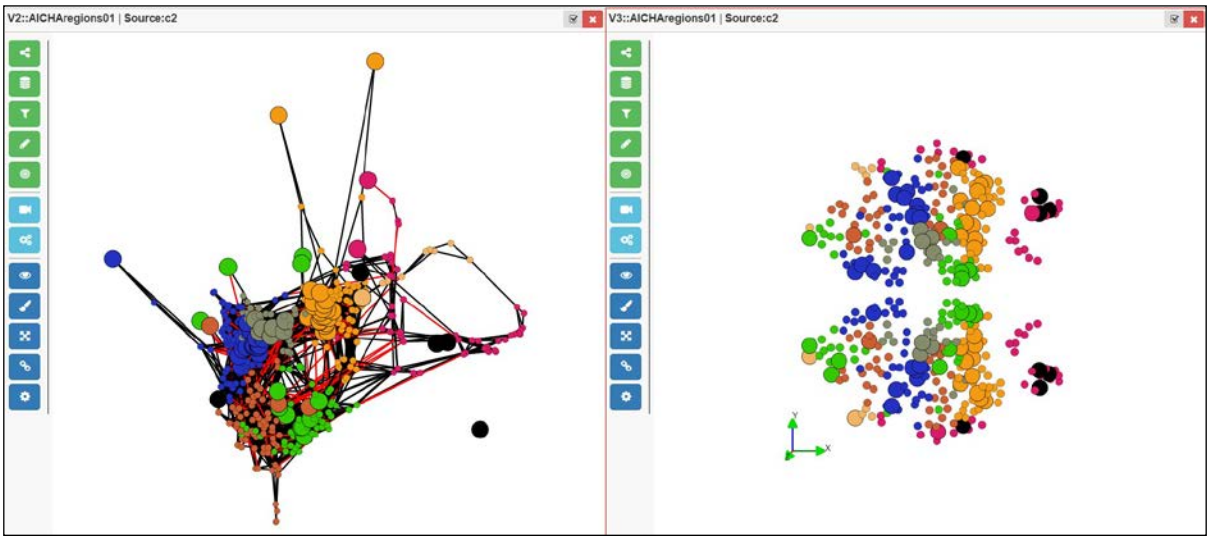
Gestion et visualisation de données hétérogènes multidimensionnelles : application PLM à la neuroimagerie

Marianne Allanic

Par **Marianne ALLANIC**

Gestion et visualisation de données hétérogènes multidimensionnelles : application PLM à la neuroimagerie

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 17 décembre 2015
Spécialité : Mécanique Avancée

D2248

UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE
ECOLE DOCTORALE

Doctorat
Mécanique avancée

Marianne ALLANIC

GESTION ET VISUALISATION DE DONNEES HETEROGENES MULTIDIMENSIONNELLES : APPLICATION PLM A LA NEUROIMAGERIE

Thèse dirigée par Benoît EYNARD et Marc JOLIOT
Soutenue le 17 décembre 2015

Jury :

Marija JANKOVIC, Maître de Conférences HDR, CentraleSupélec (*rapporteur*)

Jean-Baptiste POLINE, Directeur de Recherche, University of California (*rapporteur*)

Christophe EGLES, Professeur des Universités, Université de Technologie de Compiègne (*examinateur*)

Samuel GOMES, Professeur des Universités, Université de Technologie de Belfort-Montbéliard (*examinateur*)

Alexandre DURUPT, Maître de Conférences, Université de Technologie de Compiègne (*examinateur*)

Philippe BOUTINAUD, Directeur R&D et innovation, CADESIS (*examinateur*)

Marc JOLIOT, Directeur de Recherche, Université de Bordeaux (*examinateur*)

Benoît EYNARD, Enseignant Chercheur HDR, Université de Technologie de Compiègne (*examinateur*)

Résumé : La neuroimagerie est confrontée à des difficultés pour analyser et réutiliser la masse croissante de données hétérogènes qu'elle produit. La provenance des données est complexe – multi-sujets, multi-analyses, multi-temporalités – et ces données ne sont stockées que partiellement, limitant les possibilités d'études multimodales et longitudinales. En particulier, la connectivité fonctionnelle cérébrale est analysée pour comprendre comment les différentes zones du cerveau travaillent ensemble. Il est nécessaire de gérer les données acquises et traitées suivant plusieurs dimensions, telles que le temps d'acquisition, le temps entre les acquisitions ou encore les sujets et leurs caractéristiques. Cette thèse a pour objectif de permettre l'exploration de relations complexes entre données hétérogènes, ce qui se décline selon deux axes : (1) comment gérer les données et leur provenance, (2) comment visualiser les structures de données multidimensionnelles. L'apport de nos travaux s'articule autour de trois propositions qui sont présentées à l'issue d'un état de l'art sur les domaines de la gestion de données hétérogènes et de la visualisation de graphes.

