Réseaux multiplexes à travers les sciences sociales : une adaptation et des règles dictées par les données

Antoine Laumond, Bruno Pinaud, Guy Melançon

Réseaux multiplexes à travers les sciences sociales : une adaptation et des règles dictées par les données.

Laumond, Antoine
antoine.laumond(at)u-bordeaux(dot)fr

Pinaud, Bruno
bruno.pinaud(at)u-bordeaux(dot)fr

Melançon, Guy
guy.melancon(at)u-bordeaux(dot)fr

11 janvier 2017

Résumé

This work takes place in an interdisciplinary research project about human trafic in a criminal network. It is based on a cooperation between sociologists, jurists and computer scientists thus permitting to put forward a need of flexibility about the method and technologies to use when in front of heterogen and changeable data structure. Furthermore, the data takes the form of a multiplex network, a network where links between entities composing it define different meaningful layers. The multiplex oriented dataset and the variable shape of the data itself force a particular conception where graph databases can be a solution for efficient managing, analysis and visualization of the data.

Ces travaux s'inscrivent dans le cadre d'un projet relatif aux réseaux criminels de traite des humains. Il s'agit d'une coopération interdisciplinaire entre juristes, sociologues et informaticiens ayant permis de mettre en avant une nécessité de souplesse dans la méthodologie et les techniques utilisées afin de s'adapter à des données aux structures hétérogènes et évolutives. En outre, ces données prennent la forme d'un réseau multiplexe, un réseau dans lequel les liens entre les entités qui le composent définissent des couches à valeur sémantique forte. Ce jeu de données orienté nativement multiplexe ainsi que la nature des données elles-même imposent une conception particulière où les bases de données graphes peuvent être la solution pour une gestion, une analyse et une visualisation efficaces des données.

1 Introduction : le projet TETRUM

Ces travaux s'inscrivent dans le cadre du projet interdisciplinaire TETRUM (Projets Exploratoires Premier Soutien, PEPS 2015 CNRS/IdEx Bordeaux) qui met en collaboration sociologues (Centre Emile Durkheim CNRS UMR 5116), juristes (COMPTRASEC UMR 5114) et informaticiens (LaBRI UMR 5800).

Le projet TETRUM a pour objectif premier l'étude des réseaux de traite des humains. Cette analyse doit permettre à terme une compréhension accrue des modes opératoires des entités liées à la prostitution dans les réseaux nigérians. Pour ce faire ce groupe interdisciplinaire a été mis en place afin de traiter un corpus de documents vaste et complexe. C'est ce corpus qui fait la spécificité de ce projet : il s'agit de plusieurs dossiers judiciaires susceptibles de contenir environ 25000 pages. Ces pages traitent d'affaires répertoriées par la police et peuvent prendre des formes diverses : rapports d'interrogatoire, écoutes téléphoniques, témoignages, liste de numéros de téléphones suspects, etc.

Nous nous intéressons à l'étape préliminaire aux analyses juridiques ou sociologiques du réseau : il s'agit de concevoir un modèle abstrait de données pour stocker et interroger efficacement les données issues des dossiers judiciaires afin de modéliser le réseau criminel et ainsi de permettre son analyse par les juristes et sociologues. Des caractéristiques précises doivent cependant être prises en compte : un tel réseau se doit de pouvoir qualifier différents types de relations (relation de sang, financière, sexuelle, etc.) entre les acteurs qui le composent afin de répondre aux besoins analytiques des sociologues et juristes. Ce réseau social s'orchestre ainsi sous la forme d'un réseau multiplexe [1] i.e., un réseau composé de différentes couches déterminées par le type de relation de ses entités. Cet aspect va induire un certain nombre de contraintes et de difficultés qui seront détaillées ultérieurement.

Une autre caractéristique est de composer avec ce vaste ensemble de documents hétérogènes. Automatiser l'extraction des données susceptibles d'être exploitables est en effet difficile tant à cause de la nature du corpus que de la forme variable des données à extraire (ex : des locations renseignées que par l'adresse, par une ville et un code postal, par un pays, etc.). Les feuilles des dossiers ont été numérisées sous forme d'images PDF dont l'information textuelle n'est pas récupérable. En outre, les documents ont des structures qui diffèrent d'une page à l'autre : on ne sait pas à l'avance quel est le type du document, comment il est agencé et quelles sont les informations à récupérer. Un facteur humain est donc nécessaire pour permettre un jugement et une analyse efficace du document, imposant alors une méthodologie spécifique prenant en compte ce paramètre.

