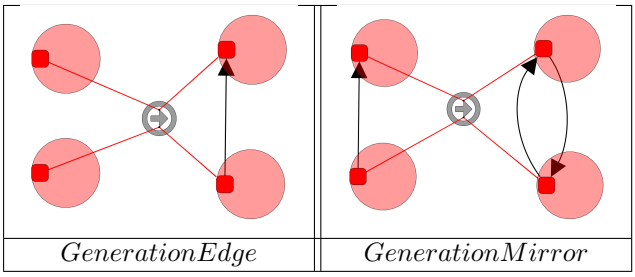


FIGURE 3.9 – Ajout d'arêtes supplémentaires (création de nouveaux contacts).

3.2.2 Définition des stratégies de réécritures

Les trois phases du modèle décrites précédemment sont pilotées par trois stratégies exécutées en suivant. Ces stratégies permettent de générer un graphe avec un nombre paramétré de sommets $|N|$ et d'arêtes $|E|$. Les graphes générés ne possèdent qu'une seule composante connexe ($|E| \geq |N|-1$) et sont simples (une seule arête entre deux sommets) donc le nombre maximum d'arêtes est $|E|_{max} = |E| \times (|N|-1)$.

Tout d'abord, à partir d'un sommet, un graphe acyclique est généré en effectuant un choix équiprobable entre les règles de la figure 3.8 à l'aide de l'opérateur *ppick* (stratégie 3.2). L'utilisation de *one*() permet de n'effectuer qu'une seule application de la règle choisie aléatoirement dans l'ensemble des applications possibles. Chaque règle n'ajoute qu'un seul sommet et une seule arête. Donc, après l'application de cette stratégie, le graphe possède $|N|$ sommets et le nombre minimum d'arêtes, soit $|N|-1$ arêtes.

Stratégie 3.2 : Génération de nouveaux sommets : à partir d'un sommet, création d'un graphe acyclique de N sommets.

```
1 //equiprobabilistic application of the two rules used for generating nodes
2 repeat(
3   ppick(one(GenerationNode1), one(GenerationNode2), {0.5, 0.5})
4 )(|N| − 1) // To eventually get N nodes
```

Ensuite, la stratégie 3.3 densifie le graphe en ajoutant aléatoirement au maximum $|E'|$ arêtes

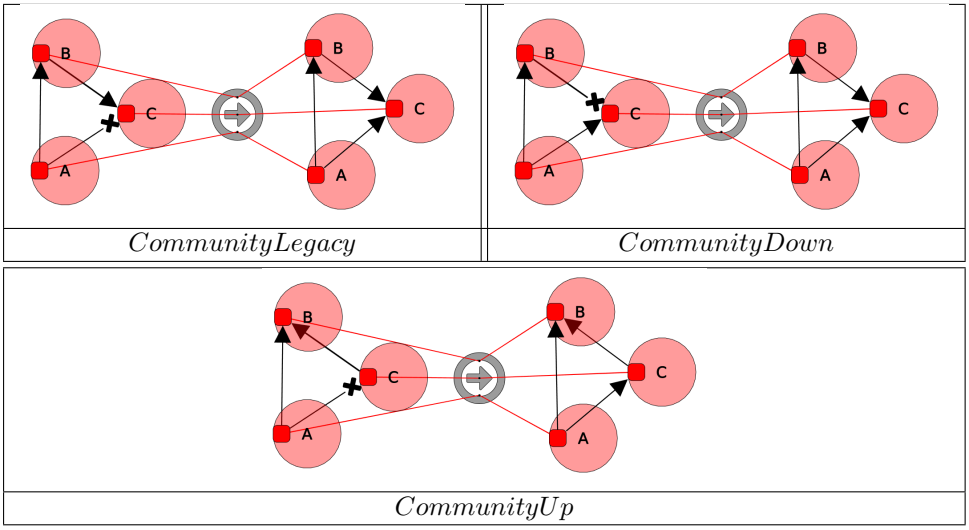


FIGURE 3.10 – Ajout d'arêtes supplémentaires basées sur les interactions dans des triades (création de groupes de personnes autour d'un intérêt commun). Les arêtes avec une croix au lieu d'une flèche à leur extrémité indiquent des connexions qui ne doivent pas exister lors de l'application de la règle (anti-arête).

avec $|E'| < |E| - |N| + 1$ pour que le nombre total d'arêtes reste inférieur à $|E|$. L'opérateur *repeat*() (ligne 1) garantie que si aucune connexion ne peut être ajoutée (aucune image des membres gauches des règles n'est trouvée), la stratégie s'arrête sans erreur. Le graphe généré est donc dans ce cas complet et le nombre d'arête demandé ne pourra pas être atteint. Cette stratégie choisie d'abord aléatoirement un sommet qui peut encore être connecté à un autre sommet qui devient ainsi un voisin supplémentaire (ligne 3). *OutArity* est une propriété maintenue par PORGY qui correspond au degré sortant d'un sommet (*InArity* et *Arity* sont aussi disponibles). On interdit ensuite l'application des règles sur les voisins sortants de ce sommet (ligne 5) pour éviter de créer des arêtes multiples (plusieurs arêtes entre deux ports). L'opérateur *ngbOut*() est une extension de l'opérateur *ngb*() qui permet de filtrer les voisins d'un ensemble de sommets donnés. Comme son nom l'indique *nbgOut*() (resp. *nbgIn*()) filtre les voisins sortants (resp. entrants). Enfin, un choix équiprobable est effectué entre les deux règles (lignes 7 à 9) en testant l'ensemble des combinaisons possibles avec l'opérateur (r_1)`orelse`(r_2) qui exécutera r_2 seulement si r_1 n'est pas applicable. Si une règle est applicable, elle est donc forcément appliquée. Nous avons montré ainsi grâce à la construction du langage de stratégie que cette stratégie ne peut pas s'arrêter prématurément tant qu'il y a au moins une application de règle possible.

Enfin, si la stratégie précédente se termine convenablement ($|E'|$ arêtes ont été ajoutées), il reste encore à ajouter au maximum $|E| - |E'| - |N| + 1$ arêtes avec la stratégie 3.4. Elle effectue un choix équiprobable entre toutes les combinaisons possibles des trois règles de la figure 3.10 si l'application de la règle choisie est possible pour les mêmes raisons que précédemment. Si aucune règle n'est applicable alors la stratégie s'arrête prématurément mais sans erreur.

3.2.3 Validation du modèle

Pour vérifier que les graphes générés grâce aux règles et stratégies présentées précédemment ont bien la propriété petit monde comme souhaité, nous avons reproduit la même démarche de

Stratégie 3.3 : Densification du graphe par l'ajout de $|E'|$ arêtes au maximum.

```
1 repeat(
2 //select one node with an appropriate number of neighbours
3   setPos(one(property(crtGraph, node, OutArity < |N| − 1)));
4 //for this node, forbid rule applications on its outgoing neighbours
5   setBan(all(ngbOut(crtPos, node)));
6 //equiprobable application of the edge generation rules
7   ppick((one(GenerationEdge))orelse(one(GenerationMirror)),
8         (one(GenerationMirror))orelse(one(GenerationEdge)),
9         {0.5, 0.5})
10 )(|E′|)
```

Stratégie 3.4 : Construction des communautés.

```
1 repeat(
2   ppick(
3     (one(CommunityDown))orelse(
4       ppick(
5         (one(CommunityUp))orelse(one(CommunityLegacy)),
6         (one(CommunityLegacy))orelse(one(CommunityUp)),
7         {0.5, 0.5})),
8     (one(CommunityUp))orelse(
9       ppick(
10        (one(CommunityLegacy))orelse(one(CommunityDown)),
11        (one(CommunityDown))orelse(one(CommunityLegacy)),
12        {0.5, 0.5})),
13    (one(CommunityLegacy))orelse(
14      ppick(
15        (one(CommunityDown))orelse(one(CommunityUp)),
16        (one(CommunityUp))orelse(one(CommunityDown)),
17        {0.5, 0.5})),
18        {1/3, 1/3, 1/3})
19 )(|E| − |E′| − |N| + 1)
```

validation que Watts et Strogatz (1998). Sur plusieurs exécutions du système de réécriture avec les mêmes paramètres, on a mesuré la longueur moyenne du plus court chemin entre chaque paire de sommets (*characteristic path length*) qui doit être relativement faible et le coefficient de clustering moyen (densité des connexions du voisinage proche des sommets) qui doit être relativement élevé. Les résultats (détaillés dans Fernandez et al. 2018, section 3.5) montrent que lorsque le nombre d'arêtes ajoutées aléatoirement est faible ($E' << E$), les graphes générés ont bien les caractéristiques d'un graphe petit monde. Quand $E' \sim E$, donc quand les arêtes sont majoritairement créées aléatoirement, on obtient des graphes avec les mêmes caractéristiques que le modèle de génération de graphes aléatoires de Erdös-Rényi à savoir une longueur des chemins courtes et un coefficient de clustering faible.

En conclusion, le système de réécriture décrit dans cette partie permet de générer des graphes avec la propriété petit monde ou selon le paramétrage simplement des graphes aléatoires. Nous avons ensuite utilisé les graphes ainsi générés pour modéliser des algorithmes de propagation et de diffusion d'informations.

3.3 Propagation et diffusion d'informations

Dans un réseau social, une propagation d'information a lieu quand un utilisateur réalise consciemment une action telle que relayer une information ou une rumeur, annoncer un évènement ou bien encore annoncer le partage d'un document. On dit alors qu'il est **actif**. Les voisins de cet utilisateur sont alors **informés** de son état et peuvent ainsi à leur tour devenir actif en relayant l'information. Ce processus est réitéré tant que des nouveaux utilisateurs deviennent actifs et donc partagent l'information avec leurs voisins pour ainsi la propager à l'ensemble du réseau. Plus généralement, chaque individu, représenté par un sommet, est toujours dans un état donné qui sert à déterminer son rôle dans le processus de propagation : non-informé, informé ou actif.

La démarche développée ci-dessous n'est pas celle généralement rencontrée dans la littérature consistant à produire un nouvel algorithme souvent plus performant que ceux disponibles dans la littérature. Notre objectif est de pouvoir comparer entre eux des algorithmes de propagation et diffusion grâce à l'utilisation d'un formalisme de modélisation commun (la réécriture) et ainsi mieux mettre en avant leurs points communs et différences en explicitant les opérations locales effectuées entre un individu et ses voisins.

L'état de l'art est partagé entre deux grandes familles d'algorithmes. D'un côté, les algorithmes probabilistes (Kempe et al., 2003; Chen et al., 2011; Wonyeol et al., 2012) pour lesquels la présence d'un seul sommet actif suffit à enclencher le phénomène de propagation et de l'autre les algorithmes basés sur des seuils qui peuvent évoluer durant la propagation (Watts, 2002; Kempe et al., 2005; Goyal et al., 2010). Les seuils représentent à la fois l'influence qu'exerce un sommet sur ses voisins et la résistance d'un sommet à effectuer une action particulière. Plus un utilisateur recevra de sollicitations à effectuer une action, plus il sera enclin à soit devenir actif et partager l'information soit au contraire à résister le plus possible.

3.3.1 Les modèles à cascades indépendantes et à seuil linéaire

Cette section synthétise la conception de systèmes de réécriture pour les deux modèles de propagation qui servent de base à de très nombreux travaux publiés : le modèle probabiliste à cascades indépendantes (Kempe et al., 2003) (**IC**, *independent cascade model)* et le modèle à seuil linéaire (Goyal et al., 2010) (**LT**, *linear threshold model*). Ces deux modèles sont basés sur le même principe général en deux étapes où les règles de réécriture servent à exprimer comment et quand les sommets peuvent changer d'état. Première étape, un sommet non-informé devient informé quand au moins un voisin actif l'influence pour que, deuxième étape, ce sommet informé

puisse devenir actif quand il a été suffisamment influencé pour à son tour commencer à propager l'information à ses voisins. La première étape est appelée « tentative d'influence » et la seconde « activation ». Du point de vue de l'implémentation, chaque sommet informé possède un attribut τ qui représente le niveau d'influence cumulée. Cet attribut $\tau \in [-1, 1]$ ($\tau = -1$ pour un sommet non-informé, si $\tau \geq 0$ alors le sommet devient actif) est mis à jour au moment de l'application des règles de tentative d'influence.

Dans sa forme la plus simple, la particularité du modèle **IC** est qu'un sommet actif v ne peut tenter d'influencer chacun de ses voisins qu'une seule fois (Kempe et al., 2003). De nombreuses variantes et extensions sont disponibles tels que Gomez-Rodriguez et al. (2012); Watts (2002) ou encore Chen et al. (2011) où les auteurs proposent une extension pour simuler la propagation et l'émergence d'opinions négatives.

Dans le modèle **LT** tel que décrit par Goyal et al. (2010), l'activation d'un sommet n'est plus réalisée en une tentative d'influence comme avec **IC**. Pour devenir actif, un sommet préalablement informé subit l'influence combinée de ses voisins et devient actif à force d'être influencé, généralement en plusieurs tentatives.

Le modèle IC. L'algorithme se déroule en deux étapes répétées tant qu'il reste des activations de sommets possibles. Après avoir défini un ensemble initial de sommets actifs, pour la première étape, chaque sommet v actif a une seule chance d'activer chacun de ses voisins selon une probabilité d'activation p donnée en paramètre de l'algorithme. La deuxième étape consiste simplement à passer dans l'état actif les sommets influencés avec succès. Ces deux étapes sont répétées tant que des sommets sont influençables. Cette description nous permet de définir une liste de propriétés nécessaires à la création des règles de réécriture et de la stratégie à suivre :

IC.1 v n'a qu'une seule possibilité d'activer chacun de ses voisins w ;

IC.2 v active w selon une probabilité de réussite $p_{v,w}$ propre à chaque arête ;

IC.3 Il n'existe pas d'ordre d'activation prédéfini des voisins de v ;

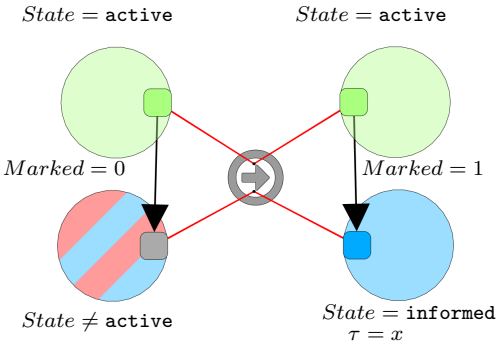
IC.4 Si v active w à l'instant t, w doit être considéré actif à $t + 1$;

IC.5 L'algorithme s'arrête quand plus aucune activation n'est possible.

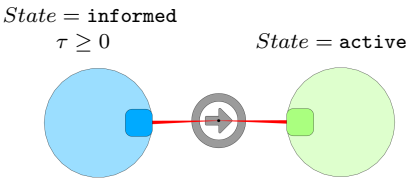
La règle de la figure 3.11a synthétise une tentative d'influence. Un sommet actif (en vert) connecté à un sommet non-informé ou informé (mais non actif) influence ce dernier pour faire évoluer son niveau d'influence global τ (**IC.2**). Si l'influence est satisfaisante ($\tau \geq 0$), la règle de la figure 3.11b peut s'appliquer pour rendre le sommet actif. L'attribut booléen *Marked* des arêtes garantie que chaque voisin d'un sommet actif ne peut être influencé qu'une seule fois (**IC.1**), la règle ne s'appliquant que si l'attribut $Marked$ vaut 0.

La stratégie 3.5 utilise ces règles. À partir des sommets actifs (ligne 1) précédemment initialisés (non détaillé ici), l'application successive des règles est répétée tant que cela est possible (ligne 2) pour chaque sommet du graphe (vérifiant ainsi **IC.3**). L'algorithme s'arrête quand plus aucune tentative d'influence n'est possible (ligne 3, validation de **IC.5**). C'est l'utilisation de l'opérateur *try* (ligne 4) qui permet de garantir cette condition d'arrêt car il n'échoue jamais (voir Fernández et al. (2019) pour plus de détails). La succession des deux règles (lignes 3 et 4) garantie par construction du langage de stratégie la validation de la propriété **IC.4**.

Le modèle LT. La modélisation par la réécriture montre que les modèles **IC** et **LT** sont au final intrinsèquement peu différent. À la différence de **IC**, dans **LT,** à chaque instant t, un sommet u aura de plus en plus tendance à devenir actif plus ses voisins deviennent actifs et ainsi l'influencent. Plus formellement, chaque sommet u possède une fonction d'activation $f_u(S) \to [0, 1]$ avec S l'ensemble des voisins actifs et un seuil θ_u. Si $f_u(S) \geq \theta_u$, alors u devient



(a) ***IC*** *influence trial*. Tentative d'influence d'un sommet non actif. Pour le calcul de τ voir la section 4.1 de Fernandez et al. (2018).



(b) ***IC*** *activate*. Activation d'un sommet informé.

FIGURE 3.11 – Règles pour exprimer le modèle à cascade indépendante **IC**. Les sommets actifs sont représentés en vert, les sommets informés (resp. non-informés) en bleu (resp. en rouge). Un sommet bicolore rouge/bleu peut correspondre à la fois à un sommet informé ou non-informé.

Stratégie 3.5 : Stratégie pour exprimer le modèle à cascade indépendante **IC** avec les règles de la figure 3.11.

```
1 setPos(all(property(crtGraph, node, State == active)));
2 repeat(
3    one(IC influence trial);
4    try(one(IC activate))
5 )
```

actif à son tour. Les propriétés à suivre pour définir les règles de réécriture et la stratégie pour chaque instant t pour un sommet u sont :

LT.1 Le sommet u possède une fonction d'activation $f_u(S)$ utilisée pour calculer l'influence cumulée de ses voisins actifs ;

LT.2 Un sommet u inactif devient actif à $t+1$ si l'influence cumulée de ses voisins ($f_u(S)$) dépasse la valeur du seuil θ_u ;

LT.3 Quand u devient actif, son influence doit alors être considérée sur ses voisins non actifs ;

LT.4 L'algorithme s'arrête quand plus aucune activation n'est possible.