Le modèle de données *BMI-LM* (*Bio-Medical Imaging – Lifecycle Management*) structure la gestion des données de neuroimagerie en fonction des étapes d'une étude et prend en compte le caractère évolutif de la recherche grâce à l'association de classes spécifiques à des objets génériques. L'implémentation de ce modèle au sein d'un système PLM (Product Lifecycle Management) montre que les concepts développés depuis vingt ans par l'industrie manufacturière peuvent être réutilisés pour la gestion des données en neuroimagerie. Les *GMD* (*Graphes Multidimensionnels Dynamiques*) sont introduits pour représenter des relations complexes entre données qui évoluent suivant plusieurs dimensions, et le format *JGEX* (*Json Graph EXchange*) a été créé pour permettre le stockage et l'échange de GMD entre applications. La méthode *OCL* (*Overview Constraint Layout*) permet l'exploration visuelle et interactive de GMD. Elle repose sur la préservation partielle de la carte mentale de l'utilisateur et l'alternance de vues complètes et réduites des données. La méthode OCL est appliquée à l'étude de la connectivité fonctionnelle cérébrale au repos de 231 sujets représentées sous forme de GMD – les zones du cerveau sont représentées par les nœuds et les mesures de connectivité par les arêtes – en fonction de l'âge, du genre et de la latéralité : les GMD sont obtenus par l'application de chaînes de traitement sur des acquisitions IRM dans le système PLM. Les résultats montrent deux intérêts principaux à l'utilisation de la méthode OCL : (1) l'identification des tendances globales sur une ou plusieurs dimensions et (2) la mise en exergue des changements locaux entre états du GMD.

Mots clés : Gestion des données ; Données Hétérogènes ; Données Multidimensionnelles ; Product Lifecycle Management ; Visualisation ; Exploration ; Théorie des Graphes ; Neuroimagerie

Abstract : Neuroimaging domain is confronted with issues in analyzing and reusing the growing amount of heterogeneous data produced. Data provenance is complex – multi-subjects, multi-methods, multi-temporalities – and the data are only partially stored, restricting multimodal and longitudinal studies. Especially, functional brain connectivity is studied to understand how areas of the brain work together. Raw and derived imaging data must be properly managed according to several dimensions, such as acquisition time, time between two acquisitions or subjects and their characteristics. The objective of the thesis is to allow exploration of complex relationships between heterogeneous data, which is resolved in two parts : (1) how to manage data and provenance, (2) how to visualize structures of multidimensional data. The contribution follow a logical sequence of three propositions which are presented after a research survey in heterogeneous data management and graph visualization.

The *BMI-LM* (*Bio-Medical Imaging – Lifecycle Management*) data model organizes the management of neuroimaging data according to the phases of a study and takes into account the scalability of research thanks to specific classes associated to generic objects. The application of this model into a PLM (Product Lifecycle Management) system shows that concepts developed twenty years ago for manufacturing industry can be reused to manage neuroimaging data. *GMDs* (*Dynamic Multidimensional Graphs*) are introduced to represent complex dynamic relationships of data, as well as *JGEX* (*Json Graph EXchange*) format that was created to store and exchange GMDs between software applications. *OCL* (*Overview Constraint Layout*) method allows interactive and visual exploration of GMDs. It is based on user's mental map preservation and alternating of complete and reduced views of data. OCL method is applied to the study of functional brain connectivity at rest of 231 subjects that are represented by a GMD – the areas of the brain are the nodes and connectivity measures the edges – according to age, gender and laterality : GMDs are computed through processing workflow on MRI acquisitions into the PLM system. Results show two main benefits of using OCL method : (1) identification of global trends on one or many dimensions, and (2) highlights of local changes between GMD states.