Du point de vue de l'informaticien, nous sommes clairement dans le contexte du "Big Data" tant au niveau des modèles et méthodes à utiliser et des technologies à mettre en œuvre. On définit classiquement "le Big Data" avec une série de V [2]. Dans le projet TETRUM, nous nous concentrons sur quatre V : Volume, Variété, Vélocité et Véracité. En effet, les différents types de lien du réseau ainsi que l'aspect changeant et polymorphe des données contraignent l'usage d'une méthodologie spécifique nécessitant une certaine souplesse dans le modèle de

donnée qui sera détaillée dans la dernière partie de l'article.

2 Un problème ancien et actuel : le Big Data

Les difficultés citées dans la partie précédente font écho à une problématique actuelle majeure : celle de la gestion de la connaissance et de l'explosion de la masse de données. Cependant, cette problématique est loin d'être nouvelle. Déjà en l'an 55, Sénèque écrivait : *"A quoi sert-il d'avoir un nombre illimité de livres qu'un lecteur ne pourra pas lire durant toute sa vie ? Cette somme d'ouvrages pèse sur l'étudiant sans l'instruire"*. Bien des années plus tard, Diderot avait des préoccupations similaires en écrivant dans l'Encyclopédie : *"Dans le futur, le nombre de livres va continuer d'augmenter régulièrement. On peut ainsi prédire qu'il sera un jour presque aussi difficile d'apprendre quoi que ce soit des livres que de l'étude directe de l'univers"*[1]. La problématique de pouvoir gérer un flot d'informations sans se laisser submerger est donc une question majeure dont l'ombre d'une solution satisfaisante n'est survenu que récemment. Même si ce terme a été plus utilisé à des fins commerciales et médiatiques que dans les milieux scientifiques, ce problème est bien celui du Big Data et des fameux 4 V présentés dans l'introduction : Volume, Variété, Vélocité et Véracité.

Le volume est l'aspect le plus évident. Une masse conséquente de données est une contrainte rendant le stockage et l'exploitation en temps réel difficile. Les requêtes pour récupérer ou modifier de l'information tendent alors à devenir coûteuses en temps, obligent à optimiser les traitements au maximum et à limiter les échanges avec la base de données, ce qui n'est pas toujours réalisable. Le volume est un aspect très présent dans TETRUM avec ses trois dossiers (ce nombre est amené à augmenter dans le futur) de 25000 pages mais pas nécessairement au niveau informatique. Il s'agit presque exclusivement de données texte n'induisant donc ni un poids élevé ni un traitement particulièrement lourd. Cependant, cela n'est valable que dans le cas d'un procédure automatisée, ce qui n'est pas le cas ici. Ces données doivent être analysées manuellement par les juristes, ce qui représente alors un investissement majeur et difficile pour un être humain.

La variété est un aspect dont les conséquences en terme de performance et développement ne sont pas nécessairement perceptibles au premier abord. Par variété, on entend la capacité à gérer des données aux types différents, provenant potentiellement de différentes sources. Parmi les données de même type, la structure peut différer rendant une généralisation du traitement toujours plus difficile. Composer avec la variété impose donc une structure souple et générique pouvant s'adapter aux différents types pouvant être rencontrés. Avec TETRUM, les informations récupérées n'ont pas de type prédéfini et peuvent à priori être extrêmement variées (un nom, une date, un numéro de téléphone, etc.).