Ces propriétés sont alors exprimées par deux règles de réécriture topologiquement identiques à celles du modèle **IC**. La première modélise toujours une tentative d'influence sur un sommet non actif et la deuxième se charge de rentre un sommet suffisamment informé actif. Seules les propriétés à maintenir sont plus complexes puisque chaque sommet a maintenant un taux d'influence cumulée de ses voisins actifs (validation de **LT.1** et **LT.2**) qui évolue à chaque tentative d'influence. Sauf pour les règles utilisées, la stratégie est identique à celle du modèle IC (stratégie 3.5). La succession des appels des deux règles garantie par construction du langage de stratégie la vérification de **LT.4**. La propriété **LT.3** est vérifiée par la construction du langage de stratégie : dès qu'un sommet devient informé, il est alors automatiquement considéré comme candidat pour devenir le prochain sommet actif.

3.3.2 Hybridation d'algorithmes de propagation et de diffusion d'informations

Pour continuer à montrer la souplesse de notre paradigme de réécriture, nous nous sommes intéressés à une autre classe de modèles que l'on qualifie de modèles de diffusion d'informations, à première vue bien différents des modèles de propagation présentés précédemment. On s'intéresse particulièrement au modèle **RIPOSTE** (**RP**) (Giakkoupis et al., 2015) qui permet de diffuser une information sur le réseau selon que cette information est jugée intéressante ou pas par un nombre suffisant d'utilisateurs. La particularité de **RP** est qu'il protège la vie privée des utilisateurs du réseau. L'opinion de chaque utilisateur ne peut pas être déterminée de façon fiable simplement en observant le processus de dissémination (qui rediffuse quoi). La diffusion de l'information est gérée de manière logicielle, c'est-à-dire sans action concrète des utilisateurs. Ces derniers évaluent l'information reçue et le système diffusera plus facilement et largement une information jugée intéressante surtout si ce dernier a un grand nombre de voisins directs (reliés par une seule arête) non encore informés.

Contrairement à **LT**, l'algorithme de diffusion **RP** n'utilise donc pas l'influence des voisins pour rendre les sommets actifs. Un utilisateur influence la diffusion d'une information selon qu'il adhère ou pas à l'information sans pour autant dévoiler cette opinion à l'ensemble du réseau puisque il n'intervient pas sur le processus de dissémination. En revanche, dans un véritable réseau social, les comportements de **LT** et **RP** sont régulièrement rencontrés ensemble. Un utilisateur peut apprécier une information ou un sujet mais il peut, en plus, être aussi influencé par ses connaissances sur un autre sujet pour lequel il n'est pas familier. De ce constat, nous avons développé un nouveau modèle de dissémination qui rassemble des éléments de **RP** et **LT**. Ce modèle, baptisé **RP-LT**, préserve les opinions personnelles sur les informations diffusées tout en prenant néanmoins en compte l'influence cumulée des voisins d'un sommet pour que ce dernier devienne actif.

Le modèle RP. La modélisation détaillée dans Fernandez et al. (2018) montre clairement que **RP** diffère sur deux aspects par rapport aux modèles présentés précédemment. Premièrement,

l'influence des voisins est remplacée par l'intérêt d'un utilisateur pour l'information. Deuxièmement, les mécanismes de diffusion de l'information et d'activation ne sont plus liés et ne sont pas non plus combinés à l'influence des voisins. Dans **RP**, la diffusion de l'information est fonction d'une probabilité. Un sommet qu'il soit actif ou non peut être utilisé pour transmettre l'information et un sommet actif n'est pas implicitement considéré pour diffuser l'information à ses voisins. De plus et cela de façon aléatoire, l'information est diffusée indépendamment de l'opinion de l'utilisateur sur cette information. **RP** préserve ainsi la vie privée des utilisateurs puisqu'il est très difficile de déterminer ou anticiper qui transmet l'information si les probabilités sont bien choisies. Néanmoins, l'opinion du membre influence sa probabilité de diffuser l'information dans le but de favoriser les informations jugées les plus intéressantes. Contrairement à **LT**, **RP** considère donc d'abord l'intérêt de l'utilisateur pour l'information et non l'influence des utilisateurs les uns sur les autres. **RP** est donc centré sur l'utilisateur et non sur le temps qui passe comme les précédents.

Plus formellement, à partir d'un graphe orienté représentant un réseau social, on suppose qu'un petit nombre d'utilisateurs a une information t à diffuser. Pour chaque utilisateur u informé de t, **RP** va décider dans un premier temps qui va diffuser t à l'ensemble de ses voisins. Le choix des sommets qui diffusent est fait en fonction de l'opinion personnelle et privée de chaque utilisateur sur t et du nombre de voisins s qui ne connaissent pas encore l'information. Si u apprécie t, alors t est diffusée plus facilement vers ses voisins que si t n'est pas appréciée. Plus précisément, soit $0 < \delta < 1 < \lambda$, si u apprécie t, alors t est diffusée par u à ses voisins s_u avec une probabilité λ/s_u. Dans le cas contraire t est diffusée avec une probabilité plus faible δ/s_u. L'algorithme s'arrête après un nombre donné de diffusions ou quand tous les utilisateurs sont informés ou peut aussi continuer indéfiniment. L'attribut τ, qui gérait précédemment le cumul des influences, sert toujours pour l'activation des sommets. Néanmoins, τ est maintenant calculé à partir de l'intérêt d'un utilisateur pour t. Cette description nous permet de définir une liste de propriétés à suivre par les règles et la stratégie pour chaque utilisateur u et l'information à diffuser t :

RP.1 Pour chaque utilisateur u au courant de l'information t à diffuser, soit t est diffusée à l'ensemble des voisins de u soit t n'est pas diffusée du tout ;

RP.2 Si u apprécie l'information t, elle est diffusée à ses voisins avec une probabilité λ/s_u ; si u n'apprécie pas t, l'information peut être quand même diffusée avec une probabilité (bien plus faible) δ/s_u ;

RP.3 L'algorithme s'arrête soit après un certain nombre de diffusions, quand plus aucune diffusion n'est possible ou alors continue indéfiniment.

Ces propriétés sont exprimées par trois règles de réécriture dont les deux premières sont comparables à celles utilisées pour **IC** et **LT**. La règle d'activation d'un sommet informé est toujours présente. Néanmoins, la règle utilisée précédemment pour les tentatives d'influence sert maintenant à la diffusion de l'information. Il faut rajouter une règle simple qui sélectionne un sommet actif ou informé pour lui faire diffuser l'information selon les probabilités (validation de **RP.2**).

La stratégie pour **RP** est composée de deux étapes. La première gère l'application des règles liées à l'activation des sommets à l'image des modèles **IC** et **LT** et la deuxième s'occupe de la diffusion de l'information (validation de **RP.1**). L'algorithme peut à chaque étape choisir un nouveau sommet initial pour diffuser l'information, rendre alors ce sommet actif et essayer d'informer les voisins du sommet choisi si il possède des voisins non informé qui à leur tour pourront diffuser l'information. L'algorithme continue indéfiniment tant qu'il reste des sommets à informer (validation de **RP.3**).

Le modèle hybride RP-LT. Le modèle **RP** se concentre sur l'opinion des utilisateurs pour diffuser rapidement l'information alors que **RP-LT** se concentre d'abord sur l'influence des utilisateurs les uns envers les autres sans pour autant négliger qu'une information jugée particulièrement intéressante peut être diffusée sans attendre. Pour le modèle hybride **RP-LT**, un sommet inactif n' est influencé par chacun de ses voisins actifs n en fonction d'une probabilité $p_{n,n'}$ et on définit $p_{n'}\left(S_{n'}\left(k\right)\right)$ l'influence cumulée sur n' à l'instant k de l'ensemble des voisins de n' noté $S_{n'}(k)$. Le seuil d'influence au delà duquel un sommet devient actif est noté $\theta_{n'}$. Comme dans **LT**, un sommet peut nécessiter d'être influencé plusieurs fois avant de devenir actif. Soit γ le nombre maximum de fois qu'un sommet peut recevoir une information afin qu'il puisse se faire une opinion dessus. Donc, un sommet peut au maximum être influencé γ fois avant de devenir actif à moins que la probabilité de devenir actif ne lui soit favorable avant. Enfin, comme dans **RP**, λ et δ sont des paramètres tels que $0 < \delta < 1 < \lambda$, et $\overline{S_{n'}}$ est l'ensemble des sommets non informés voisins de n'. Cette description nous permet à partir d'un ensemble initial de sommets actifs de définir une liste de propriétés, qui synthétisent celles de **LT** et **RP**, à suivre par les règles et la stratégie :

RP-LT.1 Pour chaque utilisateur n informé de l'information t, **RP-LT** soit diffuse l'information à l'ensemble des voisins de n ou à aucun (propriété issue de **RP**) ;

RP-LT.2 Si n apprécie l'information t, elle est diffusée aux voisins de n avec une probabilité $\lambda/\overline{S_n}$; si n n'apprécie pas l'information, elle peut néanmoins être diffusée avec une probabilité (bien plus faible) $\delta/\overline{S_n}$ (propriété issue de **RP**) ;

RP-LT.3 Un sommet non actif peut être influencé au maximum γ fois et donc a γ chances d'accepter l'information (propriété issue de **LT**) ;

RP-LT.4 Un sommet inactif n possède une fonction d'activation $p_n\left(S_n\left(k\right)\right)$ pour calculer l'influence cumulée de ses voisins actifs (propriété issue de **LT**) ;

RP-LT.5 Un sommet inactif n devient actif si l'influence cumulée de ses voisins dépasse le seuil $p_n\left(S_n\left(k\right)\right) \geq \theta_n$ (propriété issue de **LT**) ;

RP-LT.6 L'algorithme s'arrête quand plus aucune diffusion n'est possible (propriété commune).

Les règles de réécriture reprennent celles de **RP** en rajoutant simplement la gestion des attributs pour le calcul de l'influence cumulée issue de **LT**. La gestion de l'influence cumulée est toujours effectuée grâce à un attribut τ et si $\tau \geq 0$ alors le sommet devient actif. Le sommet devenu actif peut décider à son tour de propager l'information comme dans **LT**. La diffusion ne concerne plus que les sommets non informés qui ont été influencés moins de γ fois. La stratégie pour **RP-LT** est une fois encore équivalente aux modèles précédents. À partir d'un ensemble initial de sommets informés, les règles sont appliquées tant que des sommets peuvent être influencés un nombre suffisant de fois ou tant que l'influence cumulée des voisins d'un sommet peut le rendre actif à son tour. La stratégie commence d'abord par essayer d'activer les sommets qui vont ainsi pouvoir influencer leurs voisins (fonctionnement issu de **LT**). Les sommets non suffisamment influencés peuvent ensuite quand même devenir actifs (fonctionnement issu de **RP**).

3.4 Autres travaux

L'aspect générique de PORGY et de sa méthodologie sous-jacente est renforcé par plusieurs travaux auxquels je collabore notamment avec Maribel Fernández.

Un premier travail porte sur la modélisation de problèmes économiques que sont la gestion des risques dans les crédits et plus globalement la gestion des avoirs des banques (Chinelo Ene et al., 2016, 2017; Ene et al., 2018). L'objectif est de modéliser avec PORGY des modèles théoriques produits par des économistes pour comparer des simulations basées sur différentes valeurs des

paramètres des modèles. Ces modèles nécessitent toute la souplesse du langage de stratégie et de nouvelles extensions du modèle de données de PORGY pour gérer des paramètres globaux au graphe à réécrire et pas seulement aux éléments de ce graphe. Une solution simple déjà rencontrée pour du dessin de graphes (Rodgers, 1998) est d'ajouter un sommet (un graphe dans le cas de Rodgers (1998)) de contrôle dans le graphe à réécrire pour gérer ces paramètres globaux. Les règles doivent dans ce cas prendre en compte ce sommet (ou graphe) de contrôle en plus des autres sommets.

Un second travail porte sur la modélisation logique (modèle relationnel) des bases de données relationnelles (Varga, 2018) [1]. On s'intéresse ici au calcul de la fermeture transitive d'un ensemble de dépendances fonctionnelles (DF) préalable au calcul de couverture minimale et à la normalisation d'un schéma de base de données. À partir d'un ensemble de DF, il s'agit de générer à l'aide des axiomes d'Armstrong l'ensemble des DF non triviales possibles. L'utilisation de la réécriture semble évidente ne serait-ce que par la notation usuelle d'une dépendance fonctionnelle $A \to B$ qui déclare une implication universelle de l'attribut A vers l'attribut B, autrement dit, à chaque valeur de A, on ne peut associer qu'une seule valeur de B (par ex. à un numéro INSEE ne peut correspondre qu'une seule personne). Les axiomes d'Armstrong semblent aussi facile à décrire dans le formalisme de la réécriture, en effet la transitivité entre deux dépendances s'écrit très naturellement $X \to Y, Y \to Z \Longrightarrow X \to Z$.

3.5 Synthèse du chapitre et perspectives

Après la présentation dans le chapitre 2 des travaux qui ont permis d'aboutir à la plateforme PORGY, j'ai présenté et analysé dans ce chapitre différents systèmes de réécriture bâtis grâce à PORGY. Les travaux sur ces systèmes montrent la véracité des avantages présentés dans l'introduction de ce chapitre et plus encore :

— L'avantage 1 (généricité et souplesse) est partiellement validé par la section 3.2. Un système de réécriture conçu pour générer des réseaux sociaux selon différents paramètres inspirés du monde réel est présenté. Cette capacité à générer des modèles arbitraires est fondamentale pour valider de nouvelles méthodes et algorithmes ainsi que vérifier leur comportement. La généricité de l'approche est validée par la présentation d'exemples dans des domaines d'applications différents (biologie, réseaux sociaux, finance, bases de données relationnelles).

— L'avantage 2 (assisse mathématique pour des démonstrations) est validé par les publications associées aux travaux présentés. Les publications présentent toutes des démonstrations de différentes propriétés associées aux modèles notamment la terminaison des stratégies.

— L'avantage 3 (intérêt de la visualisation) est validé par la section 3.1. L'analyse des simulations réalisées sur un système de réécriture est présentée selon les bonnes pratiques issues de la communauté de la visualisation. La visualisation intervient aussi à chaque étape de la création d'un système de réécriture en permettant par exemple de visualiser tout ou partie des règles de réécriture. Les résultats de simulations des modèles de propagation et diffusion d'informations peuvent aussi être visualisés et ainsi comparés entre eux pour fournir des intuitions à une comparaison plus formelle.

— Les avantages 4 (la réécriture en tant que dénominateur commun) et 5 (émergence de briques de base) sont validés par la section 3.3 qui présente la formalisation de modèles de propagation et diffusion d'informations dans un réseau. Les parties communes des différents modèles sont clairement identifiées grâce à l'utilisation de la réécriture pour ainsi mieux

1. Le lecteur non familier du modèle relationnel peut se reporter à : Ullman et Widom (2008)

comprendre comment les modèles diffèrent ou se complètent. Cette formalisation permet de facilement hybrider deux modèles pour en produire un nouveau.

La méthodologie associée à PORGY, le modèle des graphes à ports, le langage de stratégie dédié et l'utilisation de l'outil développé ont été largement publiés et diffusés. Néanmoins, la collaboration initiale avec Inria et King's College n'est pas terminée. Lors d'un séminaire en mai 2019 [2] nous avons identifié trois problèmes dont l'étude nécessite des évolutions majeures de PORGY (méthodologie et/ou implémentation) :

— Analyse des politiques de sécurité dans les services webs (cookies, conséquence de la réglementation liée au RGPD)

— Analyse des politiques d'accès pour l'Internet des objets (IoT) ou le cloud ;

— Analyse d'historiques et de traces (*provenance graph*).

En particulier, le développement de l'Internet des Objets ouvre de nombreuses pistes de recherche. Ces objets connectés qui prennent de plus en plus des décisions par eux mêmes (voitures, maisons, robots aspirateurs, *etc.*) doivent posséder des propriétés dont la sûreté des utilisateurs, sécurité des données utilisées, protection de la vie privée et explicabilité de leur comportement. Cette dernière propriété est importante pour garantir une confiance et une transparence dans le fonctionnement de systèmes qui peuvent mettre en jeu la vie d'êtres humains ou qui manipulent des données sensibles. Cette explicabilité se modélise par des descriptions opérationnelles, causales ou logiques rejoignant ainsi la réécriture comme outil de modélisation. L'objectif est d'arriver à rendre lisible et compréhensible le comportement souhaité des objets et les politiques de sécurité associées. Techniquement, le modèle de règles de réécriture doit être étendu ainsi que l'algorithme de recherche des membres gauches avec par exemple la gestion de plus de conditions, une recherche de membre gauche non exacte, un modèle de règles d'ordre supérieur (un sommet du membre gauche peut remplacer un sous-graphe) ou bien encore un moteur d'inférences dans la stratégie pour par exemple choisir un algorithme de recherche de membre gauche parmi plusieurs.