Keywords : Data Management ; Heterogeneous Data ; Multidimensional data ; Product Lifecycle Management ; Visualisation ; Exploration ; Graph Theory ; Neuroimaging

Remerciements

Je tiens tout d'abord à remercier les membres du jury pour les échanges enrichissants que nous avons eu. Merci à Christophe Egles d'avoir excepté de présider mon jury, à Marija Jankovic et Jean-Baptiste Poline, rapporteurs de ma thèse, pour leurs remarques pertinentes et leurs conseils, et enfin à Samuel Gomes de s'être rendu disponible pour ma soutenance malgré son éloignement géographique et le décalage horaire.

Mes remerciements les plus sincères sont adressés à mes directeurs et encadrants, pour leur bienveillance et pour la confiance qu'ils m'ont accordé durant ces trois années de doctorat – incluant leur première expérience d'acteurs de cinéma : Philippe Boutinaud, Benoît Eynard, Marc Joliot et Alexandre Durupt.

Je remercie Xavier Ruffenach, directeur de la société Cadesis, pour avoir financé ce doctorat, ainsi que les équipes de Cadesis pour m'avoir intégrée et encouragée, avec une pensée particulière pour mes collègues de l'agence de Courbevoie et pour ceux avec lesquels j'ai été amenée à travailler : Thierry, Nicolas – mention spéciale –, Olivier et Jérôme de l'équipe R&D, et Pierre et Arthur de l'équipe PLM.

Les membres de l'équipe SIM de Roberval m'ont accueilli chaleureusement à chacune de mes visites et je tiens à les en remercier, tant le personnel administratif – Muriel, Sylvie –, que scientifique – Matthieu, Emmanuel, Joanna, Charles, Flore, Frédéric, Julien, Magali, Laurent, Christine... J'ai une pensée affectueuse pour les doctorants de l'équipe : Gaëtan – compère de toujours –, Christophe, Marina, les deux Fabien, Chen, Cong, et tous les autres !

Merci aux membres du GIN d'avoir partagé leur expertise métier et de m'avoir ouvert la porte sur le monde fascinant des neurosciences, en particulier Pierre-Yves avec lequel j'ai beaucoup échangé dans le cadre du projet Biomist, et Gaëlle.

Je remercie également tous ceux que j'ai rencontré durant mon doctorat et avec lesquels j'ai pu échanger ou qui m'ont soutenu, ils se reconnaitront.

Merci à Damien pour tous les beaux projets que nous partageons.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

L'ère du numérique

Les données sont devenues omniprésentes dans nos vies. Les productions scientifique et industrielle n'ont jamais été aussi nombreuses, mais ce sont aussi nos activités quotidiennes qui génèrent une quantité croissante de données : que ce soit en allant faire nos courses ou en navigant sur internet, chacune de nos actions génère de l'information. Il est devenu facile de produire, de reproduire et de modifier des données depuis qu'elles sont sous forme numérique, et c'est Internet qui garantit leur diffusion à travers le monde, multipliant les échanges qui sont devenus quasi-instantanés.

Selon Keller & Tergan (2005), on appelle *donnée* une unité brute qui peut être un symbole ou un fait isolé et non interprété. Une *information* est une donnée à laquelle du sens est attribué dans un contexte défini. Une information pour une personne peut demeurer une donnée pour d'autres personnes qui n'ont pas accès à sa signification, faute de contexte ou d'outil de lecture. La *connaissance* est une information qui a été transformée et intégrée dans une structure de connaissances humaine existante.

Si produire des données n'a jamais été aussi facile, leur donner du sens pour les transformer en information exploitable, voire en connaissance, est difficile. L'hétérogénéité (origine, nature, format, degré de structuration), la redondance et le partage (gestion de l'accès et de la sécurité) des données augmentent d'autant la complexité de leur analyse. Comment trouver une information pertinente ? Quels outils fournir pour naviguer dans ce "déluge de données" ? Une recherche active existe pour faire face à ces nouveaux défis. Des moteurs de recherche puissants sont développés et l'idée d'un web sémantique a largement fait son chemin. La navigation et l'analyse de grandes quantités de données constituent une branche très active de l'économie, et s'appliquent à de nombreux domaines : analyse de réseaux (sociaux, criminels, énergie...), biologie (bases de données scientifiques à grande échelle de protéines ou de gènes), médecine (suivi des données physiologiques de patients ayant des maladies chroniques, parcours médical...), publicité ciblée (vente d'espaces publicitaires en fonction du profil de l'internaute) ou encore projections marketing (analyse des tendances).