1. citations récupérées sur http ://www.superception.fr/2012/02/09/la-democratisation-de-linformation/

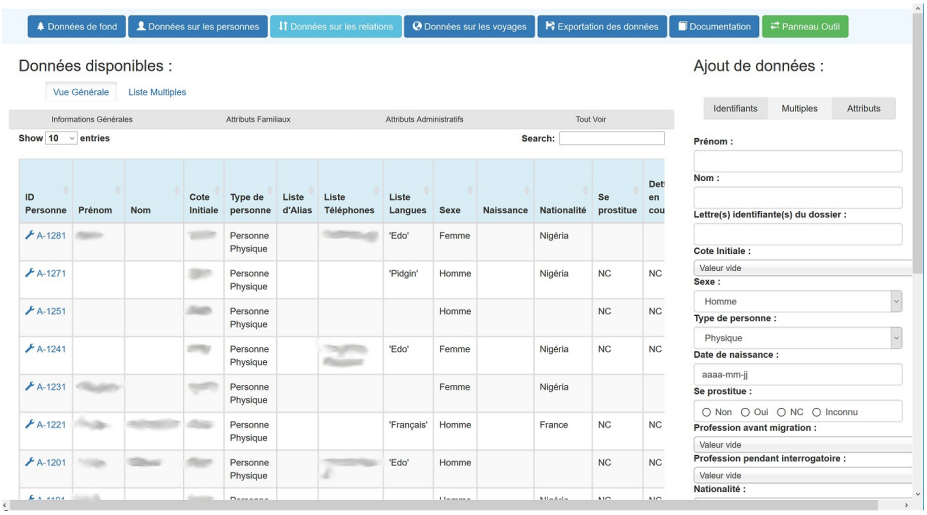

FIGURE 1 – Aperçu de l'application TETRUM

La vélocité fait directement écho aux deux précédents aspects. Tant un volume élevé de données que des données variées vont induire un traitement plus complexe pour faire face au besoin de généricité et à la masse d'information à traiter. Obtenir un temps réel lors de l'exploitation des données peut alors devenir un véritable challenge qu'il est néanmoins nécessaire de réussir pour une étude efficace des données.

La véracité, enfin, est un point important qui va dicter la suite de nos travaux. On peut définir la véracité par la qualité des données utilisées. Des valeurs manquantes, des imprécisions ou des valeurs erronées sont autant de facteurs qui vont entacher la qualité des données et rendre leur exploitation plus difficile voir rendre les résultats obtenus imprécis ou erronés. S'assurer de la qualité des données est donc un facteur capital pour assurer la pertinence des études effectuées. Dans TETRUM, il est impossible de faire abstraction du facteur humain pour extraire les données. Des travaux ont donc été nécessaires pour donner le moyen aux juristes et sociologues de récupérer les données sans que la véracité en pâtisse.

C'est dans l'optique de répondre à ces aspects tout en trouvant une solution aux éléments énoncés dans la première partie (conceptualiser différents types de lien et composer avec un ensemble massif de documents hétérogènes) que nous avons développé un programme d'aide à la saisie pour les juristes et sociologues.

3 L'application TETRUM

Avant notre collaboration, les données étaient stockées dans un fichier Excel sur de nombreuses feuilles de calculs et de très nombreuses colonnes. Une saisie intégralement manuelle n'était pas viable sur le long terme, nous avons donc

développé une solution qui puisse répondre aux nombreux besoins de ce projet. Nous avons donc développé une application en ligne (1) afin de structurer et hiérarchiser les données extraites par les juristes et ainsi les aider à naviguer parmi elles. Les juristes sont contraints de parcourir manuellement les dossiers afin d'identifier les éléments qui permettent d'enrichir nos connaissances sur le réseau. Cette démarche est fastidieuse mais nécessaire tant le facteur humain est indispensable.

Cette application est développée en PHP/Javascript/HTML5 et est hébergée sur un serveur distant et fonctionne conjointement avec une base de données relationnelle (MySQL). Des formulaires permettent la création de trois types majeurs de données : les personnes, les relations et ce que nous appelons les données de fond. Le terme "données de fond" qualifie les données utilisées par les personnes ou les relations. Ainsi, une adresse est une donnée de fond qui pourrait être liée à des personnes y habitant ou à des relations (un échange financier ayant eu lieu à cette adresse par exemple). Le spectre des données de fond est très large : source, rôle, action dans le réseau, contexte socio-géographique, alias, profession, etc. C'est en utilisant ces données de fond et en les proposant via des champs de saisie semi-automatique [2] que le travail des juristes est facilité. Ainsi les juristes et sociologues enrichissent progressivement leur champ de possibilités en ajoutant les données de fond parallèlement à la saisie des nouvelles personnes et relations. Cette application permet aussi d'explorer les données et de chercher des informations particulières en triant ou cherchant certaines valeurs parmi les entités existantes. C'est un détail important car les informations autour d'une personne ou d'une relation ne surviennent pas nécessairement dans les même documents. Il est alors possible par exemple que les données relatives à une personne soient complétées (ou devenues fausses) plus tard, il est alors important de pouvoir facilement recouper certaines informations afin de retrouver rapidement la personne concernée afin de compléter ce que l'on sait d'elle.