Au sujet de l'implémentation, il existe un problème de passage à l'échelle. En effet, les réseaux rencontrés dans la littérature ou dans des environnements ouverts (open-data) vont de quelques dizaines d'éléments à plusieurs milliards sachant que les plus gros réseaux sont les plus populaires grâce aux plateformes de réseaux sociaux en ligne. PORGY gère raisonnablement des réseaux de quelques centaines voire quelques milliers d'éléments. Au delà, les temps de calculs deviennent prohibitifs car la taille des réseaux a un impact direct sur les performances de l'algorithme de recherche des membres gauches autrement appelé recherche d'isomorphismes graphe/sous-graphes. Ce problème est étudié depuis longtemps et la littérature est bien fournie (Conte et al., 2004; Lee et al., 2012). Même si l'algorithme de Ullman (1976) que nous utilisons n'est pas le plus efficace (Lee et al., 2012) il reste néanmoins facile à mettre en œuvre (Weber et al., 2012; Cordella et al., 2004). La recherche d'isomorphismes graphe/sous-graphes est de toute façon un problème qui reste étudié (Nabti, 2017). Des évaluations de différents algorithmes d'isomorphisme associées à différents systèmes de réécriture sont à effectuer pour vérifier si dans le contexte de la réécriture, tous les algorithmes se valent. Une idée à développer est qu'à partir d'une taxonomie à créer des formats de règles de réécriture, un algorithme d'isomorphisme puisse être associé à chaque entrée de la taxonomie. Ainsi, lors d'une étape de réécriture, un algorithme pourrait être choisi dynamiquement en fonction du contexte d'exécution de la règle. Le challenge est d'être capable d'analyser les règles pour déterminer l'algorithme en pénalisant le moins possible l'utilisateur de la plateforme de réécriture.

2. Research and Innovation Joint Workshops, King's College London, 21-23 mai 2019, financement principal Ambassade de France au Royaume-Uni et King's College London.

PORGY est une plateforme visuelle et interactive. Le passage à l'échelle ne concerne pas seulement les aspects liés à la réécriture. Il est aussi indispensable d'améliorer les mécanismes de visualisation de PORGY. Le passage à l'échelle nécessite l'ajout de nouvelles méthodes de visualisation de très grands graphes comme l'algorithme JASPER (Vallet et al., 2016), développé durant la thèse de Jason Vallet, pour la visualisation rapide de réseaux sociaux de grande taille.

Chapitre 4

Visualisations « détail vers le contexte global » et réseaux multicouches

Sommaire

Dans les chapitres 2 (page 11) et 3 (page 27), la plateforme Porgy est présentée selon le processus, traditionnel en visualisation, de la mantra de Shneiderman (cf. section 1.1 page 2). Pour l'utilisateur de Porgy ce processus consiste à aller d'une vue globale sur les données représentée par l'arbre de dérivations vers des vues détaillées notamment la visualisation des réseaux associés aux sommets de l'arbre de dérivations et les transitions entre ces sommets. Néanmoins, seulement une partie du processus de réécriture est ici prise en compte. La construction de l'arbre de dérivations n'apparaît pas. En effet, un processus de réécriture commence par l'application d'une règle, c'est-à-dire une modification locale, puis l'expert des données s'intéresse à l'impact de ces modifications locales sur le système complet par l'intermédiaire de l'arbre de dérivations qui est construit au fur et à mesure des applications de règles. Une évolution du processus traditionnel de visualisation est donc à envisager pour prendre en compte la totalité du processus de réécriture qui commence par des modifications locales qui ont un impact global que l'on souhaite étudier sur le système modélisé.

Je me suis ainsi, depuis 2014, intéressé à développer de différentes manières cette autre façon d'aborder la visualisation analytique grâce à des collaborations avec des experts en Sciences Humaines et Sociales (SHS). Par expériences, dont trois font l'objet de ce chapitre, en SHS, un processus de visualisation « détail vers le contexte global » s'avère être une méthode de travail efficace pour collaborer avec les experts. En effet, travailler sur une abstraction des données n'est

bien souvent pas pertinent en SHS ainsi la vue qui montre la totalité des données est souvent difficile voire impossible à utiliser (visualisation trop dense, c'est l'effet « plat de spaghettis » ou «hair-ball », Fig. 4.8b page 64, le réseau en rouge en haut à gauche) et les experts ne savent, la plupart du temps, pas exprimer clairement les questions pour lesquelles la visualisation pourra aider à trouver des réponses (ou au minimum guider vers une réponse). En revanche, les experts connaissent « leurs données » détaillées et finalement souhaitent le plus souvent étudier ces données détaillées dans le contexte global de la totalité des données. Grâce à des collaborations variées et de nombreux encadrements (stagiaires de la L1 au M2 dont Antoine Laumond passé de stagiaire M2 à ingénieur puis doctorant en co-direction avec Guy Melançon), nous avons petit à petit mis au point une approche qui peut être vue comme une inversion de la mantra de Shneiderman : en partant de données détaillées (que l'expert des données maîtrisent le plus souvent au moins en partie), et en y ajoutant des informations de contexte pour aller vers une vue globale qui ait du sens pour l'expert. Dans la communauté visualisation, ce type d'approche semble être seulement apparue depuis environ une décennie (van Ham et Perer, 2009) et reçoit de plus en plus d'attention de la part de la communauté internationale (Luciani et al., 2019).

Les données utilisées, notamment en SHS, sont bien souvent qualifiées de « Small Data » en (mauvaise) opposition aux « Big Data » par le fait que les volumes de données manipulées sont relativement faibles. Néanmoins, sauf bien sûr pour le « V » de volume, les définitions de « V » généralement utilisées pour les « Big Data » fonctionnent parfaitement pour les « Small Data » :

— les **volumes** de données sont le plus souvent raisonnables pour être manipulés par un ordinateur moderne bien équipé contrairement aux Big Data où les données ne tiennent pas dans la mémoire d'un seul ordinateur. Les temps de calculs espérés ne sont donc en rien prohibitifs et ne nécessitent pas des architectures complexes largement utilisées en Big Data (Apache Spark [1] par exemple) ;

— la **variété** est importante : dans la suite de ce chapitre, on trouve des criminels et des activités criminelles décrits par de nombreuses variables, des infrastructures de données géographiques, des articles de journaux, des données de géolocalisation pour ne citer que quelques exemples. Nous verrons dans la suite que les données récoltées sont souvent incomplètes et hétérogènes nous obligeant à revoir et à adapter les méthodes habituelles de fouille et de visualisation qui attendent des formats de données bien précis et des données de qualité ;

— le besoin de **véracité** est permanent. Que l'on travaille sur des données géographiques ou des réseaux criminels (section 4.1) ou des données historiques (section 4.2), il n'est pas possible de déformer la réalité ou de l'approximer. L'expert saura vite si les méthodes de fouilles et visualisations produisent des résultats pertinents ou pas ;

— la **vélocité** des données (aspect dynamique) est présente et le temps qui passe est toujours présent d'une façon ou d'une autre. Les experts sont bien souvent intéressés par étudier ou voir les évolutions temporelles ce qui engendre une complexité accrue des méthodes de visualisation ;

— et évidemment la **visualisation** est toujours un bon outil pour comprendre et expliquer les données ainsi que les phénomènes sous-jacents. Les défis à relever sont nombreux et variés ne serait-ce que pour prendre en compte les points précédents.

Réaliser une implémentation efficace est aussi un challenge. Les modèles et structures de données existants ne permettent pas de capturer facilement et complètement toute la complexité du système à modéliser ainsi que le processus de visualisation « inversé » mis en œuvre par les experts. Un système complexe réel n'est pas correctement modélisé par un seul réseau mais plutôt

1. https://spark.apache.org/

par un ensemble interdépendants de sous-systèmes ou couches. Les réseaux multicouches (Kivelä et al., 2014) sont ainsi une bonne solution pour prendre en compte correctement et complètement toute la complexité d'un système en rendant le modèle plus utile et plus proche de la réalité. En combinant des visualisations et interactions qui partent d'une ou plusieurs données détaillées et les réseaux multicouches, l'expert du domaine reste l'acteur principal du pilotage et de la supervision de son processus de fouille des données.

La chronologie des travaux présentés et les évolutions de notre méthodologie suivent les différents financement obtenus. Tout a commencé par une collaboration avec des géographes et le projet GEOBS (http://geobs.cnrs.fr, financement région Aquitaine, 2015–2018) qui portait sur l'analyse de la circulation d'informations géographiques numériques (section 4.1.1). Ces travaux montrent bien l'émergence d'un réseau multicouche. Puis, nous avons commencé à collaborer avec des juristes et des sociologues sur l'étude de réseaux criminels de traite des êtres humains (2016–2018, section 4.1.2). Après un premier projet relativement court dans le temps où nous avons essayé sans succès les architectures logicielles classiques du web (LAMP [2]), nous avons finalement implémenté un modèle de données plus souple et moins contraint grâce à l'utilisation d'une base de données de graphes. Enfin, un passage à l'échelle est effectué et les réseaux multicouches mis en avant avec la coordination du projet ANR international BLIZAAR (https://blizaar.list.lu). Ce projet effectué avec deux partenaires luxembourgeois (et un autre partenaire français) porte sur la modélisation et visualisation de réseaux multicouches appliqués sur des données dans le domaine des humanités numériques (conservation et exploitation du patrimoine culturel numérique, section 4.2).

4.1 Vers la modélisation et la visualisation de réseaux multicouches

Je synthétise ci-dessous les travaux réalisés en collaboration avec des géographes (section 4.1.1) puis avec des juristes sur des réseaux criminels (section 4.1.2). *A posteriori*, la structure multicouche des données manipulées apparait clairement. Je mets volontairement cette structure en valeur dans ce document d'habilitation car elle est au centre du projet décrit dans la section 4.2 et de mes perspectives de recherche décrites dans le chapitre 5.

4.1.1 Analyse de la circulation d'informations géographiques

Le projet région Aquitaine GEOBS (http://geobs.cnrs.fr) consistait à analyser l'information géographique sur le territoire national qui circule sur le web par l'intermédiaire des « Infrastructure de données géographiques (IDG) » telle que la plateforme PIGMA de la région Aquitaine (devenue Nouvelle-Aquitaine depuis, https://portail.pigma.org/). Pour simplifier, une IDG peut être vue comme un site web qui met à disposition de l'information géographique via un catalogue pour une certaine échelle (nationale, régionale, départementale ou locale). Chaque donnée géographique présente dans une IDG est décrite par un ensemble de méta-données propre à cette IDG. A partir des méta-données récupérées par nos partenaires géographes à différents points dans le temps et d'interviews des responsables des IDG, il s'agissait principalement d'établir un état des lieux et une chronologie de la circulation de l'information géographique au sein d'une IDG (par exemple similarité entre données ou couverture du territoire) et à terme entre différentes IDG (une IDG d'un niveau n récupère souvent les données des IDG au niveau $n-1$).

Le stage de M2 de Antoine Laumond (février à juin 2015) portait sur l'étude des fiches de méta-données de PIGMA (environ 1600 fiches) et la construction de différents réseaux sociaux

2. **L**inux **A**pache **M**ySQL **P**HP, https://en.wikipedia.org/wiki/LAMP_(software_bundle)

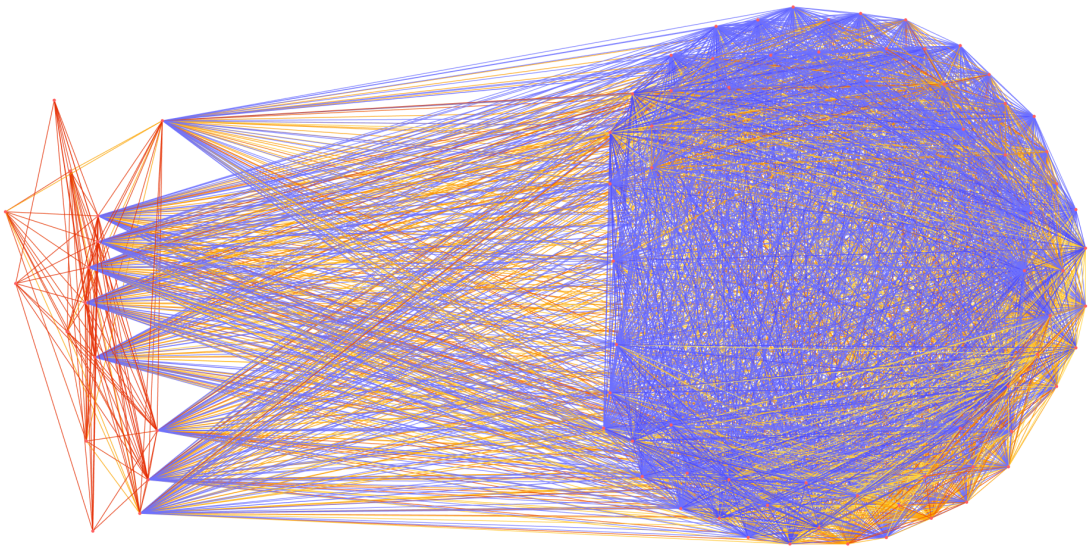


FIGURE 4.1 – Extrait du graphe de superposition des emprises géographiques pour l'IDG PIGMA de la région Aquitaine. Un sommet est une fiche de méta-données et un lien indique une superposition des emprises supérieure à un seuil fixé. La couleur des liens qui varie du bleu au rouge indique le taux de superposition (de faible à important). Cette visualisation montre bien que le territoire Aquitain est bien couvert (peu de recouvrement, la couleur bleue domine) sauf pour quelques fiches (sur la gauche du dessin) qui se recouvrent quasi entièrement.

pour notamment analyser les similarités entre fiches de méta-données (similarité du contenu via une description par mots-clés et des acteurs impliqués), entre fiches et acteurs (retrouve-t-on des groupes d'acteurs spécialisés ?). Un résultat intéressant est la construction du réseau des emprises géographiques (surface du territoire national couvert). À partir des coordonnées géographiques contenues dans les fiches de méta-données, nous avons construit un réseau reliant deux fiches si leurs emprises se superposent au delà d'un seuil préalablement fixé (Fig. 4.1). Ce réseau montre bien que les données contenues dans l'IDG couvre des zones différentes du territoire (peu de superposition) sauf pour quelques sommets positionnés sur la gauche du dessin.

La méthodologie développée dans ces travaux a ensuite été étendue pour être appliquée à 45 IDG françaises afin de comparer les objectifs affichés par leurs promoteurs et leur contenu et services effectifs (Noucher et al., 2016). En plus d'étudier chaque IDG, les relations entre IDG ont aussi été explicitées (interviews des responsables de chaque IDG). L'étude montre que la volonté politique de partage et diffusion de la donnée géographique reste confinée à des thématiques spécifiques avec des acteurs principalement issus du service public. Les réseaux construits sont des réseaux sociaux et nous les avons aussi analysés comme tels (Georis-Creuseveau et al., 2018) notamment pour aider à structurer l'étude des évolutions de l'usage des IDG entre les années 2012 et 2017. Une structuration multicouche apparaît clairement (acteurs, années, portée des IDG) avec des arêtes au sein des couches ou entre les couches (Figure 4.2) mais sur deux réseaux (2012 et 2017). Les acteurs sont pour la plupart dupliqués ce qui est dommage.

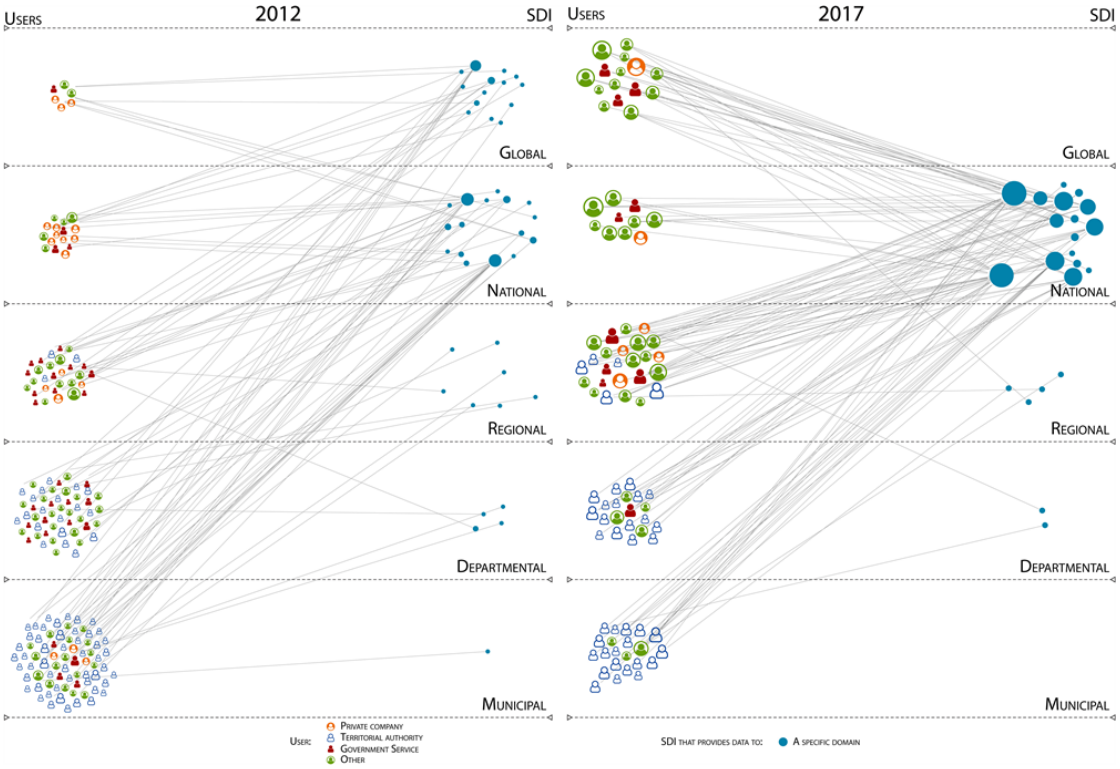


FIGURE 4.2 – Réseaux montrant les évolutions des usages des IDG (SDI en anglais, *Spatial Data Infrastructures*) entre 2012 et 2017 selon leur portée territoriale. Image issue de Georis-Creuseveau et al. 2018. La structure en couches (types d'utilisateurs, échelles d'usage, années) apparaît clairement et aide à apporter des réponses aux questions des experts.