Les données sont facilement duplicables, et elles n'en sont pas moins extrêmement volatiles : les formats évoluent très vite et les supports cessent au fur et à mesure de prendre en charge les formats les plus anciens, condamnant des données à devenir indéchiffrables. Il s'agit d'une obsolescence des données, du point de vue de leur intégrité. Ces données ne sont pas nettoyées et

restent dans la masse globale, inutilisables. Si les données sont faciles à produire, leur maintien dans une structure a un coût, autant écologique (terres rares, composants non recyclables et énergie) qu'économique (coûts en temps humain liés à la pérennisation des formats, des structures de stockage et à l'analyse des données), et nombre de bases de données sont abandonnées faute de moyens pour assurer leur exploitation. Dans un environnement toujours plus compétitif, s'intéresser à une gestion raisonnée et responsable des données semble incontournable.

Protection et partage des données

L'omniprésence des données bouleverse depuis deux décennies les équilibres traditionnels des entreprises, car ce ne sont plus les actifs matériels d'une entreprise qui font sa valeur, mais ses actifs *immatériels* (Negri & Vercellone, 2008) [1] : "Le phénomène clé n'est plus l'accumulation de capital fixe, mais la capacité d'apprentissage et de création de la force de travail." La sauvegarde et l'analyse des données et des connaissances sont devenues des enjeux stratégiques pour les entreprises dans un environnement mondialisé ultra-compétitif.

Deux approches de gestion des données peuvent être distinguées : proposer un accès ouvert aux données et protéger les données. De plus en plus d'entreprises monétisent leurs données, qui vont ensuite être exploitées par d'autres entreprises qui vont leur attribuer du sens et les vendre à leur tour : il peut s'agir de données scientifiques, de fichiers client, de données sur le comportement ou le profil des clients, etc. Google, Facebook ou des grandes enseignes de vente en ligne en sont les exemples les plus connus. Une pratique récente consiste à publier massivement et gratuitement (sous une *licence ouverte* [2]) des données, de la part de collectivités et d'entreprises publiques ou privées : c'est l'*ouverture des données* (*Open Data* en anglais). Les bénéfices directs sont de valoriser des données et de favoriser les innovations. Une démarche similaire – en dehors du contexte des données ouvertes – sont les bases de données scientifiques accessibles à tous qui facilitent les découvertes scientifiques en réutilisant des données existantes pour de nouvelles recherches. C'est le cas par exemple de GenBank, une base de données génétiques ouverte dont les bénéfices apportés à la communauté ne sont plus à prouver (Benson *et al.*, 1999).

Les entreprises qui protègent leurs données peuvent mettre en place différentes stratégies qui s'avèrent complémentaires : soit conserver les données et l'expertise en interne, soit les rendre publiques tout en gardant les droits sur leur utilisation grâce aux outils légaux de protection de la propriété industrielle, tels que les marques, les dessins et modèles ou encore les brevets.

Le cas de l'industrie manufacturière

Dans le domaine de l'industrie manufacturière, il est primordial de protéger les données de conception et de production pour rester compétitif et garder une avance technologique. Les temps de développement des produits doivent être toujours plus courts pour faire face

1. Selon certains économistes, ces transformations sont à l'origine du *capitalisme cognitif*, une nouvelle forme de capitalisme en place depuis la fin des années 90 (Boutang, 2007).

2. Licence Libre de Diffusion (LLD) : garantit un libre accès aux données, et l'autorisation de les réutiliser sans restriction juridique ou financière.

à la concurrence, ce qui nécessite d'accéder aux bonnes informations au bon moment pour permettre une collaboration efficace entre toutes les équipes qui interviennent sur le produit : depuis l'apparition de l'*ingénierie collaborative*, il est devenu indispensable de structurer les données et de les partager entre tous les acteurs (travail multi-métiers, multi-équipes, multi-sites) (Belkadi *et al.*, 2010). Cela permet de réduire les temps de développements et donc d'améliorer la compétitivité d'une entreprise.