TETRUM est un projet exploratoire limité dans le temps (une année). L'application a donc été développée simultanément avec l'analyse et l'extraction des premières données. L'application a ainsi évoluée progressivement en fonction des nouveaux besoins rencontrés. C'est pendant son développement qu'un certain nombre de difficultés et d'améliorations potentielles ont été soulevées et c'est ce que nous allons développer dans la prochaine partie.

4 Obstacles et limitations

4.1 Le nécessaire développement incrémental du modèle sociologique

Une des conséquences de ce développement incrémental est le fait de ne pas avoir pu établir un cahier des charges fixe afin de savoir dans quelle direction

2. Auto-complétion ou recherche de la valeur souhaitée en fonction des valeurs préexistantes dans la base.

diriger la conception du modèle de données et de l'interface. Lorsque le développement du projet a commencé, il était difficile tant pour les informaticiens que pour les juristes et sociologues d'estimer les besoins qui allaient être rencontrés. Les choix d'implémentations ont ainsi été réalisés conformément aux réseaux traditionnellement rencontrés [3] dans le domaine sociologique en utilisant des technologies et des méthodes d'implémentations bien connues. Au fur et à mesure de l'avancée des juristes dans l'analyse des dossiers, de nouveaux éléments exprimant de nouveaux besoins étaient découverts régulièrement, incluant de fait un changement au niveau de la base de données et de son modèle. Il était impossible de prévoir ne serait-ce quel type de donnée allait être rencontré dans les pages suivantes du dossier.

4.2 Des données inévitablement polymorphes

Outre l'apparition régulière de nouveaux types de données, un autre écueil est l'aspect polymorphe et changeant de données sémantiquement identiques. Les exemples types sont les informations temporelles ou spatiales. Les informations contenues dans les dossiers ne sont pas toujours précises (lors d'un interrogatoire, un individu peut essayer de rester volontairement évasif par exemple) et cet aspect est difficilement retranscriptible dans une base de données relationnelle. Une relation financière peut par exemple être renseignée uniquement par une année, un mois mais aussi par un jour de la semaine voire par un intervalle de temps dont on ne connaît pas forcément les bornes. On peut même imaginer des choses encore plus imprécises comme "après un événement A" ou "entre l'année 2015 et l'événement A". Ainsi dans la base de données et l'interface, pour chaque information temporelle, nous avons deux champs date "début"/"fin" en cas de période précise (dont seul le premier champ est utilisé en cas de date ponctuelle et non d'intervalle), deux champs texte "début approximation"/"fin approximation" en cas de période approximative ainsi qu'un champ de texte libre "note" en cas d'informations encore plus brumeuses. Le problème est similaire avec les informations spatiales où l'on ne peut avoir uniquement qu'une adresse, une ville, une localisation plus ou moins précise etc. En plus de cet aspect polymorphe des données, il faut garder à l'esprit que les données saisies sont susceptibles d'évoluer au fur et à mesure de la lecture des dossiers. Ainsi une information temporelle peut par exemple changer de forme et devenir un intervalle précis là où elle n'était qu'une approximation de prime abord. Tous ces éléments mettent en évidence une absolue nécessité de souplesse dans le modèle de données que le modèle relationnel n'est pas à même d'offrir de par sa conception [6]. Le modèle de données est composé de très nombreuses relations de fort degré. De nombreux attributs ne sont jamais exploités simultanément et laissent nombre de valeurs nulles dans la base. Les requêtes, l'affichage et la saisie s'en retrouvent fortement dégradées en terme de lisibilité, de maintenabilité et d'efficacité (voir 2).