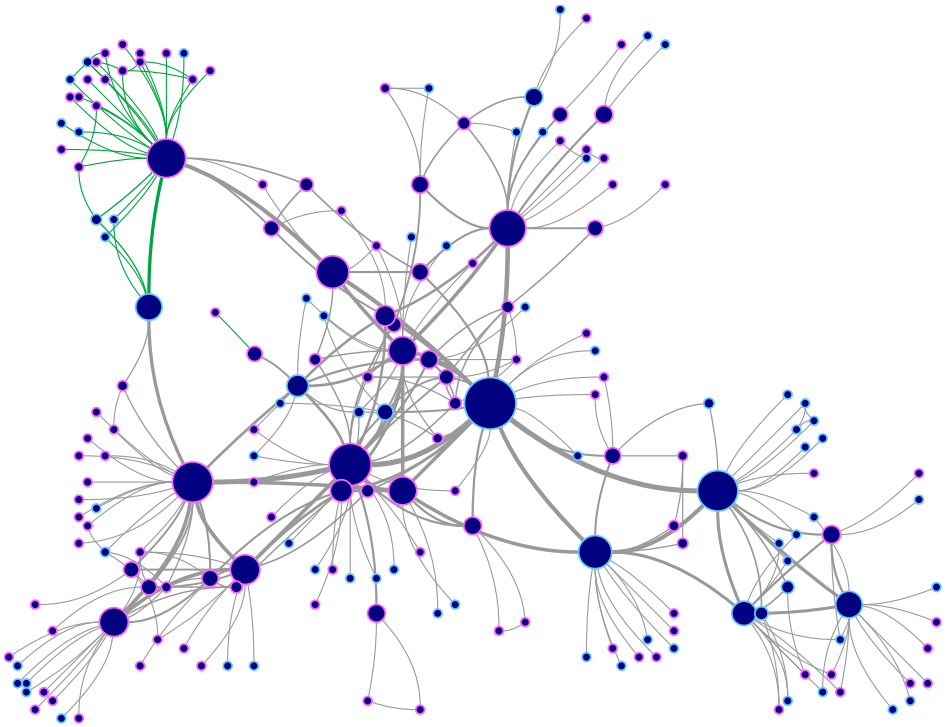
4.1.2 Modélisation et analyse de réseaux criminels

Une autre collaboration importante (débutée au printemps 2015) nous a conforté dans l'idée que les réseaux multicouches sont un outil idéal dans un cadre de collaboration pluri-disciplinaires. Une collègue juriste (Bénédicte Lavaud-Legendre, CNRS, U. Bordeaux) accompagnée d'une autre collègue sociologue en thèse sont venues nous trouver pour étudier la construction et l'analyse de réseaux sociaux impliquant les acteurs (criminels et victimes) d'un dossier judiciaire clôturé (un jugement définitif a été rendu) sur des questions de traite des êtres humains. Le volume de données est tel (dossier papier de 50000 pages) qu'elles ont rapidement conclu que leur approche traditionnelle d'utilisation d'un tableur n'était pas envisageable. L'objectif final, à plus ou moins long terme, de cette collaboration est de mettre au point une approche globale (analyse des acteurs, de leurs attributs, des différents types de liens présents) et un outil informatique pour l'observation des pratiques criminelles afin de mieux comprendre comment fonctionne les réseaux criminels d'exploitation d'êtres humains afin d'accélérer les travaux des policiers et des autorités judiciaires. Plusieurs séminaires ont permis de susciter un intérêt de la communauté de la justice (magistrats, enquêteurs) comme le confirme un article du Figaro paru en avril 2019 [3]. Une synthèse de la méthodologie mise en œuvre est disponible dans Lavaud-Legendre et al. (2017).

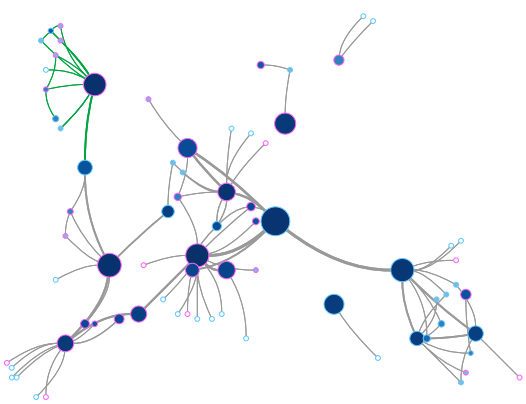
Tout d'abord la nature des données (dossier judiciaire format papier contenant de nombreux types de documents non destinés à une analyse détaillée) ne rend pas envisageable une extraction automatique des informations nécessaires à la construction du réseau. Une lecture du dossier imprimé document par document est donc indispensable. Vu les ressources (humaines, matérielles et temporelles) à notre disposition pour cette gigantesque tâche, nous avons, tout d'abord, mis au point un outil en ligne simple et théoriquement rapide à concevoir d'aide à la saisie. Ce premier outil était basé sur les technologies web classiques ainsi qu'une base de données relationnelles. La nature des données particulièrement les multiples liens possibles entre acteurs, les approximations et données incomplètes inhérentes à ce type de dossier (par exemple, usage de plusieurs pseudonymes pour une même personne, dates et lieux plus ou moins précis) ont rendu ce travail difficile. Nous avons dû effectuer de multiples allers-retours avec les experts et par conséquent de multiples évolutions du modèle relationnel qui par définition n'est pas conçu pour être évolutif (Ullman et Widom, 2008). L'utilisation d'un tel modèle, initialement conçu à partir de la connaissance *a priori* du domaine par les experts, a entraîné d'important coûts (humain et temporel) de développement de l'outil (avec Antoine Laumond en tant que ingénieur). Au final, le modèle relationnel est fort complexe (environ 50 relations) et l'outil développé ne satisfait personne. La multiplicité des relations rend très complexe l'écriture des requêtes SQL (nombreuses jointures) nécessaires à la construction des réseaux à visualiser. Nous avons alors basculé vers l'utilisation d'une base de données de graphes (Neo4J, Robinson et al. 2015) qui apporte au modèle de données la souplesse et la flexibilité qui manquait au modèle relationnel. Cette technologie des bases de données de graphes est au coeur des implémentations des travaux présentés dans la section suivante.

Au final, nous avons pu reconstituer un réseau multicouche à partir des informations extraites du dossier judiciaire et produire des visualisations grâce à Tulip. Une couche est caractérisée par un lien particulier entre les acteurs (par exemple échange d'argent contre un service, lien d'exploitation, lien de fratrie/filiation). Le réseau complet avec l'ensemble des liens confondus est représenté figure 4.3a. Les figures 4.3b et 4.3c montrent les visualisations de deux couches. L'utilisation de Tulip simplifie l'analyse des réseaux. Les positions des sommets restent identiques dans l'ensemble des visualisations, seuls les liens changent. De plus, les sommets et liens sélectionnés en vert (en haut à gauche) dans le réseau complet sont visibles dans les visualisations des

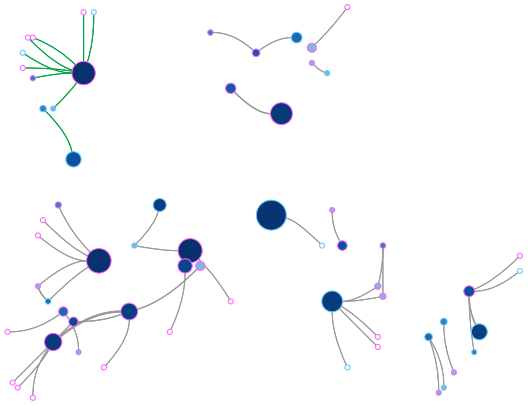
3. http://www.lefigaro.fr/actualite-france/la-technologie-au-secours-de-la-lutte-contre-l-exploitation-sexuelle-20190412



(a) Visualisation du réseau tous types de liens confondus.



(b) Représentation des liens financiers (échange d'argent en échange d'un service).



(c) Représentation des liens de sang (filiation, fratrie).

FIGURE 4.3 – Visualisations des acteurs du réseau pour différentes couches. La position des sommets est fixée. Seuls les liens changent. Certains liens sont sélectionnés en vert dans la figure (a) et apparaissent aussi dans les figures (b) et (c). Extrait de Lavaud-Legendre et al. (2017).

différentes couches. Ces réseaux illustrent bien un scénario de fouille typique s'appuyant sur une cartographie interactive du réseau et des combinaisons de couches. Les dessins rendent compte d'une dynamique des liens et les métaphores visuelles utilisées pour les sommets apportent des informations sur les acteurs. Ces principes ont été repris et étendus dans le projet BLIZAAR présenté ci-dessous.

4.2 Réseaux multicouches et humanités numériques

Les collaborations présentées précédemment ont aiguisé notre intérêt pour les réseaux multicouches et leurs applications. Avec des collègues français et luxembourgeois, nous avons commencé fin 2014 à travailler sur ce qui est devenu le projet bilatéral France-Luxembourg BLIZAAR (ANR PRCI, 01/01/2016–30/06/2019) dont j'étais le coordinateur. Nous avons collaboré avec des historiens spécialistes en humanités numériques (ils préfèrent le terme de « digital history ») et plus précisément sur la préservation et la mise en valeur du patrimoine culturel numérique. Les données utilisées portaient sur l'histoire de la construction de l'Union Européenne à travers le prisme de la couverture médiatique. Nos collègues historiens ont pour cela une plateforme web publique baptisée CVCE (http://www.cvce.eu). Le projet BLIZAAR consiste grossièrement à aller plus loin que cette plateforme en proposant des visualisations, interactions et méthodes de fouilles novatrices du réseau multicouche composé de l'ensemble des données recueillies par les historiens.

Comme pour tout projet de ce type, nous avons effectué un important et long travail d'état de l'art sur la visualisation de réseaux multicouches (sections 4.2.1 à 4.2.3). Un état de l'art se justifie notamment par le fait que dans la communauté internationale en visualisation, les réseaux multicouches sont mal reconnus. Je ne présente pas l'état de l'art en tant que tel (cf. McGee et al. (2019), en libre accès) mais plutôt les arguments qui montrent pourquoi il faut s'intéresser à ces réseaux et les pistes de nouvelles recherches que nous proposons. Ensuite en section 4.2.4, je synthétise les travaux de thèse de Antoine Laumond (co-dirigée avec Guy Melançon, financement obtenu sur le projet BLIZAAR) sur la mise en œuvre d'un processus itératif de fouille et navigation détails vers le contexte global dans un réseau multicouche qui tient compte du degré d'intérêt de l'utilisateur.

4.2.1 Un outil de modélisation

Pour modéliser convenablement un système complexe réel, une bonne approximation [4] est d'utiliser un ensemble inter-dépendant de sous-systèmes ou couches. C'est le domaine des systèmes complexes dans lequel on retrouve de nombreux modèles de réseaux adaptés : réseaux multimodaux ou multivariés, N-parti (biparti par ex.), multiplexes, réseaux de réseaux, réseaux interconnectés, *etc.* Ces modèles, et plus encore, sont unifiés dans le modèle des réseaux multicouches proposé par Kivelä et al. (2014). Dans un réseau multicouche, les éléments fondamentaux sont les sommets et arêtes mais aussi les couches. Les sommets peuvent appartenir à une ou plusieurs couches et les arêtes sont à l'intérieur des couches ou entre les couches. La sémantique associée aux arêtes et sommets peut changer selon les couches considérées. La figure 4.4 est une illustration d'un tel réseau en science de la vie où les biologistes peuvent ainsi mettre en avant le fonctionnement multi-échelles des phénomènes qu'ils étudient. Une autre illustration est la figure 1.4 page 8 sur les données du CVCE. La notion de couches amène des questions nouvelles puisque celles-ci deviennent un artefact mobilisable par l'analyse et la visualisation.

4. Le statisticien George Box a écrit en 1976 : "All models are wrong, some are useful".

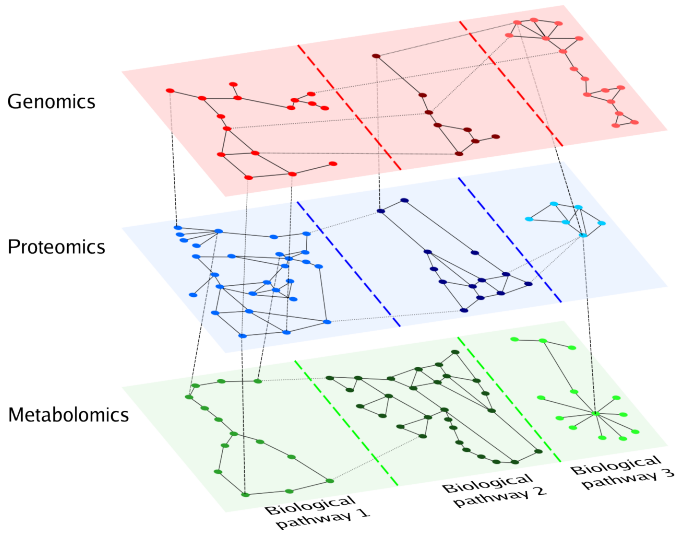


FIGURE 4.4 – Réseau multicouche (purement illustratif) qui modélise l'aspect multiéchelle d'un processus biologique. Les biologistes travaillent le plus souvent avec des réseaux interconnectés à différentes échelles (gênes, protéines et métabolites) qui sont ici les couches. Le fonctionnement de chaque couche est spécifique. Les arêtes au sein des couches ou entre les couches ont des significations différentes.

4.2.2 Des visualisations orientées couches

La visualisation de réseaux multicouches implique particulièrement de prendre en compte la visualisation et l'interaction avec les couches à la place ou en complément des sommets et arêtes. Les états de l'art en visualisation régulièrement publiés (Beck et al., 2017; Hadlak et al., 2015; Vehlow et al., 2017; Nobre et al., 2019) proposent, en lien avec des taxonomies de tâches associées aux sommets et arêtes, des techniques qui sont pour la plupart non dédiées à la visualisation et à l'interaction avec des couches. Néanmoins, les couches sont des éléments fondamentaux d'un réseau multicouche. Les tâches relatives aux couches sont donc à considérer comme des tâches élémentaires et pas des abstractions sur des tâches liées aux sommets et arêtes. Dans notre état de l'art, nous proposons donc une taxonomie de tâches spécifiques aux couches. Cette taxonomie nous a servi pour ensuite faire émerger de la littérature (notamment en dehors de la communauté de la visualisation) les travaux qui justement considèrent les couches comme un élément fondamental qui doit être visualisé et manipulé comme tel. On peut vouloir s'intéresser aux connexions entre les couches, à comparer des entités ou groupes d'entités entre couches, manipuler et reconfigurer les couches, ou bien comparer des couches que ce soit à partir des attributs des éléments qui les composent ou de la topologie des réseaux formés par chaque couche. Les représentations visuelles des couches sont ensuite nombreuses et variées (nœuds-liens en 1D, 2D, 2.5D ou 3D, à base de matrices, d'approches hybrides ou synthétiques) selon que l'on souhaite visualiser et interagir avec la topologie du réseau (voir un exemple figure 4.5) ou la distribution des valeurs de certains attributs (un exemple figure 4.6).

4.2.3 Des pistes de recherches

Dans l'ensemble, les travaux classifiés et décrits dans l'état de l'art ne sont pas spécifiquement mis au point en pensant aux réseaux multicouches. Contrairement à nombre de taxonomies de

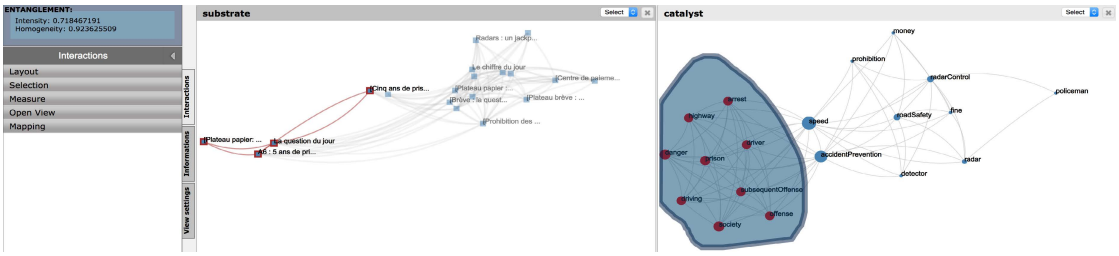


FIGURE 4.5 – Une copie d'écran de Detangler (Renoust et al., 2015) qui montre comment les sommets (à gauche) sont connectés aux couches (à droite). La sélection de couches par l'utilisateur entraîne la sélection des sommets concernés par ces couches (en rouge, à gauche).

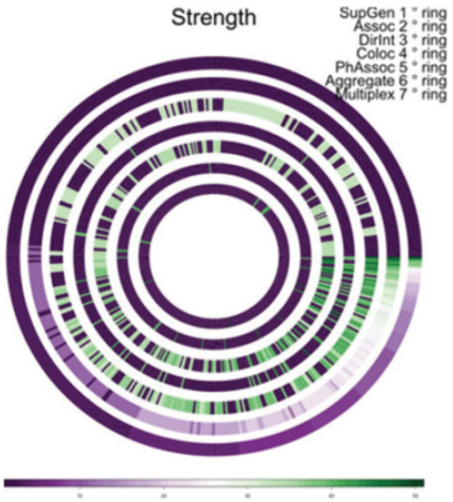


FIGURE 4.6 – Une copie d'écran de MuxViz (De Domenico et al., 2015) qui illustre les valeurs d'un indice de centralité calculé pour chaque couche. Chaque anneau représente une couche. Les sommets sont les mêmes pour chaque couche (réseau multiplexe) ainsi que leur ordonnancement dans chaque couche. Cet ordre est déterminé par la valeur de l'indice de centralité sur l'anneau extérieur. Il est ainsi aisé de comparer les valeurs de l'indice pour chaque couche.

tâches et d'algorithmes ou techniques qui en résultent, les couches sont à considérer comme un élément de premier ordre du réseau. Des travaux sont à mener pour étendre encore les tâches liées aux couches. Plus généralement, en reprenant le processus de visualisation analytique décrit au début de ce manuscrit (figure 1.3 page 5), on peut énumérer quelques pistes de nouvelles recherches dédiées aux couches :

— Pour les parties données et modèles, il est nécessaire d'avoir des algorithmes qui définissent des ensembles de couches à considérer pour l'analyse ou la visualisation comme de nombreux le font déjà pour la définition des sommets et arêtes quand il s'agit de modéliser un système par un graphe. Les couches ou groupes de couches peuvent être naturellement présents dans les données (ex. modélisation des déplacements domicile-travail en prenant en compte l'ensemble des moyens de transport utilisés ; chaque couche est dans ce cas un réseau de transport spécifique) ou générés après analyse (ex. après application d'un algorithme de partitionnement). Les couches sont alors simplement basées sur des ensembles de sommets ou d'arêtes mais il est possible d'être plus « créatif » pour reprendre l'expression de Kivelä et al. (2014).