La fin des années 80 ont vu l'émergence de systèmes de gestion de bases de données pour gérer les informations du produit, appelés PDM (pour *Product Data Management* ou SGDT pour Système de Gestion des Données Techniques en français) (Liu & Xu, 2001). Ces systèmes permettent d'organiser les données, en particulier les données techniques et CAO (pour Conception Assistée par Ordinateur). Les SGDT ont progressivement évolué jusqu'aux systèmes PLM (pour *Product Lifecycle Management*) qui gèrent selon un modèle métier les données partagées entre les acteurs, les processus et les organisations à toutes les phases du cycle de vie d'un produit (Terzi *et al.*, 2010). Un nouveau produit n'est jamais conçu de zéro, et les systèmes PLM permettent d'assurer la traçabilité nécessaire à la gestion de la réutilisation des données produit (Assouroko *et al.*, 2014).

Exploration des données

Structurer des données et conserver leur traçabilité ne suffit pas à garantir qu'elles vont être correctement utilisées pour produire de l'information : il faut encore qu'elles soient rendues *accessibles*. L'accessibilité peut être abordée de deux façons : au sens strict par la capacité d'un utilisateur à consulter les données structurées, et d'un point de vue qualitatif par les fonctionnalités d'analyse offertes par l'interface de consultation, c'est-à-dire comment l'utilisateur peut agir sur les données pour les étudier et produire de l'information.

Les deux aspects de l'accès aux données peuvent être définis comme deux actions distinctes pour l'utilisateur. La première action est la *navigation*, qui consiste à parcourir la structure de données, comme par exemple localiser des données, traverser des relations entre des données via la traçabilité ou des liaisons connexes, ou encore calculer le chemin entre deux données. La seconde action est l'*exploration* qui permet l'analyse, en cherchant au-delà des données isolées l'information qui n'est pas encore connue.

Grâce aux modèles métiers du cycle de vie et aux systèmes PLM qui permettent de les gérer, les données de l'industrie manufacturière sont structurées et navigables. Pourtant rien n'est encore prévu pour l'exploration des données dans le PLM, malgré la complexité organisationnelle du cycle de vie des produits manufacturés (Troussier, 2010). La figure 1 illustre les relations entre les différentes structures de données techniques (Assouroko, 2012) : les exigences, les spécifications, l'arbre de modélisation CAO, le modèle de simulation numérique sont des données nécessaires à la conception d'un produit. L'abondance de relations entre ces données, et leur historisation (les versions successives sont conservées) les rend difficilement compréhensibles. Que se passe-t'il quand d'autres données comme celles associées à la fabrication du produit sont ajoutées ? Que se passe-t'il quand l'évolution des versions du produit est suivie ? La caractère

multidimensionnel des données rend ardues non seulement leur exploration mais aussi leur navigation.