4.3 Multiplexité : un manque de souplesse criant du modèle relationnel

Une des caractéristiques précédemment citées est la nécessité de pouvoir modéliser différents types de lien entre les entités du réseaux. Ainsi entre deux acteurs A et B du réseau, on peut par exemple observer une relation de sang (A est le frère de B), une relation financière (A fait un virement à B), une relation sexuelle (A s'est prostitué avec B), etc. Le réseau comporte actuellement 7 différents types de lien qui ont des attributs complètement différents. Ces différents liens possibles entre les entités du réseau permettent la représentation de ce que l'on appellera des couches du réseau multiplexe. Ainsi, l'ensemble des liens sexuels définit la couche "sexuelle", etc. Le dernier problème soulevé est directement lié à cette forme du jeu de données. Si le contexte initial du projet nous a fait opter pour le modèle relationnel, l'apparition de nouveaux besoins pour les juristes a mis en évidence que ce modèle n'est pas adapté pour modéliser de multiples relations entre différentes entités dans un réseau. Pour une simple requête "trouver la personne A", cela ne pose aucun problème et un simple "SELECT ... FROM ... WHERE ..." suffit. Lorsque l'on souhaite "trouver A qui est lié à B par une relation quelconque qui lui même est lié à C qui lui même ...", cela devient très vite difficile. Mais ce type de requête est primordiale pour effectuer une analyse correcte du réseau criminel. Cela nécessite de parcourir les 7 tables des différents types de liens pour tester à chaque fois si un lien existe entre deux personnes ainsi que de re-parcourir autant de fois la table des personnes que de personnes identifiées dans la requête. Une telle requête nécessite plusieurs jointures sur la table des personnes ainsi que de multiples jointures avec les 7 tables des 7 types de relation. Même si le système de gestion de bases de données est capable d'optimiser les requêtes SQL, la requête équivaut à un produit cartésien d'autant d'ensembles que de jointures présentes dans la requête. Cependant ce n'est pas la seule difficulté. Les "données de fond" précédemment citées sont des données liées tant aux personnes qu'aux relations. En outre, un nombre indéfini de données de fond peuvent être liées à la même personne ou relation (une personne peut avoir de multiples numéros de téléphone ou une relation peut avoir de multiples endroits). De telles données impliquent au niveau du modèle de nombreuses tables annexes qui nécessitent chacune des tables d'association afin de lier les données de fond aux personnes et aux relations. On ajoute donc aux requêtes autant de jointures supplémentaires que de données de fond à récupérer. Les requêtes atteignent alors rapidement une complexité élevée et des performances désastreuses.

5 Une solution : les bases de données orientées graphe

Nous nous sommes alors tournés vers les bases de données orientée graphe. Ces bases de données NoSQL [5] ont la particularité de stocker nativement leurs données sous la forme d'un graphe et acquièrent ainsi un certain nombre de