— Pour la visualisation et les interactions associées, les visualisations hybrides qui combinent plusieurs techniques de visualisations semblent être les plus prometteuses. On peut penser à des représentations optimisées pour mettre en avant la structure entre les couches en complément d'approches plus classiques pour visualiser les éléments d'une couche en particulier. La prise en compte de l'aspect dynamique des données dans les visualisations est aussi un challenge. Le temps peut être modélisé par une ou plusieurs couches mais il est sûrement possible de faire autrement notamment par des interactions novatrices sur l'évolution des arêtes entre les couches et la création/fusion/séparation de couches. La fouille et la navigation dans ces réseaux est aussi un axe à développer. Nous y avons contribué avec les travaux de thèse de Antoine Laumond présentés ci-dessous.

— Enfin, pour la partie connaissance et donc le fait d'apprendre de nouvelles connaissances en utilisant des réseaux multicouches, les modèles, visualisations et interactions ne pourront être jugés efficaces et pertinents qu'après des évaluations rigoureuses classiquement employés dans le domaine (Purchase, 2012). Les méthodes d'évaluations évoluent aussi vers des méthodologies d'évaluations avec un nombre de participants bien plus important qu'à l'accoutumée (Soni et al., 2018). Dans le cas des réseaux multicouches, on pourrait ainsi faire participer de nombreux experts du domaine bien au delà de notre petit cercle de collaborations habituelles (cf. chapitre suivant pour le développement de cette perspective).

4.2.4 Contributions pour la navigation et la fouille

Avec Antoine Laumond, pour sa thèse, nous avons travaillé sur la fouille et la navigation dans le réseau multicouche des données du CVCE. Lors du montage du projet, les discussions avec nos collègues historiens ont montré qu'ils ont souvent besoin de faire des investigations sur le rôle d'un personnage particulier ou d'une organisation. De plus, la recherche est contrainte. Il est nécessaire notamment d'avoir un équilibre dans les types de documents (articles de journaux, dessins de presse, émissions TV, textes officiels) et dans la couverture temporelle (prise en compte de toute la période et pas de trou). La taille de la base est telle que chaque expert historien ne connaît parfaitement qu'une partie des données à disposition. Il faut ainsi une méthode capable de naviguer dans le réseaux des données sans but initial parfaitement défini mais avec quelques contraintes fortes (par ex. couverture temporelle complète et équilibre dans les types de documents).

Nous avons ainsi proposé une extension aux réseaux multicouches des travaux de van Ham et

Perer (2009) sur le calcul du degré d'intérêt des sommets dans un réseau de grande taille (Laumond et al., 2017, 2019). Les challenges sont multiples car :

1. en naviguant, l'expert peut se tromper et suivre une piste qui s'avère sans issue. Il est donc nécessaire de pouvoir revenir dans l'historique des recherches effectuées pour suivre une autre piste ;

2. l'intérêt de l'expert pour les données peut varier selon la couche (ou le groupe de couches) considérée.

Dans ce cadre contraint, les travaux ont abouti à la méthode MQuBE^3 pour fouiller et naviguer dans un réseau multicouche en proposant une série de sous-réseaux issue de l'ensemble des données tout en conservant une trace interactive des actions de l'utilisateur (voir figure 4.7). La trace interactive permet de revenir sur un état précédent pour tester d'autres hypothèses en créant une nouvelle branche. Cette partie de l'interface est une extension des travaux sur l'arbre de dérivations de Porgy réalisée lors d'un stage de M2 (cf. section 2.2.3 page 17).

La méthode MQuBE^3 est composée d'un calcul itératif d'intérêt des sommets (baptisé eScore, bloc B, figure 4.8a) fonction de la ou des couches considérées à partir d'un ensemble de sommets dit « Focus » (bloc A), pour extraire un nouveau sous-réseau (bloc C). Cette méthode est sensée fournir un mécanisme incrémental d'exploration qui doit améliorer la pertinence des sous-réseaux proposés à l'utilisateur itération après itération grâce à l'évolution constante des sommets focus.

Le principe d'extraction d'un sous-réseau d'intérêt $R+1$ à partir d'un sous-réseau R est synthétisé sur la figure 4.8b. Cette méthode est itérative car les sommets « focus » qui sont importants pour l'utilisateur sont la première fois sélectionnés par une recherche de mots-clés classiques et ensuite les sommets focus de R sont reportés dans $R+1$. L'utilisateur est alors libre de modifier cette sélection avant de lancer un nouveau calcul d'intérêt. Le calcul d'intérêt n'est pas le même selon les couches considérées (par ex. l'expert pourrait ne pas vouloir de dessins de presse dans ses résultats ou au contraire les favoriser). Il est basé sur une moyenne pondérée de différentes fonctions qui peuvent être basées sur la topologie d'une ou plusieurs couches (et des arêtes entre les couches) ou sur les propriétés associées aux éléments. Une fois le score calculé, un algorithme glouton se charge de construire le nouveau sous-réseau connecté en prenant les sommets avec les meilleurs scores en priorité.

L'outil a été développé (Fig. 4.7) en constante interaction avec les experts historiens. Pour la fin du projet une session d'évaluation a eu lieu concernant l'usage de l'outil à partir d'un paramétrage effectué par nos soins. Les experts ont apprécié la navigation à travers des vues partielles ainsi que la gestion de l'historique. Notamment la gestion et l'interaction avec l'historique correspond parfaitement à leur façon traditionnelle de fouiller leur base de données. Néanmoins, il ressort une certaine frustration sur le côté boite noire de l'outil et de ne pas comprendre pourquoi certains éléments sont favorisés aux détriments des autres. Le contexte autour des sommets sélectionnés pourrait être plus détaillé notamment en affichant les éléments proches non choisis. Néanmoins, il faut veiller à ne pas trop surcharger visuellement le graphe affiché à l'écran et ne pas trop alourdir l'effort cognitif important que nécessite l'usage d'un tel outil.

En bref, la méthode développée de navigation par succession de vues partielles successives basées sur une mesure d'intérêt des sommets fonction des couches considérées répond aux attentes des experts. Les commentaires positifs ou négatifs des experts montrent qu'ils ont parfaitement compris nos travaux. Le point faible comme dans de nombreux travaux en visualisation est l'aspect boite noire et de comment avoir confiance dans la visualisation. Je reviens sur ce dernier point dans mes perspectives de recherche détaillées dans le prochain chapitre.

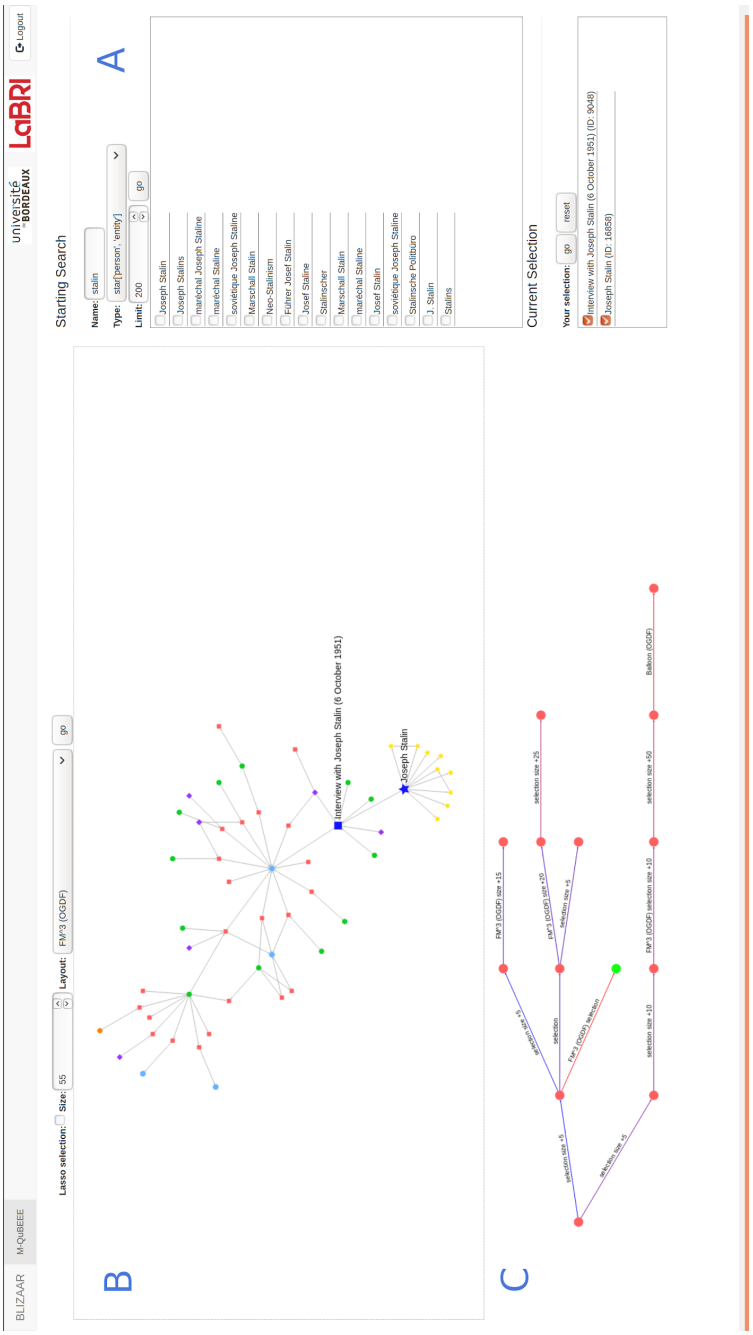
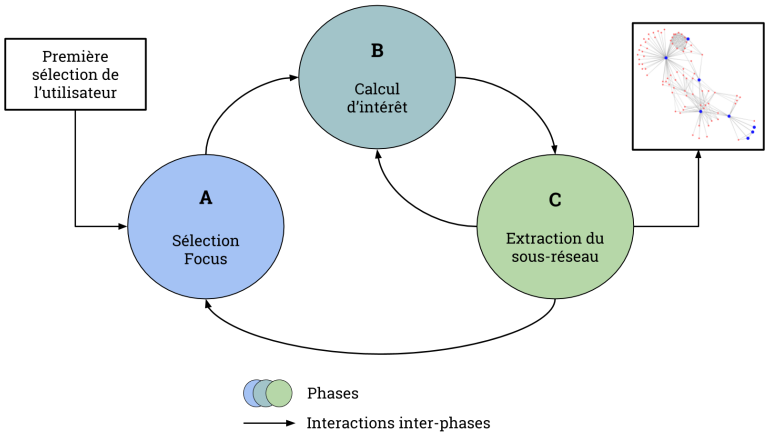
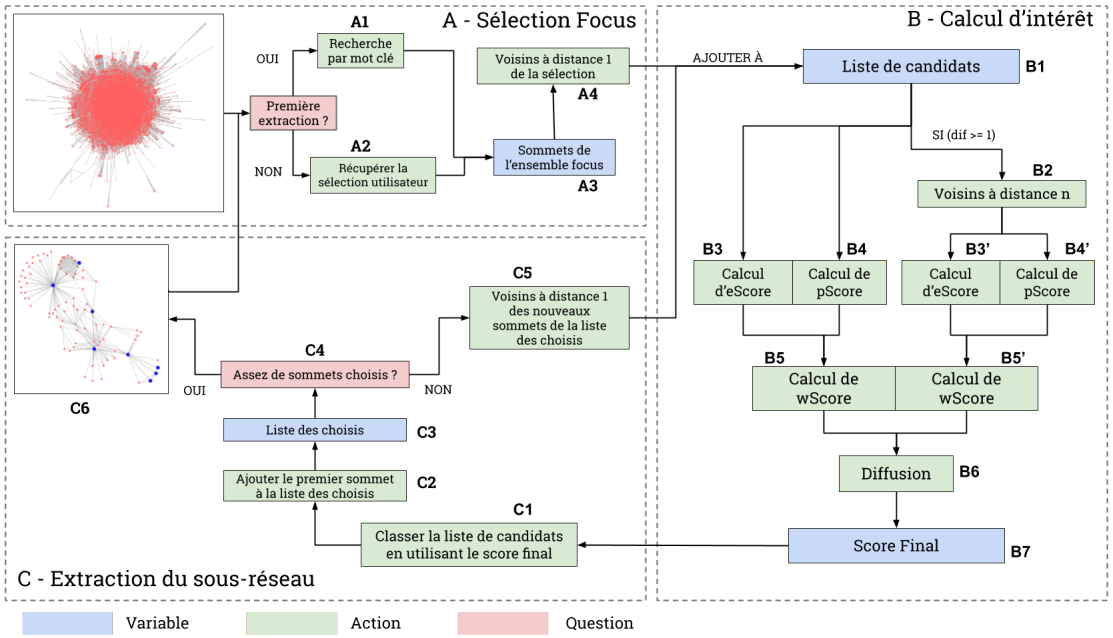


FIGURE 4.7 – Implémentation de MQuBE[3] dans une solution web. Après une recherche classique par mots clés (A), un sous réseau extrait de l'ensemble des données est produit en tenant compte de l'intérêt de l'utilisateur selon les couches. L'utilisateur peut ensuite sélectionner ou enlever de la sélection des nouveaux sommets selon son intérêt (B). Un nouveau sous-graphe est ainsi recalculé. L'historique de l'ensemble des sous-réseaux extraits est accessible en permanence (C). Il suffit de cliquer sur un sommet de l'historique pour visualiser le sous-graphe associé et l'utiliser pour reprendre l'exploration à partir de ce point.



(a) Synthèse de la méthode MQuBE^3.



(b) Détail du fonctionnement des trois blocs.

FIGURE 4.8 – **Méthode MQuBE^3 :** description du fonctionnement de l'extraction de sous-réseaux d'intérêt et principe du calcul d'un sous-réseau à partir des mesures d'intérêts appliquées aux couches (eScore). A partir de l'ensemble des données (le graphe rouge) et des résultats d'une recherche part mots-clés, un premier sous-réseau d'intérêt est extrait de l'ensemble des données après calculs et combinaisons des scores sur les couches concernées. Sur ce sous-réseau de taille réduite, l'utilisateur peut faire évoluer sa sélection et un nouveau sous-réseau d'intérêt est calculé selon le même principe. Ce processus est itéré jusqu'à satisfaction de l'utilisateur. Voir Laumond et al. (2019) pour plus de détails.

4.3 Synthèse du chapitre

J'ai synthétisé dans ce chapitre les travaux sur trois projets pluridisciplinaires en collaboration avec des experts de données en SHS (géographes, juristes, sociologues, historiens). Ces projets sont tous différents notamment par leur envergure ou leur financement : participation à un projet régional avec les géographes, participation à un projet émergent avec les juristes et sociologues, montage et coordination d'un projet ANR de collaboration internationale (PRCI) avec les historiens. Ces trois projets montrent chacun, mais différemment, que la méthode de travail des experts est globalement identique quelque soit l'application. En SHS, cette méthode de travail se résume par une inversion des pratiques courantes de la communauté de la visualisation que je synthétise rapidement par une approche « détails vers contexte global ». La chronologie de ces projets montrent aussi l'évolution de nos méthodes et outils ainsi que l'intérêt d'utiliser le modèle des réseaux multicouches tels que formalisé par Kivelä et al. (2014). Ces réseaux sont une approche transversale en étant à l'intersection des domaines d'applications et des données. Ils sont un bon outil de modélisation et d'échange entre les experts du domaine et les experts en visualisation. Pour les experts du domaine, notamment en SHS, la modélisation en couches est naturelle et évidente. Les discussions pour la mise au point de visualisation interactive s'en trouvent alors simplifiées. Le développement de l'usage des réseaux multicouches notamment en visualisation à été renforcé dans la communauté visualisation grâce aux résultats de notre projet ANR PRCI bilatéral notamment la publication de l'état de l'art et sa présentation orale à la communauté internationale [5] ainsi que l'organisation d'un séminaire Dagstuhl (séminaire 19061, https://www.dagstuhl.de/19061) sur les réseaux multicouches, leurs visualisations et utilisations dans de nombreux domaines d'applications en présence de nombreux experts.

Ces travaux ouvrent des perspectives de recherches à plus ou moins longs termes. De nombreux challenges sur les réseaux multicouches restent ouverts notamment pour leur analyse avec le calcul de mesures comme par exemple l'étude de la distribution des arêtes entre les couches, le suivi de communautés dans des données dynamiques, ou bien encore la détection de motifs (cf. chapitre suivant). De plus, je reste ouvert à toute nouvelle application. Pour cela, je vais continuer à essayer de tisser des liens avec des collègues des autres disciplines même si les SHS ont une place prépondérante de part la diversité des données et le besoin de diffusion vers la société (notamment pour les réseaux criminels qui sont régulièrement évoqués dans les médias). D'un point de vue technique, les différentes collaborations évoquées et l'objectif permanent de rendre facilement utilisable nos outils de visualisation et d'analyse aux experts des données (au moins les chercheurs) est loin d'être atteint. Il reste un problème de confiance à résoudre dans les outils mis à disposition des experts ainsi que dans les visualisations (cf. chapitre suivant).