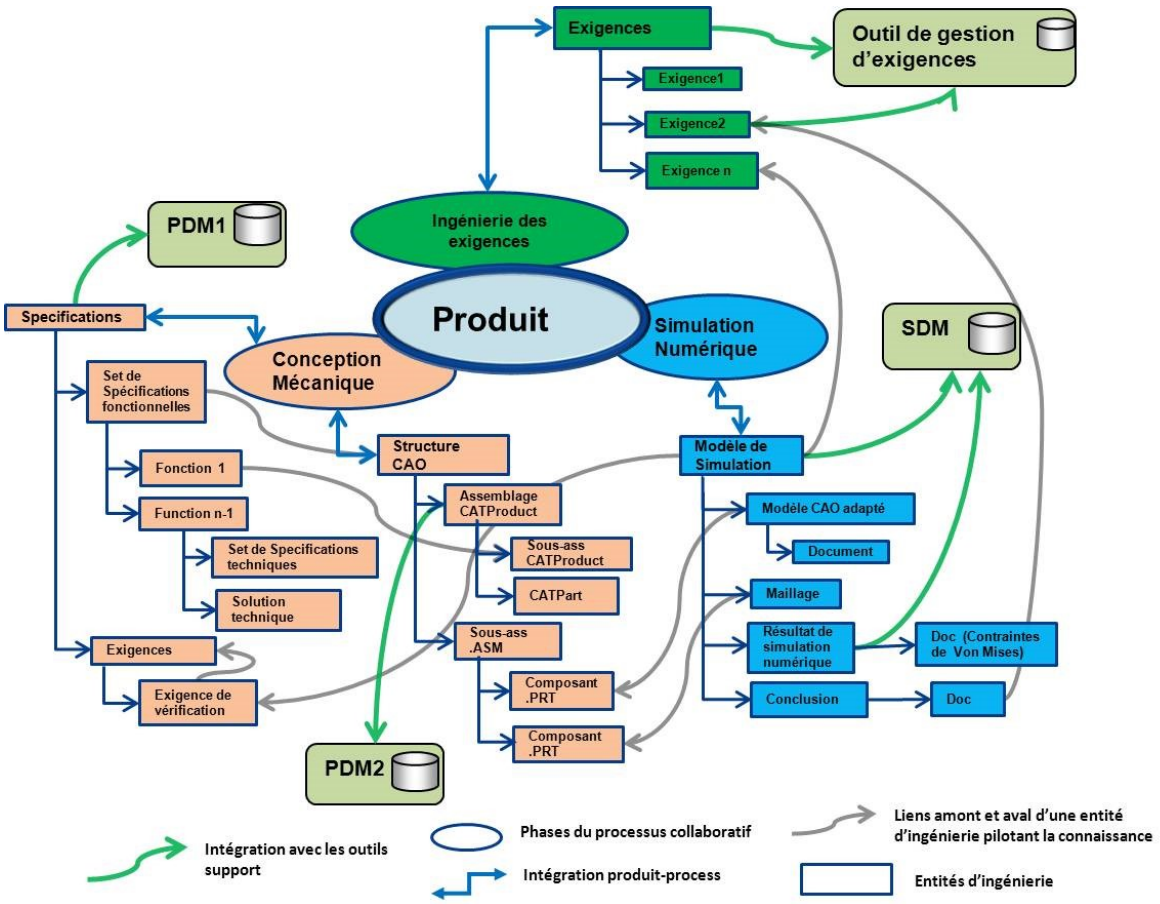


FIGURE 1 – Données du produit et leurs liens éventuels au sein du référentiel numérique commun (Assouroko, 2012)

Les interfaces des systèmes PLM actuels ne permettent pas une navigation aisée entre toutes les relations des données produit (cf annexe B), ce qui nuit à l'efficacité des systèmes : les interfaces ne sont pas conçues avec un souci d'ergonomie et de prise en compte de l'utilisateur final (*user-friendly*). Dans les conditions actuelles de compétitivité accrue, l'industrie manufacturière prend progressivement conscience de la perte de temps engendrée par les systèmes qui ne permettent ni une navigation efficace des données, ni une utilisation occasionnelle de leur interface.

Dans un monde où chacun veut accéder à l'information pertinente vite et sans effort, le moyen d'exploration le plus naturel semble être la *visualisation*. Pour Keller & Tergan (2005), la visualisation est une aide cognitive qui permet d'augmenter les capacités d'analyse humaines. En effet, la visualisation est un moyen efficace de présenter des informations complexes à un public profane, comme en atteste la multiplication des infographies dans le domaine du journalisme (data journalism) ou de la vulgarisation scientifique qui permettent de comprendre rapidement des informations complexes et d'interagir avec les données. Les nouvelles généra-

tions d'utilisateurs qui arrivent sur le marché du travail jouent aisément avec les codes du web et de la visualisation des données. Ils sont habitués à manipuler des systèmes intuitifs et interactifs qui permettent aussi bien à l'utilisateur quotidien qu'occasionnel de naviguer dans des jeux de données. Ces éléments doivent être pris en compte pour concevoir des interfaces d'exploration plus adaptées aux nouveaux enjeux du monde actuel.

Orientation de la thèse

La génération croissante de données numériques hétérogènes dans le monde bouleverse la recherche et la construction d'informations pertinentes dans tous les domaines. Cette transformation impacte les rapports entre les actifs matériels et immatériels, en particulier pour l'industrie manufacturière qui a développé les systèmes de gestion des données PLM qui permettent de gérer l'ensemble des données du produit tout au long de son cycle de vie et entre tous les acteurs. Cependant, l'accès – et donc l'exploitation de données structurées – demeure un problème, car posséder des données sans pouvoir les explorer et les enrichir est stérile. Une visualisation construite des données semble être une piste intéressante dans le contexte actuel car elle permet de faciliter l'exploration de données complexes.

Dans notre thèse, nous nous intéresserons à l'exploitation et la valorisation de **données hétérogènes**. Notre travail de recherche portera à la fois sur **la gestion et la visualisation des données en vue de leur réutilisation**, puisque la valeur d'une donnée ne se limite pas seulement à sa structuration mais également à la façon dont on y accède.