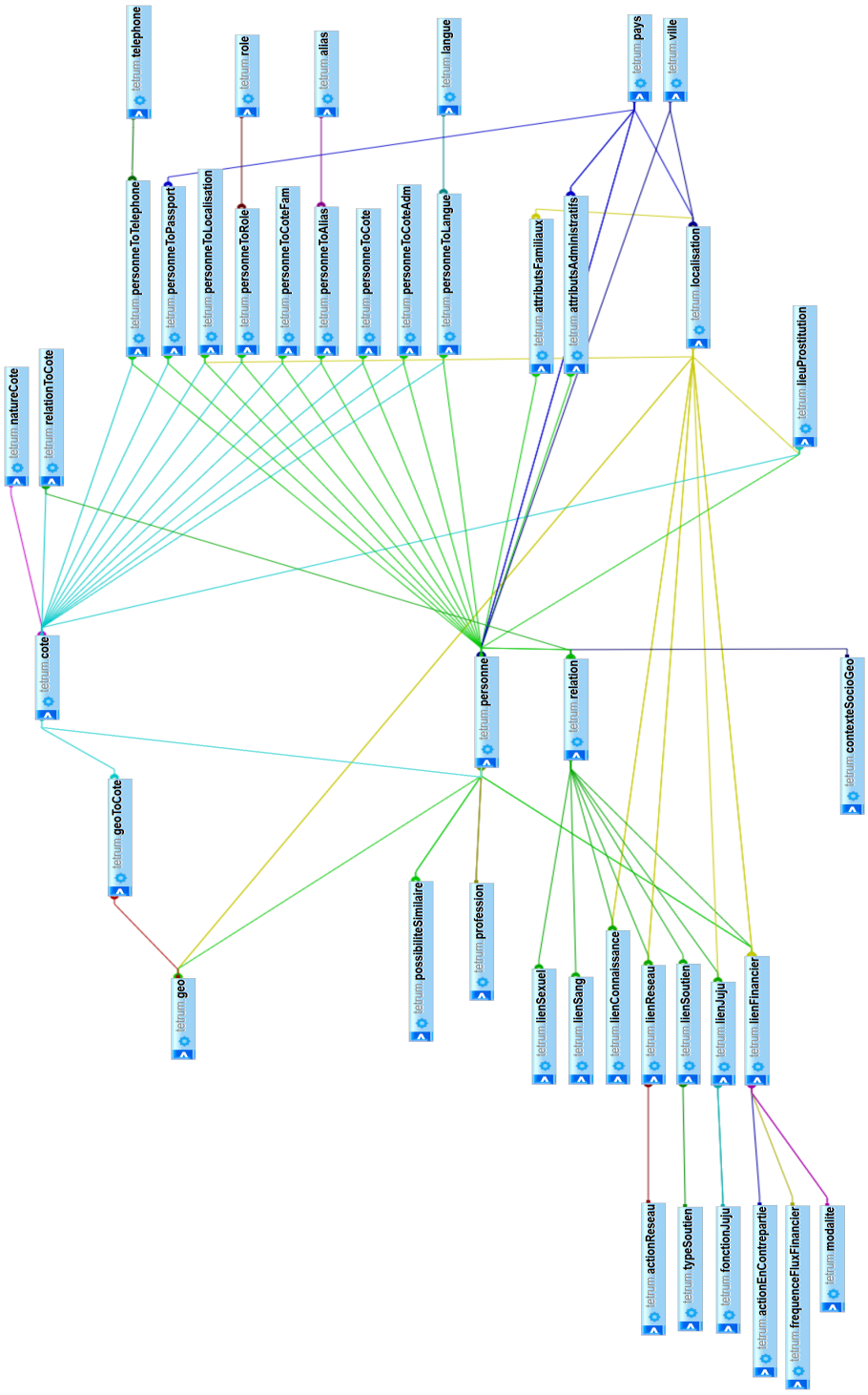

FIGURE 2 – Aperçu de la haute connectivité au sein de la base relationnelle

propriétés qui seront à même de résoudre nos limitations.

Dans une base de données graphe [4], tout entité stockée prend la forme d'un sommet. Nous avons choisi de travailler avec le logiciel Neo4J. Le modèle de données est simple : les sommets sont étiquetés et un sommet peut avoir plusieurs étiquettes. On peut ensuite ajouter un nombre quelconque d'attributs clé-valeur à chaque sommet. Dans une BD graphe, il n'y a pas donc de structure stricte comme pour le modèle relationnel. Il est ainsi possible d'avoir deux sommets avec des étiquettes communes mais des attributs différents. Les étiquettes sont principalement utilisées pour identifier, indexer ou chercher les sommets initiaux dans une requête sans avoir à parcourir l'ensemble des sommets de la base. En plus des sommets, on a la possibilité d'insérer un ou plusieurs liens entre les sommets. Les liens ne peuvent avoir qu'une seule étiquette mais un nombre quelconque d'attributs. Dans TETRUM, les étiquettes des liens donnent des indications sur la nature de la relation (financière, familiale, sexuelle, etc.).

Nous avons déjà présenté la difficulté de ne pas pouvoir prévoir le modèle de données initial puisque nous découvrons le modèle au fur et à mesure de la lecture des dossiers par les juristes et sociologues. C'est ici que le modèle graphe prend tout son sens dans la mesure où il n'est pas figé. On peut donc le faire évoluer sans risque de compromettre la structure et la cohérence de la base existante. Le fait d'ajouter un nouveau type de données de fond correspond à ajouter un certain nombre de sommets avec une nouvelle étiquette propre à ce nouveau type de donnée. On n'a pas à remettre en cause l'existant. Idéalement, il est préférable, même avec une base de données graphe, d'avoir une idée du modèle initial afin d'orienter et d'optimiser sa conception. Néanmoins, une BD graphe offre une très grande souplesse par rapport au modèle relationnel lorsque trop d'inconnues sont présentes.