5. Conférence Eurovis 2019. Il est d'usage que les articles publiés directement dans les journaux majeurs de la communauté soient présentés oralement lors des conférences internationales majeures du domaine.

Chapitre 5

Conclusion et perspectives

Sommaire

Dans ce manuscrit, j'ai présenté une synthèse des travaux les plus marquants effectués depuis ma nomination MCF à l'université de Bordeaux en septembre 2008. Après avoir positionné mes travaux dans une introduction (chapitre 1), j'ai présenté dans les deux chapitres suivants, les travaux sur lesquels j'ai passé le plus de temps. Premièrement, la modélisation et le développement de la plateforme Porgy (chapitre 2), ensuite, les principales applications de Porgy (chapitre 3) qui permettent de montrer différents usages de la plateforme et de sa méthodologie associée de modélisation à base de règles. Enfin, j'ai synthétisé les travaux plus récents qui ont conduit à développer mon intérêt pour les réseaux multicouches appliqués sur des données et problématiques issues du monde des SHS (chapitre 4). Chaque chapitre comporte une mise en contexte, une synthèse des travaux qui s'appuie sur les publications et des perspectives directes.

Je ne reviens donc pas dans cette conclusion sur ces perspectives. Je vais plutôt prendre un peu de recul. J'esquisse quelques perspectives orientées vers le développement des réseaux multicouches et de leurs usages. La première sur la combinaison possible de la modélisation à base de règles et des réseaux multicouches (section 5.1) notamment grâce à une collaboration naissante avec des experts suisses en sciences criminelles (École des Sciences Criminelles, Université de Lausanne). La deuxième sur comment faire en sorte que les experts des domaines d'applications aient plus confiance dans les visualisations que nous produisons et donc des pistes d'évolutions des pratiques d'évaluation de nos travaux afin de mesurer cette confiance (section 5.2).

5.1 Combiner modélisation à base de règles et réseaux multicouches

La synthèse des travaux sur Porgy présentée dans ce manuscrit (section 3.5 page 47) fait états de plusieurs perspectives dont plusieurs extensions du modèle de réécriture qui nécessitent d'importantes évolutions de la méthodologie associée et de la plateforme Porgy. Un autre type d'évolution est d'envisager l'utilisation de la modélisation à base de règles, donc de la recherche

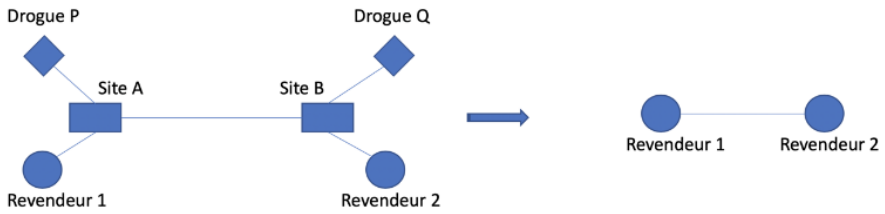


FIGURE 5.1 – Exemple purement illustratif de transformation de graphe pour reconstruire un réseau de personnes.

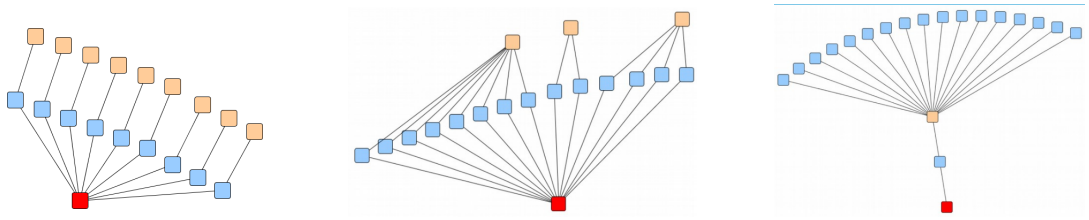
de motifs, sur des réseaux multicouches. Je décrits ci-dessous deux exemples rencontrés dans le cadre d'une nouvelle collaboration avec l'École des Sciences Criminelles de Lausanne et des experts en science forensique [1].

Nos premiers travaux avec les collègues de Lausanne et les travaux déjà effectués sur les réseaux criminels (section 4.1.2 page 56) montrent dans les deux cas qu'une problématique des experts est de détecter dans les données des tendances, des schémas répétitifs ou bien faire apparaître des liens peu évidents aux premiers abords. Ces recherches de motifs correspondent à des inférences spécifiques de la forensique numérique développées par l'école des sciences criminelles. Elles reposent sur la détection de traces numériques signes d'activités criminelles particulières. La détection de similarités entre les traces vise, au travers d'un raisonnement par analogie, à inférer des causes communes : mêmes criminels, mêmes groupes d'auteurs, mêmes modes opératoires, *etc.* Elles impliquent des opérations de transformations et de simplifications des graphes qu'il faut pouvoir caractériser – afin de les embarquer dans des systèmes interactifs de visualisation analytique tel que PORGY. Par exemple, dans le cadre de la reconstruction d'un trafic de biens illicites sur Internet, les réseaux à reconstruire ne contiennent au départ aucun lien direct évident entre les revendeurs. La détection de relations entre des acteurs sociaux implique de détecter des traces de leurs activités sur des espaces de ventes en ligne, puis d'inférer les relations entre personnes (Fig. 5.1).

Ainsi, la modélisation de réseaux pour répondre à différentes questions d'analyse implique des transformations appliquées aux graphes complexes initialement reconstruits. Typiquement, un lien entre revendeurs est indirectement reconstruit par un motif relationnel indirect observé dans le graphe initial. La complexité des problèmes à reconstruire implique également de pouvoir les modéliser et les représenter selon de multiples perspectives en parallèle tout en pouvant naviguer entre les niveaux d'abstraction et de généralisation. Un motif analogue à celui décrit plus haut permet de dériver un réseau liant des produits illicites. Ces deux réseaux, de personnes ou de produits, se trouvent alors implicitement liés : étant donné un groupe de personnes, le sous-graphe des produits liés doit pouvoir être sélectionné via le motif de gauche, puis modélisé dans un réseau ad hoc de produits dont le modèle peut lui-même reposer sur une transformation dérivée du graphe initial. Ce fonctionnement peut être vu comme une généralisation des travaux de Renoust et al. (2015) sur Detangler.

Par hypothèse de nos collègues suisses, la détection de motifs spécifiques dans des graphes peut également permettre de caractériser des activités particulières. Par exemple, nos collègues effectuent régulièrement des relevés de traces numériques sans lien apparent sur des sites de l'internet clandestin (*dark web*) qui ont des activités illicites pour essayer d'identifier les serveurs et/ou leurs administrateurs. Ces traces semblent révéler des structures spécifiques lorsqu'elles sont liées entre elles (Figure 5.2).

1. http://criminologie.com/article/science-forensique



(a) Hébergement de différents sites originaux (ou non rattachés à d'autres traces)

(b) Sites miroirs originaux.

(c) Duplication de contenu (un même site est accessible depuis plusieurs adresses).

FIGURE 5.2 – Motifs spécifiques à détecter dans un réseau de traces relevées sur le darkweb pour identifier les serveurs, leurs administrateurs ou tout lien non explicite. Les sommets rouges sont des serveurs, les bleus sont des adresses (URL) et les autres sommets représentent les sites web.

Ces manipulations, cruciales pour l'expert, et qui peuvent paraître simples (d'où leur intérêt du point de vue de la visualisation analytique), ne le sont pas lorsqu'elles sont étudiées en toute généralité et sur des cas réels. À titre d'exemple, les motifs de la figure 5.2 sont à détecter dans un graphe composé des traces de plus de 26000 sites (les sommets jaunes). Il est ainsi nécessaire d'apporter des réponses méthodologiques et algorithmiques qui permettent d'embarquer ce type de manipulations dans un système interactif qui doit pouvoir répondre dans des temps très courts et sur des graphes de bonne taille (plusieurs dizaines de milliers de sommets et d'arêtes). Plus généralement, un tel système générique doit répondre à trois questions principales :

1. Quels sont types de transformations typiques et récurrentes dans les processus d'analyse ?
2. Comment identifier une/des famille/s de motifs et formaliser des processus de détection fondés sur des transformations de graphes (plutôt que de passer par la recherche exhaustive des motifs) ?
3. Comment opérationnaliser efficacement la synchronisation de vues dérivées d'un ou plusieurs motifs ?

5.2 Améliorer la confiance des experts dans les visualisations

En tant que chercheur, nous ambitionnons à ce que nos travaux soient repris pour être cités et à terme transférés hors des murs de nos communautés scientifiques. En visualisation, nous travaillons avant tout pour les experts des domaines d'applications (cf. la citation en exergue du chapitre 1). Nous souhaitons donc que les experts avec lesquels nous collaborons finissent par adopter les outils et visualisations développés pour eux et *in fine* faire évoluer leurs pratiques. Pour cela, les experts doivent avoir confiance dans les visualisations, comprendre les interactions proposées et être certains que leur données soient bien représentées et non déformées. Néanmoins, mon expérience personnelle, dont une partie est décrite dans ce manuscrit, est qu'obtenir la confiance des experts est un véritable challenge le plus souvent sous-estimé.

Ce fait est partagé par de nombreux chercheurs rencontrés ici et là (notamment lors de mes passages à Dagstuhl). En effet, les experts ne sont pas forcément familiers du domaine de la visualisation et souvent les visualisations produites peuvent être déroutantes pour eux car loin de leur pratique quotidienne. Par exemple, sur une visualisation de graphe nœuds-liens, la faible distance (visuellement parlant) entre deux sommets ne signifie la plupart du temps

rien alors que les experts voudraient, le plus souvent, y voir une proximité sémantique entre les entités représentées par les sommets. Cette proximité n'est en fait qu'une conséquence de l'algorithme de dessin utilisé. Plus généralement, depuis presque 20 ans et le développement de la presse numérique (donc de l'interactivité), on a gagné en popularité sur des visualisations que la communauté propose. On ne voyait pas de graphes dans la presse des années 2000. Maintenant, la plupart des quotidiens ont régulièrement recours à des visualisations (treemaps, nœuds-liens, mais aussi, visualisations géolocalisées, orientée pixels, etc.). En collaboration avec des scientifiques, le New York Times propose même des visualisations en ligne à comprendre et à discuter (https://www.nytimes.com/column/whats-going-on-in-this-graph).

Cette acculturation doit maintenant être suivie d'une bonne éducation du lecteur qui doit prendre conscience des biais que les visualisations contiennent nécessairement. Le public est aujourd'hui plus soucieux de comprendre les conclusions statistiques des sondages (cf. les mentions maintenant en notes de bas de page sous-jacentes aux conclusions sur les échantillons, et l'interprétation des chiffres donnés) sans pour autant devenir expert en statistiques.

En pratique, les utilisateurs voient le plus souvent la production d'une visualisation comme une boite noire. Les données connues et maîtrisées de l'utilisateur sont utilisées pour produire la visualisation qui apparaît un peu comme par magie, et ensuite, il est même possible d'interagir avec. La communauté de la visualisation commence à se préocuper de ce problème de confiance. L'appel à communications de l'atelier « TrustVis 2019 » organisé sous ce nom pour la première fois en marge de la conférence Eurovis 2019 (https://trustvis.org/) donne une bonne définition de la confiance en une visualisation. Je reprends ci-dessous les trois principaux points qui sont donnés du point de vue de l'expert :

— la visualisation montre les nombres que je connais de mon tableur (outil largement répandu) ;

— je comprends comment mes données sont visualisées et comment interpréter tout ça (perception et interactions avec la visualisation) ;

— les résultats issus de la visualisation peuvent être vérifiés ou validés par d'autres moyens.

Pour améliorer la confiance des experts dans nos visualisations, il faut évidemment faire preuve de pédagogie et de méthodologie. Cela passe par plusieurs canaux, perceptifs et articulatoires (au sens de Bertin) et aussi sémantiques. Dans cette optique, les réseaux multicouches semblent être un bon candidat pour cela (section 5.2.1). Mais, il faut aussi être capable de mesurer régulièrement cette confiance afin d'en évaluer les évolutions positives ou négatives. Cette mesure peut passer par des évaluations simples, régulières et pas forcément très précises des visualisations contrairement aux pratiques traditionnelles de la communauté visualisation (section 5.2.2).

5.2.1 Les réseaux multicouches comme un outil pédagogique ?

Dans notre état de l'art sur la visualisation de réseaux multicouches (McGee et al., 2019) nous avons montré que ce modèle généralise de nombreux autres modèles (voir aussi dans ce manuscrit la section 4.2 page 58). Mon expérience montre que les experts des domaines d'applications n'ont aucun mal à utiliser un réseau multicouche surtout si la notion de couche est rattachée à la sémantique des données. Dans le modèle imbriqué de Munzner (Munzner, 2009) qui permet de structurer et valider les développements de visualisations (utilisé pour le développement de Porgy, voir le début du chapitre 2 et pour structurer le projet BLIZAAR), les couches permettent à la fois d'exprimer les questions des experts et elles sont aussi utilisées en tant que structure de données de haut-niveau validant ainsi les deux premiers niveaux du modèle. L'expert peut alors comprendre comment sont manipulées ses données, augmentant ainsi sa confiance dans nos travaux. Il est donc nécessaire de promouvoir les réseaux multicouches qui

restent mal connus et compris. Les résultats du projet BLIZAAR (notamment la publication de l'état de l'art) sont un premier pas. Il reste de nombreuses opportunités de recherche autour de la visualisation et l'analyse de réseaux multicouches tels que la création, manipulation et comparaison de couches de manière interactive, l'analyse des distributions des arêtes entre les couches ou encore l'adaptation de nombreux algorithmes de partitionnement, de calcul de communautés et de dessin.

5.2.2 Simplifier les processus d'évaluations

En visualisation, il est normal d'évaluer nos travaux avec des utilisateurs pour vérifier que les visualisations répondent bien aux problèmes qu'elles sont sensées résoudre. Néanmoins, la méthode traditionnelle d'évaluation repose sur un protocole lourd et complexe comme pour les évaluations sur lesquelles j'ai contribué (Archambault et al., 2010a,b, 2011; Sansen et al., 2015). Des évaluations de ce type sont difficiles à mettre en œuvre, chronophages et à réaliser dans un environnement de laboratoire contrôlé sur un nombre restreint de participants (souvent une vingtaine d'étudiants de passage ou de collègues du laboratoire). En effet, il faut s'assurer de la qualité et de la fiabilité des résultats, de leur reproductibilité et donc d'éviter un maximum de biais pour conserver une évaluation juste et précise entre les méthodes évaluées. Pour cela, il est d'usage courant d'abstraire les tâches à résoudre vers les tâches issues des taxonomies de tâches références dans la communauté (Lee et al., 2006; Ahn et al., 2013) comme l'estimation du degré des sommets, l'évaluation de la longueur des chemins, la reconnaissance de groupes de sommets fortement connectés, *etc.* Néanmoins, cette façon de faire ne fonctionne pas avec les réseaux multicouches car les taxonomies précédemment citées ne prennent pas en compte les tâches spécifiques aux réseaux multicouches telles que la comparaison et la manipulation de couches (McGee et al., 2019). De plus, réseaux multicouches ou non, les résultats d'évaluation sont le plus souvent difficiles à interpréter et à généraliser malgré l'utilisation de tests statistiques principalement à cause du faible nombre de participants et de leur provenance souvent unique. On comprend ainsi aisément pourquoi on ne fait pas plus souvent ce genre de travaux qui, pourtant, sont utiles et nécessaires. Dans de nombreuses publications, les auteurs font à la place des validations minimales mal documentées et non reproductibles pour s'assurer que les experts qui ont participé au développement du projet sont satisfaits. La communauté visualisation s'est emparée de ce problème des évaluations depuis de nombreuses années attesté depuis 2006 par l'atelier bi-annuel BELIV (`https://beliv-workshop.github.io/`) dont l'acronyme vient d'évoluer pour la dernière édition en 2018 de « Beyond Time And Errors: Novel Evaluation Methods For Visualization » vers « evaluation and BEyond - methodoLogIcal approaches for Visualization ». Cet élargissement du spectre de l'atelier montre bien que les pratiques en évaluation doivent évoluer.

Une proposition liée à une évolution récente en visualisation est de recourir à des évaluations plus intéressantes pour les participants, plus simples, plus rapides, et à plus large échelle en utilisant du *crowdsourcing*. Sur ce sujet, l'état de l'art de Borgo et al. (2018) montre parfaitement cette évolution ainsi que les risques inhérents à cette pratique notamment sur le recrutement des participants et la qualité des réponses obtenues. Les auteurs de l'état de l'art valident cette nouvelle approche en citant notamment des travaux qui reprennent en mode crowdsourcing des évaluations menées de façon traditionnelle tout en retrouvant des résultats comparables. Les auteurs font aussi états de défis à relever pour simplifier et étendre l'utilisation de ce mode d'évaluation en visualisation. Je m'inscrit dans ces perspectives notamment le développement de plateformes d'évaluation dédiées pour la visualisation et ouvertes à la communauté avec des fonctionnalités de crowdsourcing. Un telle plateforme doit permettre d'atteindre plus de participants tout en élargissement leur profil car les experts des domaines d'applications ne

sont le plus souvent ni informaticien ni mathématicien. Mais il faut aussi avoir une plateforme qui motive à répondre convenablement aux questions posées et pourquoi pas une dimension ludique. Ces caractéristiques sont celles des jeux sérieux (*serious games*) qui sont bien plus intéressants à utiliser (mais autrement plus difficile à concevoir) que simplement comparer deux images entre elles et ensuite mesurer la qualité des réponses (vraie/fausse) et le temps de réponse. Les jeux sérieux sont maintenant bien répandus notamment dans le domaine de la formation. Des chercheurs et ingénieurs du LaBRI spécialisés dans le son et le multimedia ont ainsi conçu la plateforme de jeux sérieux SEGMent[2] [2] dans cette optique. Cette plateforme sert notamment pour le jeu sérieux « Subpœna » de l'université de Bordeaux conçu pour sensibiliser les étudiants sur le plagiat [3]. Nous avons ainsi à disposition les méthodes, outils et compétences pour envisager le développement d'une plateforme pour le web de type crowdsourcing qui soit de plus ouverte et accessible pour produire facilement des jeux sérieux configurables et adaptables pour l'évaluation de visualisations et plus généralement l'évaluation d'un processus de visualisation analytique à destination du grand public d'une part mais aussi des experts de domaines d'applications d'autre part.

2. https://scrime.u-bordeaux.fr/Arts-Sciences/Projets/Projets/SEGMent2-Study-and-Education-Game-Maker

3. https://www.u-bordeaux.fr/Actualites/De-la-formation/Un-serious-game-pour-sensibiliser-les-etudiants-au-plagiat

Chapitre 6

Bibliographie

Sommaire

6.1 Publications depuis ma nomination MCF

La liste complète de mes publications est disponible à http://www.labri.fr/perso/bpinaud/?Publications. Cette page est générée automatiquement à partir des archives ouvertes HAL (https://hal.archives-ouvertes.fr/). Elle contient un lien vers une version « auteur » pour chaque article.

b Journaux internationaux avec comité de lecture (8)

Fernández, M., H. Kirchner, et B. Pinaud (2019). Strategic Port Graph Rewriting : an Interactive Modelling Framework. *Mathematical Structures in Computer Science 29*(5), 615–662, doi:10.1017/S0960129518000270.

McGee, F., M. Ghoniem, G. Melançon, B. Otjacques, et B. Pinaud (2019). The state of the art in multi-layer network visualization. *Computer Graphics Forum 38*(6), 125–149, doi:10.1111/cgf.13610.

Marai, G. E., B. Pinaud, K. Bühler, A. Lex, et J. H. Morris (2019). 10 simple rules to create biological network figures for communication. *PLoS Computational Biology 15*(9), doi:10.1371/journal.pcbi.1007244.

Fernandez, M., H. Kirchner, B. Pinaud, et J. Vallet (2018). Labelled Graph Strategic Rewriting for Social Networks. *Journal of Logical and Algebraic Methods in Programming 96*(C), 12–40, doi:10.1016/j.jlamp.2017.12.005.

Georis-Creuseveau, J., C. Claramunt, F. Gourmelon, B. Pinaud, et L. David (2018). A Diachronic Perspective on the Use of French Spatial Data Infrastructures. *Journal of Geographic Information System 10*(04), 344–361, doi:10.4236/jgis.2018.104018.

Pinaud, B., G. Melançon, et J. Dubois (2012). PORGY : A Visual Graph Rewriting Environment for Complex Systems. *Computer Graphics Forum 31*(3), 1265–1274, doi:10.1111/j.1467-8659.2012.03119.x.

Archambault, D., H. Purchase, et B. Pinaud (2011). Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs. *IEEE Transactions on Visualization and Computer Graphics 17*(4), 539–552, doi:10.1109/TVCG.2010.78.

Archambault, D., H. Purchase, et B. Pinaud (2010a). The readability of Path-Preserving Clusterings of Graphs. In G. Melançon, T. Munzner, et D. Weiskopf (Eds.), *Eurovis 2010, 12th annual Eurographics/IEEE Symposium on Visualization*, Volume 29(3) of *Eurographics/ IEEE-VGTC Symposium on Visualization 2010*, Bordeaux, France, pp. 1173–1182. WILEY, doi:10.1111/j.1467-8659.2009.01683.x.

Conférences internationales avec comité de lecture (6)

Varga, J., M. Fernandez, et B. Pinaud (2019). A port graph rewriting approach to relational database modelling. In *29th International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR)*.

Laumond, A., G. Melançon, B. Pinaud, et M. Ghoniem (2019). M-QuBE 3 : Querying Big Multilayer Graph by Evolutive Extraction and Exploration. *Journal of Imaging Science and Technology*, doi:10.2352/ISSN.2470-1173.2019.1.VDA-686.

Fernández, M., H. Kirchner, et B. Pinaud (2018). Labelled Port Graph – A Formal Structure for Models and Computations. *Electronic Notes in Theoretical Computer Science 338*, 3 – 21, doi:10.1016/j.entcs.2018.10.002.

Vallet, J., G. Melançon, et B. Pinaud (2016). JASPER : Just A new Space-filling and Pixel-oriented layout for large graph ovERview. In *Conference on Visualization and Data Analysis (VDA 2016)*, Electronic Imaging, pp. 1–10. doi:10.2352/ISSN.2470-1173.2016.1.VDA-484.

Sansen, J., R. Bourqui, B. Pinaud, et H. Purchase (2015). Edge Visual Encodings in Matrix-Based Diagrams. In 19th *International Conference on Information Visualisation (IV)*, pp. 62–67. doi:10.1109/iV.2015.22.

Fernandez, M., H. Kirchner, I. Mackie, et B. Pinaud (2014). Visual Modelling of Complex Systems : Towards an Abstract Machine for PORGY. In A. Beckmann, E. Csuhaj-Varjú, et K. Meer (Eds.), *Computability In Europe (CIE 2014) : Language, Life, Limits*, Volume 8493 of *Lecture Notes in Computer Science*, pp. 183–193. Springer International Publishing, doi:10.1007/978-3-319-08019-2_19.

Archambault, D., H. Purchase, et B. Pinaud (2010b). Difference Map Readability for Dynamic Graphs. In U. Brandes et S. Cornelsen (Eds.), *18th International Symposium on Graph Drawing*, Volume 6502 of *LNCS*, pp. 50–61. Springer, doi:10.1007/978-3-642-18469-7_5.

Chapitres de livres (3)

Andrei, O., M. Fernández, H. Kirchner, et B. Pinaud (2019). Strategy-Driven Exploration for Rule-Based Models of Biochemical Systems with Porgy. In B. Hlavacek (Ed.), *Modeling Biomolecular Site Dynamics*, Volume 1945 of *Methods in Molecular Biology*, pp. 43–70. Springer, doi:10.1007/978-1-4939-9102-0_3.

Auber, D., D. Archambault, R. Bourqui, M. Delest, J. Dubois, A. Lambert, P. Mary, M. Mathiaut, G. Mélançon, B. Pinaud, B. Renoust, et J. Vallet (2017). TULIP 5. In R. Alhajj et J. Rokne (Eds.), *Encyclopedia of Social Network Analysis and Mining*, pp. 1–28. Springer New-York, doi:10.1007/978-1-4614-7163-9_315-1.

Auber, D., D. Archambault, R. Bourqui, M. Delest, J. Dubois, B. Pinaud, A. Lambert, P. Mary, M. Mathiaut, et G. Melancon (2014). Tulip III. In *Encyclopedia of Social Network Analysis and Mining*. Springer New-York, doi:10.1007/978-1-4614-6170-8_315.

Manifestations (workshops) internationales avec comité de lecture (6)

Ene, N., M. Fernández, et B. Pinaud (2018). A Graph Transformation Approach to the Modelling of Capital Markets. In *GMC 2018 – 9th International Workshop on Graph Computation Model*, pp. 204 – 207.

Chinelo Ene, N., M. Fernández, et B. Pinaud (2017). Attributed Hierarchical Port Graphs and Applications. In *4th International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE 2017)*, Volume 265 of *Electronic Proceedings in Theoretical Computer Science*, pp. 2–19. doi:10.4204/EPTCS.265.2.

Fernández, M., H. Kirchner, B. Pinaud, et J. Vallet (2016). Labelled Graph Rewriting Meets Social Networks. In D. Lucanu (Ed.), *Rewriting Logic and Its Applications (WRLA)*, Volume 9942 of *LNCS*, pp. 1–25. Springer International Publishing Switzerland, doi:10.1007/978-3-319-44802-2_1.

Vallet, J., H. Kirchner, B. Pinaud, et G. Melançon (2015). A Visual Analytics Approach to Compare Propagation Models in Social Networks. In A. Rensink et E. Zambon (Eds.), *Graphs as Models*, Volume 181. doi:10.4204/EPTCS.181.5. arXiv :1504.02448.

Fernandez, M., H. Kirchner, et B. Pinaud (2014). Strategic Port Graph Rewriting : An Interactive Modelling and Analysis Framework. In D. Bošnački, S. Edelkamp, A. L. Lafuente, et A. Wijs (Eds.), *3rd Workshop on GRAPH Inspection and Traversal Engineering (GRAPHITE)*, Volume 159 of *Electronic Proceedings in Theoretical Computer Science*, pp. 15–29. doi:10.4204/EPTCS.159.3.

Andrei, O., M. Fernandez, H. Kirchner, G. Melançon, O. Namet, et B. Pinaud (2011). PORGY : Strategy-Driven Interactive Transformation of Graphs. In R. Echahed (Ed.), *6th International Workshop on Computing with Terms and Graphs (TERMGRAPH 2011)*, Volume 48 of *Electronic Proceedings in Theoretical Computer Science (EPTCS)*, pp. 54–68. doi:10.4204/EPTCS.48.7.

Journaux nationaux avec comité de lecture (1)

Lavaud-Legendre, B., C. Plessard, A. Laumond, G. Melançon, et B. Pinaud (2017). Analyse de réseaux criminels de traite des êtres humains : méthodologie, modélisation et visualisation. *Journal of Interdisciplinary Methodologies and Issues in Science volume Graphes et systèmes sociaux*, doi:10.18713/JIMIS-300617-2-5.

Conférences nationales avec comité de lecture (2)

Noucher, M., F. Gourmelon, A. Laumond, G. Melançon, B. Pinaud, A. Maulpoix, J. Pierson, O. Pissoat, et M. Rouan (2016). Un cadre d'analyse des Infrastructures de Données Géographiques pour interroger la mise en réseaux des acteurs et des outils. In *SAGEO : Spatial Analysis & Geomatic*, Nice, France.

Vallet, J., B. Pinaud, et G. Melançon (2015). Une approche de visualisation analytique pour comparer les modèles de propagation dans les réseaux sociaux. In J. Darmont, B. Otjacques, et T. Tamisier (Eds.), *Extraction et Gestion de Connaissances (EGC 2015)*, Volume RNTI-E-28, pp. 365–376. Prix du meilleur article académique (http://www.egc.asso.fr/Manifestations_dEGC/5-FR-Prix_EGC).

Co-direction d'ouvrages (7)

Pinaud, B., F. Guillet, F. Gandon, et C. Largeron (2019). *Advances in Knowledge Discovery and Management, Vol. 8*, Volume 834 of *Studies in Computational Intelligence*. Springer International Publishing, doi:10.1007/978-3-030-18129-1.

Pinaud, B., F. Guillet, B. Crémilleux, et C. De Runz (2018). *Advances in Knowledge Discovery and Management, Vol. 7*, Volume 732 of *Studies in Computational Intelligence book series (SCI)*. Springer, Cham, doi:10.1007/978-3-319-65406-5.

Guillet, F., B. Pinaud, et G. Venturini (2017). *Advances in Knowledge Discovery and Management, Volume 6*, Volume 665 of *Studies in Computational Intelligence*. Springer International Publishing, doi:10.1007/978-3-319-45763-5.

Guillet, F., B. Pinaud, G. Venturini, et D. A. Zighed (2016). *Advances in Knowledge Discovery and Management*, Volume 615 of *Studies in Computational Intelligence*. Springer International Publishing, doi:10.1007/978-3-319-23751-0.

Guillet, F., B. Pinaud, G. Venturini, et D. A. Zighed (2014). *Advances in Knowledge Discovery and Management Volume 4*, Volume 527 of *Studies in Computational Intelligence*. Springer, doi:10.1007/978-3-319-02999-3.

Guillet, F., B. Pinaud, G. Venturini, et D. A. Zighed (2013). *Advances in Knowledge Discovery and Management. Studies in Computational Intelligence - volume 3 - Vol. 471.* Springer, doi:10.1007/978-3-642-35855-5.

Lechevallier, Y., G. Melançon, et B. Pinaud (2012). *Extraction et Gestion des Connaissances, EGC2012*, Volume RNTI-E-23 of *Revue des Nouvelles Technologies de l'Information*. Hermann.

Posters en conférences internationales avec comité de lecture (4)

Laumond, A., G. Melançon, et B. Pinaud (2017). eDOI : Exploratory Degree of Interest Exploration of Multilayer Networks Based on User Interest. In *VIS 2017, Poster session*.

Vallet, J., B. Pinaud, et G. Melançon (2014). Studying propagation dynamics in networks through rule-based modeling. Visual Analytics Science and Technology (IEEE VAST). Poster.

Pinaud, B., J. Dubois, et G. Melançon (2011). PORGY : Interactive and Visual Reasoning with Graph Rewriting Systems. Conf. on Visual Analytics Science and Technology (VAST), 2011 IEEE (Poster Abstract). doi:10.1109/VAST.2011.6102480, Poster.

Pinaud, B. et P. Kuntz (2010). GVSR : an On-Line Guide for Choosing a Graph Visualization Software. In U. Brandes et S. Cornelsen (Eds.), *18th International Symposium on Graph Drawing*, Volume 6502 of *LNCS*, pp. 400–401. Springer, doi:10.1007/978-3-642-18469-7_41.

Autres publications (11)

Pinaud, B., O. Andrei, M. Fernández, H. Kirchner, G. Melançon, et J. Vallet (2017). PORGY : a Visual Analytics Platform for System Modelling and Analysis Based on Graph Rewriting. In *Extraction et Gestion de Connaissances*, Volume RNTI-E-33 of *Extraction et Gestion de Connaissances 2017 (EGC2017)*, pp. 473–476. Revue des Nouvelles Technologies de l'Information . Démonstration logicielle.

Pinaud, B., O. Andrei, M. Fernández, H. Kirchner, G. Melançon, et J. Vallet (2017b). PORGY : a Visual Analytics Platform for System Modelling and Analysis Based on Graph Rewriting. Atelier Visualisation d'informations, interaction et fouille de données, Conférence EGC2017.

Laumond, A., G. Melançon, et B. Pinaud (2017). Exploration visuelle de graphes multi-couches basée sur un degré d'intérêt. In *Atelier Visualisation d'informations, interaction et fouille de données, Conférence EGC2017.*

Laumond, A., B. Pinaud, et G. Melançon (2016). Réseaux multiplexes à travers les sciences sociales : une adaptation et des règles dictées par les données. In *7ème conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques (MARAMI).*

Vallet, J., G. Melançon, et B. Pinaud (2016a). JASPER : Visualisation orientée pixel de grands graphes. In *Ateliers Visualisation d'informations, Interaction, et Fouille de données (VIF 2016).*

Chinelo Ene, N., M. Fernández, et B. Pinaud (2016). Graph Models for Capital Markets. In *23rd International Workshop on Algebraic Development Techniques.* Conference based on a 2-page abstract. (http ://cs.swan.ac.uk/wadt16/submission.html).

Vallet, J., B. Pinaud, et G. Melançon (2016). A "small-world" graph generative model mimicking social networks. In 7^e *conférence sur les Modèles et l'Analyse des Réseaux : Approches Mathématiques et Informatiques (MARAMI).*

Vallet, J., B. Pinaud, G. Melançon, et H. Kirchner (2014). Propagation Dynamics in Social Networks Through Rule-Based Modeling. In *1st European Conference on Social Network (EUSN).*

Melançon, G., M. Noirhomme-Fraiture, et B. Pinaud (2013). Évaluation des interfaces visuelles. In *Atelier Visualisation d'informations, interaction et fouille de données - Conférence Extraction et Gestion des Connaissances 2013 (EGC'2013).*

Auber, D., P. Mary, M. Mathiaut, J. Dubois, A. Lambert, D. Archambault, R. Bourqui, B. Pinaud, M. Delest, et G. Melançon (2010). Tulip : a Scalable Graph Visualization Framework. In S. B. Yahia et J.-M. Petit (Eds.), *Extraction et Gestion des Connaissances (EGC) 2010*, Volume RNTI E-19 of *Extraction et Gestion des connaissances EGC'2010*, pp. 623–624. RNTI.

Pinaud, B. et G. Melançon (2010). PORGY : réécriture et visualisation de graphes dynamiques. In *Extraction et Gestion des Connaissances (EGC 2010), 8e Atelier Visualisation et Extraction de Connaissances.*

Rapports de recherche (3)

Fernández, M., H. Kirchner, B. Pinaud, et J. Vallet (2017). Porgy Strategy Language : User Manual. Research report, Université de Bordeaux, LaBRI ; Inria Bordeaux Sud-Ouest ; King's College London.

Georis-Creuseveau, J., F. Gourmelon, G. Le Campion, A. Maulpoix, M. Noucher, B. Pinaud, et O. Pissoat (2017). Enquête 2017 auprès des usagers des Infrastructures de Données Géographiques en France. Rapport intermédiaire du projet de recherche GEOBS. Research report, LETG - Brest Géomer ; Passages UMR 5319 ; LaBRI.

Auber, D., R. Bourqui, M. Delest, A. Lambert, P. Mary, G. Melançon, B. Pinaud, B. Renoust, et J. Vallet (2016). TULIP 4. Research report, LaBRI - Laboratoire Bordelais de Recherche en Informatique.

6.2 Autres références citées

Ahn, J.-w., C. Plaisant, et B. Shneiderman (2013). A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics 99*(PP), 365–376, doi:10.1109/TVCG.2013.238.

Andrei, O. (2008). *A Rewriting Calculus for Graphs : Applications to Biology and Autonomous Systems*. Ph. D. thesis, Institut National Polytechnique de Lorraine.