Un autre point concerne la forme changeante des données. Comme énoncé précédemment, ce problème est résolu par la souplesse d'évolution du modèle. Deux sommets avec une étiquette donnée A peuvent avoir des attributs radicalement différents l'un de l'autre car ceux-ci dépendent entièrement du sommet et non de ses étiquettes. Ainsi ajouter de nouveaux paramètres pour enrichir ou modifier le modèle d'une entité (personne, donnée de fond ou relation) ne force plus à devoir ré-écrire les requêtes ni changer le modèle de la base. Pour ce qui est de l'imprécision de certaines valeurs comme les adresses ou les dates, le modèle graphe propose une solution qui lui est propre. On peut par exemple décomposer une information spatiale en plusieurs entités : “pays”, “ville”, “adresse”,“code postal" et “descriptif du lieu" qui sont stockées en temps que sommets dans le graphe. On lie alors ces éléments dans un arbre à partir d'un sommet racine “localisation” : une adresse est liée à une ville, elle même à un pays, etc. Ainsi, avoir des informations incomplètes amenées à être enrichies ultérieurement (comme n'avoir qu'une adresse ou qu'une ville) ne nécessite plus d'avoir à jongler entre plusieurs tables d'associations et table d'entités comme avec le modèle relationnel. De manière assez similaire, pour les informations temporelles, on peut créer un arbre où les sommets sont de type différent et de plus en plus précis en fonction de leur profondeur : à partir d'une racine “time-tree” [4, p.72] dont les fils sont des années qui eux mêmes ont des fils mois et ainsi de suite

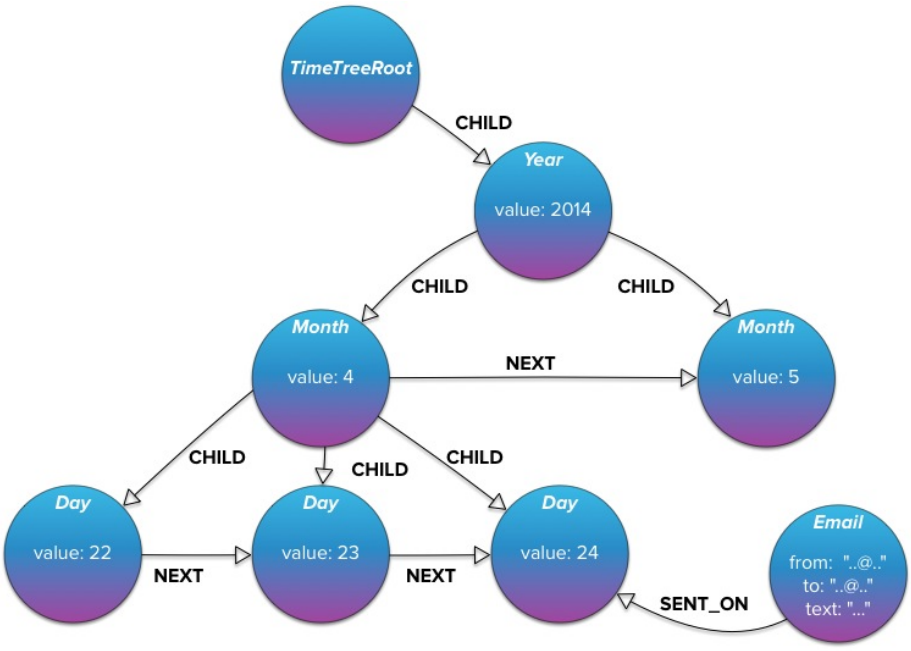

FIGURE 3 – Exemple d'un time-tree

jusqu'aux feuilles qui peuvent être des jours, des heures ou encore plus précis si nécessaire (voir 3 [3]).