Arendt, T., E. Biermann, S. Jurack, C. Krause, et G. Taentzer (2010). Henshin : Advanced concepts and tools for in-place emf model transformations. In D. C. Petriu, N. Rouquette, et Ø. Haugen (Eds.), *Model Driven Engineering Languages and Systems*, Berlin, Heidelberg, pp. 121–135. Springer Berlin Heidelberg, doi:10.1007/978-3-642-16145-2_9.

Auber, D. (2004). *Tulip — A Huge Graph Visualization Framework*, pp. 105–126. Berlin, Heidelberg : Springer Berlin Heidelberg, doi:10.1007/978-3-642-18638-7_5.

Beck, F., M. Burch, S. Diehl, et D. Weiskopf (2017). A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum 36*(1), 133–159, doi:10.1111/cgf.12791.

Bertin, J. (1967). *Sémiologie graphique. Les diagrammes, les réseaux, les cartes*. Les réimpressions des Editions de l'École des Hautes Études en Sciences Sociales. 4^e édition (2005).

Borgatti, S. P., A. Mehra, D. J. Brass, et G. Labianca (2009). Network analysis in the social sciences. *Science 323*(5916), 892–895, doi:10.1126/science.1165821.

Borgo, R., L. Micallef, B. Bach, F. McGee, et B. Lee (2018). Information visualization evaluation using crowdsourcing. *Computer Graphics Forum 37*(3), 573–595, doi:10.1111/cgf.13444.

Boutillier, P., M. Maasha, X. Li, H. F. Medina-Abarca, J. Krivine, J. Feret, I. Cristescu, A. G. Forbes, et W. Fontana (2018). The kappa platform for rule-based modeling. *Bioinformatics 34*(13), i583–i592, doi:10.1093/bioinformatics/bty272.

Brandes, U., M. Eiglsperger, I. Herman, M. Himsolt, et M. S. Marshall (2002). Graphml progress report : Structural layer proposal. In *Proc. of the* 9^{th} *Int. Symp. on Graph Drawing (GD '01)*, Volume 2265 of *LNCS*, pp. 501–512. Springer-Verlag, doi:10.1.1.4.4374.

Brandes, U. et T. Erlebach (2015). *Network Analysis – Methodological Foundations*. Springer-Verlag Berlin Heidelberg, doi:10.1007/b106453.

Brandes, U. et D. Wagner (2004). Analysis and visualization of social networks. In M. Jünger et P. Mutzel (Eds.), *Graph Drawing Software*, Mathematics and Visualization, pp. 321–340. Springer Berlin Heidelberg, doi:10.1007/978-3-642-18638-7_15.

Cao, L. (2017). Data science : A comprehensive overview. *ACM Comput. Surv. 50*(3), 43 :1–43 :42, doi:10.1145/3076253.

Card, S. K., J. D. Mackinlay, et B. Shneiderman (Eds.) (1999). *Readings in Information Visualization : Using Vision to Think*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

Chen, W., A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincón, X. Sun, Y. Wang, W. Wei, et Y. Yuan (2011). Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pp. 379–390. doi:10.1137/1.9781611972818.33.

Conte, D., P. Foggia, M. Vento, et C. Sansone (2004). Thirty Years Of Graph Matching In Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence 18*(3), 265–298, doi:10.1142/S0218001404003228.

Cordella, L. P., P. Foggia, C. Sansone, et M. Vento (2004). A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*(10), 1367–1372, doi:10.1109/TPAMI.2004.75.

Courcelle, B. (1990). Graph Rewriting : An Algebraic and Logic Approach. In J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science, Volume B : Formal Models and Semantics*, pp. 193–242. Elsevier and MIT Press.

Danos, V., J. Feret, W. Fontana, R. Harmer, J. Hayman, J. Krivine, C. Thompson-Walsh, et G. Winskel (2012). Graphs, Rewriting and Pathway Reconstruction for Rule-Based Models. In S. D. L.-Z. fuer Informatik (Ed.), *FSTTCS 2012 - IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Volume 18 of *LIPIcs*, Hyderabad, India, pp. 276–288. doi:10.4230/LIPIcs.FSTTCS.2012.276.

Danos, V., J. Feret, W. Fontana, R. Harmer, et J. Krivine (2007). Rule-based modelling of cellular signalling. In L. Caires et V. Vasconcelos (Eds.), *CONCUR 2007 - Concurrency Theory*, Volume 4703 of *Lecture Notes in Computer Science*, pp. 17–41. Springer Berlin Heidelberg, doi:10.1007/978-3-540-74407-8_3.

De Domenico, M., M. A. Porter, et A. Arenas (2015). Muxviz : a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks 3*(2), 159–176, doi:10.1093/comnet/cnu038.

Eberle, W. et L. Holder (2007). Anomaly detection in data represented as graphs. *Intelligent Data Analysis 11*(6), 663–689.

Faeder, J., M. Blinov, et W. Hlavacek (2009). Rule-based modeling of biochemical systems with bionetgen. In I. V. Maly (Ed.), *Systems Biology*, Volume 500 of *Methods in Molecular Biology*, pp. 113–167. Humana Press, doi:10.1007/978-1-59745-525-1_5.

Fernández, M., H. Kirchner, et O. Namet (2012). A strategy language for graph rewriting. In G. Vidal (Ed.), *Logic-Based Program Synthesis and Transformation*, Volume 7225 of *Lecture Notes in Computer Science*, pp. 173–188. Springer Berlin Heidelberg, doi:10.1007/978-3-642-32211-2_12.

Fernández, M. et O. Namet (2010). Strategic programming on graph rewriting systems. In *Proceedings International Workshop on Strategies in Rewriting, Proving, and Programming, IWS 2010*, Volume 44 of *Electronic Proceedings in Theoretical Computer Science*, pp. 1–20. doi:10.4204/EPTCS.44.1.

Gansner, E. R., E. Koutsofios, S. C. North, et K.-P. Vo (1993). A technique for drawing directed graphs. *IEEE Trans. on Software Engineering 19*(3), 214–230, doi:10.1109/32.221135.

Geiß, R., G. V. Batz, D. Grund, S. Hack, et A. Szalkowski (2006). GrGen : A Fast SPO-Based Graph Rewriting Tool. In *Proc. of ICGT*, Volume 4178 of *Lecture Notes in Computer Science*, pp. 383–397. Springer, doi:10.1007/11841883_27.

Giakkoupis, G., R. Guerraoui, A. Jégou, A.-M. Kermarrec, et N. Mittal (2015). Privacy-Conscious Information Diffusion in Social Networks. In Y. Moses et M. Roy (Eds.), *DISC 2015*, Volume LNCS 9363 of 29^{th} *Int. Symp. on Distributed Computing*. Toshimitsu Masuzawa and Koichi Wada : Springer-Verlag Berlin Heidelberg, doi:10.1007/978-3-662-48653-5_32.

Golbeck, J. (2013). *Analyzing the Social Web*. Morgan Kaufmann.

Gomez-Rodriguez, M., J. Leskovec, et A. Krause (2012). Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data 5*(4), 21 :1–21 :37, doi:10.1145/2086737.2086741.

Goyal, A., F. Bonchi, et L. V. Lakshmanan (2010). Learning influence probabilities in social networks. In *Web Search and Data Mining, Proc. of the 3rd ACM Int. Conf. on*, WSDM '10, pp. 241–250. ACM, doi:10.1145/1718487.1718518.

Hadlak, S., H. Schumann, et H.-J. Schulz (2015). A survey of multi-faceted graph visualization. In *Eurographics Conference on Visualization (EuroVis). The Eurographics Association*, pp. 1–20. doi:10.2312/eurovisstar.20151109.

Hullman, J. (2019). The purpose of visualization is insight, not pictures : An interview with visualization pioneer ben shneiderman. https://medium.com/multiple-views-visualization-research-explained/the-purpose-of-visualization-is-insight-not-pictures-an-interview-with-visualization-pioneer-ben-beb15b2d8e9b. Publié le 12 mars 2019.

Keim, D. A., J. Kohlhammer, G. Ellis, et F. Mansmann (Eds.) (2010). *Mastering the Information Age. Solving Problems with Visual Analytics*. Eurographics Association, Goslar.

Keim, D. A., F. Mansmann, J. Schneidewind, J. Thomas, et H. Ziegler (2008). *Visual Analytics : Scope and Challenges*, pp. 76–90. Berlin, Heidelberg : Springer Berlin Heidelberg, doi:10.1007/978-3-540-71080-6_6.

Kejžar, N., Z. Nikoloski, et V. Batagelj (2008). Probabilistic inductive classes of graphs. *The Journal of Mathematical Sociology 32*(2), 85–109, doi:10.1080/00222500801931586.

Kempe, D., J. Kleinberg, et É. Tardos (2003). Maximizing the spread of influence through a social network. In *Proc. of the 9^{th} ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '03, pp. 137–146. ACM, doi:10.1145/956750.956769.

Kempe, D., J. Kleinberg, et É. Tardos (2005). Influential nodes in a diffusion model for social networks. In L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, et M. Yung (Eds.), *Automata, Languages and Programming*, Volume 3580 of *Lecture Notes in Computer Science*, pp. 1127–1138. Springer Berlin Heidelberg, doi:10.1007/11523468_91.

Kivelä, M., A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, et M. A. Porter (2014). Multilayer networks. *Journal of Complex Networks 2*(3), 203–271, doi:10.1093/comnet/cnu016.

Kuntz, P., B. Pinaud, et R. Lehn (2006). Minimizing crossings in a hierarchical digraphs with a hybridized genetic algorithm. *Journal of Heuristics 12*(1-2), 23–36, doi:10.1007/s10732-006-4296-7.

Lafont, Y. (1990). Interaction nets. In *Proceedings of the 17th ACM Symposium on Principles of Programming Languages (POPL'90)*, pp. 95–108. ACM Press, doi:10.1145/96709.96718.

Lee, B., C. Plaisant, C. S. Parr, J.-D. Fekete, et N. Henry (2006). Task taxonomy for graph visualization. In *Proceedings AVI workshop on BEyond time and errors : novel evaluation methods for information visualization*, Venice, Italy, pp. 1–5. ACM, doi:10.1145/1168149.1168168.

Lee, J., W.-S. Han, R. Kasperovics, et J.-H. Lee (2012). An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proc. VLDB Endow. 6*(2), 133–144, doi:10.14778/2535568.2448946.

Löwe, M. (1993). Algebraic approach to single-pushout graph transformation. *Theoretical Computer Science 109*(1–2), 181–224, doi:10.1016/0304-3975(93)90068-5.

Luciani, T., A. Burks, C. Sugiyama, J. Komperda, et G. E. Marai (2019). Details-first, show context, overview last : Supporting exploration of viscous fingers in large-scale ensemble simulations. *IEEE Transactions on Visualization and Computer Graphics 25*(1), 1225–1235, doi:10.1109/TVCG.2018.2864849.

Melançon, G. et F. Philippe (2004). Generating connected acyclic digraphs uniformly at random. *Information Processing Letters 90*(4), 209–213, doi:10.1016/j.ipl.2003.06.002.

Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Trans. on Visualization and Computer Graphics 15*(6), 921–928, doi:10.1109/TVCG.2009.111.

Munzner, T. (2014). *Visualization Analysis & Design*. AK Peters Visualization series.

Nabti, C. E. (2017). *Subgraph Isomorphism Search In Massive Graph Data*. Ph. D. thesis, Université de Lyon. https://tel.archives-ouvertes.fr/tel-01781831.

Namet, O. (2011). *Strategic Modelling with Graph Rewriting Tools*. Ph. D. thesis, King's College London.

Newman, M., A.-L. Barabási, et D. J. Watts (2006). *The structure and dynamics of networks*. Princeton Studies in Complexity. Princeton University Press.

Nobre, C., M. Meyer, M. Streit, et A. Lex (2019). The state of the art in visualizing multivariate graphs. *Computer Graphics Forum 38*, 807–832, doi:10.1111/cgf.13728.

Plotkin, G. D. (2004). A structural approach to operational semantics. *Journal of Logic and Algebraic Programming 60-61*, 17–139, doi:10.1016/j.jlap.2004.05.001.

Plump, D. (2009). The Graph Programming Language GP. In S. Bozapalidis et G. Rahonis (Eds.), *Algebraic Informatics CAI*, Volume 5725 of *Lecture Notes in Computer Science*, pp. 99–122. Springer, doi:10.1007/978-3-642-03564-7_6.

Plump, D. (2011). The design of GP 2. In *Proceedings 10th International Workshop on Reduction Strategies in Rewriting and Programming, WRS 2011, Novi Sad, Serbia, 29 May 2011.*, pp. 1–16. doi:10.4204/EPTCS.82.1.

Purchase, H. C. (2012). *Experimental human-computer interaction : a practical guide with visual examples.* Cambridge University Press, doi:10.1017/CBO9780511844522.

Renoust, B., G. Melançon, et T. Munzner (2015). Detangler : Visual analytics for multiplex networks. *Computer Graphics Forum 34*(3), 321–330, doi:10.1111/cgf.12644.

Rensink, A. (2003). The GROOVE Simulator : A Tool for State Space Generation. In *Applications of Graph Transformations with Industrial Relevance (AGTIVE)*, Volume 3062 of *Lecture Notes in Computer Science*, pp. 479–485. Springer, doi:10.1007/978-3-540-25959-6_40.

Robinson, I., J. Webber, et E. Eifrém (2015). *Graph Databases (2nd Edition).* O'Reilly Media.

Rodgers, P. (1998). A graph rewriting programming language for graph drawing. In *Proc. of IEEE Symp. on Visual Languages*, pp. 32–39. doi:10.1109/VL.1998.706131.

Sallaberry, A., F. Zaidi, et G. Melançon (2013). Model for Generating Artificial Social Networks having Community Structures with Small World and Scale Free Properties. *Social Network Analysis and Mining 3*(597–609), doi:10.1007/s13278-013-0105-0.

Sander, G. (1995). Graph layout through the VCG tool. In *Proc. Graph Drawing (GD'94)*, Volume 894 of *Lecture Notes in Computer Science*, pp. 194–205. Springer, Berlin, Heidelberg, doi:10.1007/3-540-58950-3_371.

Schürr, A., A. J. Winter, et A. Zündorf (1997). The PROGRES Approach : Language and Environment. In H. Ehrig, G. Engels, H.-J. Kreowski, et G. Rozenberg (Eds.), *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 2 : Applications, Languages, and Tools*, pp. 479–546. World Scientific.

Shneiderman, B. (1996). The eyes have it : A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symp. on Visual Languages*, pp. 336–343. IEEE Computer Society Press, doi:10.1109/VL.1996.545307.

Smith, A. M., W. Xu, Y. Sun, J. R. Faeder, et G. Marai (2012). Rulebender : integrated modeling, simulation and visualization for rule-based intracellular biochemistry. *BMC Bioinformatics 13*(8), S3, doi:10.1186/1471-2105-13-S8-S3.

Soni, U., Y. Lu, B. Hansen, H. C. Purchase, S. G. Kobourov, et R. Maciejewski (2018). The perception of graph properties in graph layouts. *Computuer Graphics Forum 37*(3), 169–181, doi:10.1111/cgf.13410.

Spönemann, M., H. Fuhrmann, R. von Hanxleden, et P. Mutzel (2010). Port constraints in hierarchical layout of data flow diagrams. In *Proc. of the* 17^{th} *Int. Symp. on Graph Drawing (GD'09)*, Volume 5849 of *Lecture Notes in Computer Science*, pp. 135–146. Springer, doi:10.1007/978-3-642-11805-0_14.

Ullman, J. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM 23*(1), 31–42, doi:10.1145/321921.321925.

Ullman, J. D. et J. Widom (2008). *A First Course in DATABASE SYSTEMS.* Pearson, Prentice Hall.

Vallet, J. (2017). *Where Social Networks, Graph Rewriting and Visualisation Meet : Application to Network Generation and Information Diffusion.* Ph. D. thesis, University of Bordeaux, France. https://tel.archives-ouvertes.fr/tel-01691037.

van den Elzen, S. et J. J. van Wijk (2014). Multivariate network exploration and presentation : From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics 20*(12), 2310–2319, doi:10.1109/TVCG.2014.2346441.

van Ham, F. et A. Perer (2009). Search, show context, expand on demand : Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics 15*(6), 953–960, doi:10.1109/TVCG.2009.108.

Varga, J. (2018). Finding the transitive closure of functional dependencies using strategic port graph rewriting. In *Proc. of the* 10^{th} *International Workshop on Computing with Terms and Graphs (TERMGRAPH)*, Volume 288 of *Electronic Proceedings in Theoretical Computer Science (EPTCS)*. doi:10.4204/EPTCS.288.5.

Vehlow, C., F. Beck, et D. Weiskopf (2017). Visualizing group structures in graphs : A survey. *Computer Graphics Forum 36*(6), 201–225, doi:10.1111/cgf.12872.

Ward, M., G. G. Grinstein, et D. Keim (Eds.) (2010). *Interactive Data Visualization : Foundations, Techniques, and Application*.

Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences 99*(9), 5766–5771, doi:10.1073/pnas.082090499.

Watts, D. J. et S. H. Strogatz (1998). Collective dynamics of s̀mall-worldńetworks. *Nature 393*, 440–442, doi:10.1038/30918.

Weber, M., M. Liwicki, et A. Dengel (2012). Faster subgraph isomorphism detection by well-founded total order indexing. *Pattern Recognition Letters 33*(15), 2011–2019, doi:10.1016/j.patrec.2012.04.017. Graph-Based Representations in Pattern Recognition.

Wenskovitch, J. E., L. A. Harris, J.-J. Tapia, J. R. Faeder, et G. E. Marai (2014). Mosbie : a tool for comparison and analysis of rule-based biochemical models. *BMC Bioinformatics 15*(1), 316, doi:10.1186/1471-2105-15-316.

Wonyeol, L., K. Jinha, et Y. Hwanjo (2012). Ct-ic : Continuously activated and time-restricted independent cascade model for viral marketing. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 960–965. doi:10.1109/ICDM.2012.40.