Ainsi, on fabrique les chemins adéquats et on les attribue en liant les sommets qui nous intéressent aux entités à renseigner. S'il y a nécessité d'avoir des informations imprécises, on peut aussi lier à la racine de l'arbre "time-tree" des sommets avec des champs texte (ex : "après que personne A soit revenue"). On peut alors librement renseigner n'importe quel format de date tout en conservant une certaine hiérarchie. Pour renseigner un intervalle, des liens baptisés "début" et/ou "fin" vers deux feuilles de l'arbre de temps permet de modéliser n'importe quelle période. Enfin, il reste la question de la modélisation des liens et des problèmes de performance que cela engendre. C'est ici la caractéristique la plus importante des bases de données graphe : chercher une entité liée à une autre ne nécessite pas de parcourir plusieurs fois le même ensemble de données. En effet, la base de donnée graphe est conçu pour accéder rapidement au voisin d'un élément donné. L'information d'adjacence est toujours accessible quelle que soit l'entité observée. On peut alors éviter les produits cartésiens lors des requêtes et ainsi avoir des performances bien supérieures à ce qui a été obtenu précédemment. C'est aussi cette caractéristique qui fait que les solutions précédemment évoquées ne sont pas coûteuses. Rajouter des entités supplémentaires

3. Image provenant de http ://graphaware.com/neo4j/2014/08/20/graphaware-neo4j-timetree.html

dans le cas des informations temporelles et spatiales ne coûte presque rien en terme d'espace ou de calcul. Les requêtes permettent de chercher facilement des entités ou relations ayant un label donné et l'on peut indiquer directement un pattern pour trouver le chemin qui nous intéresse (dans le cas du "time-tree" par exemple). A l'image des BD relationnelles, il est aussi possible de créer des indexes sur les sommets fréquemment sollicités en fonction de leurs attributs si les requêtes en fonction des étiquettes (qui peuvent être vus comme des indexes) ne suffisent pas. Il résulte de tout cela un développement facilité au niveau de l'interrogation de la base et une réactivité bien supérieure de l'application due aux meilleurs performances des requêtes.

6 Conclusion

Au final, pour s'adapter aux méthodes de travail des juristes et sociologues et à la découverte itérative du modèle de données, les bases de données orientées graphe offre la souplesse nécessaire que le modèle relationnel ne peut pas fournir. Cependant, il ne faut pas oublier l'apport du modèle relationnel. Le choix de la base de données doit être réfléchi en amont en fonction des spécificités du projet. Il est parfois nécessaire d'avoir un modèle de données à la structure rigide afin de s'assurer de la cohérence de la base (gestion de stock pour un site marchand). Auquel cas, un modèle aussi souple que le modèle graphe n'est pas forcément la meilleure option. De la même manière, de simples données "clé-valeur" ne nécessiteront pas forcément tous les outils offerts par cette solution. Cependant, en cas de développement exploratoire et incrémental ou de données typées réseaux simples ou multiplexes, les bases de données orientées graphe offrent une souplesse et une efficacité indéniables et nécessaires. Les sciences sociales étant un terrain favorable pour rencontrer ces caractéristiques, il est intéressant d'envisager l'utilisation des bases de données graphes dans ce domaine à plus large échelle. En outre, il est très intuitif de stocker un réseau dans une base de données graphe tant le modèle de la base sera similaire au modèle initial des données. Dans l'hypothèse d'une visualisation typée nœud-lien, utiliser une base de données graphe ne pourra même que réduire les procédures nécessaires pour adapter les données à la visualisation.

Remerciements

Merci à Bénédicte Lavaud-Legendre, Hélène Pohu et Cécile Pléssard pour leur partage d'expertise et leur bonne humeur perpétuelle.

Références

[1] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, (2) :203–271, 2014.

[2] Douglas Laney. 3D data management : Controlling data volume, velocity, and variety. Technical report, META Group, February 2001.

[3] Paul F. Lazarsfeld. Evidence and inference in social research. *Daedalus*, 87(4) :99–130, 1958.

[4] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. O'Reilly Media, Inc., 2013.

[5] Pramod J. Sadalage and Martin Fowler. *NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley Professional, 1st edition, 2012.

[6] Jeffrey D. Ullman and Jennifer Widom. *First Course in Database Systems*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2007.