De nombreux domaines partagent les mêmes problématiques de structuration, d'analyse, de pérennisation et d'accès à des données hétérogènes. Dans cette thèse nous choisissons **la recherche en neuroimagerie** comme domaine d'application, car elle présente les caractéristiques qui nous intéressent : forte hétérogénéité des données renforcée par la pluridisciplinarité des études, nécessité du partage et de la réutilisation des données dont la génération a un coût financier non négligeable, et analyse visuelle des données incontournable. Depuis vingt-cinq ans, la communauté a pris progressivement conscience de l'importance d'une gestion durable de ses données, et les travaux qui ont déjà été menés permettront de constituer une base solide à notre recherche.

Une attention particulière sera portée sur la gestion et la visualisation de **données multidimensionnelles dynamiques**. Le monde évolue constamment et les données numériques produites ne sont pas statiques, quelles évoluent dans le temps ou en fonction de paramètres (dimensions d'évolution). Pourtant, lors de l'exploration de données multidimensionnelles, l'analyse porte souvent sur des données réduites – un point dans le temps, une combinaison de paramètres ou l'agrégation des données –, ce qui ne permet pas de comprendre la dynamique des données ou l'interaction entre les paramètres.

L'originalité de notre positionnement est de réutiliser des concepts développés dans un domaine pour les appliquer à un autre domaine qui présente des besoins similaires. En effet, les domaines de l'industrie manufacturière et de la neuroimagerie ont suffisamment de caractéristiques communes pour nous laisser penser que **les systèmes de gestion du cycle de vie des**

données produit (PLM) développés par l'industrie manufacturière peuvent être appliqués à la gestion des données en neuroimagerie.

Le manuscrit est organisé en huit chapitres :

Chapitre 1 – présentation du cadre des travaux de recherche, du positionnement scientifique de la thèse et de la méthode utilisée.

Chapitre 2 – étude de la gestion des données hétérogènes pour les domaines de l'industrie manufacturière et de la neuroimagerie : les modèles de données et les solutions de gestion existantes sont discutées pour les deux domaines.

Chapitre 3 – état de l'art de la théorie des graphes et des techniques en analyse et visualisation de graphes. Nous nous intéressons particulièrement à la visualisation de graphes dynamiques.

Chapitre 4 – modélisation d'une structure de données pour la gestion des données en neuroimagerie qui répond aux trois axes principaux identifiés : provenance, hétérogénéité et flexibilité.

Chapitre 5 – définition d'une représentation sous forme de graphe pour stocker et manipuler des données multidimensionnelles et dynamiques.

Chapitre 6 – conception d'une méthode de visualisation interactive qui facilite l'exploration de données multidimensionnelles dynamiques.

Chapitre 7 – application des propositions à l'étude de la connectivité fonctionnelle cérébrale au sein du laboratoire GIN (Groupe d'Imagerie Neurofonctionnelle, CNRS CEA Université de Bordeaux).

Chapitre 8 – conclusion des travaux, discussion des résultats et présentation des pistes de recherche futures.

Chapitre 1

Cadre et positionnement scientifique de la thèse

Dans ce chapitre sont présentés le cadre et les questions de recherche traitées dans notre thèse. Les problèmes actuellement rencontrés par le domaine de la neuroimagerie en analyse dynamique de données hétérogènes multidimensionnelles et en gestion de données sont présentés. Puis les bénéfices de l'usage de la visualisation de graphes pour explorer les données complexes sont introduits, ainsi que leurs limites Pour finir, nous nous situons par rapport à ce cadre général et formulons des hypothèses. Le positionnement scientifique présenté dans ce chapitre guidera l'analyse de l'existant sur les deux principaux domaines de recherche identifiés (chapitre 2 et 3).

Sommaire

1.1 Besoins actuels en neuroimagerie

1.1.1 Un domaine pluridisciplinaire

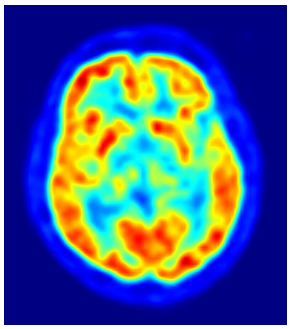
Le domaine de la neuroimagerie étudie le fonctionnement du cerveau à l'aide de techniques issues de l'imagerie médicale. Les années 90 ont vu l'apparition de techniques d'imagerie numérique tridimensionnelle et quadridimensionnelle, permettant d'étudier le cerveau de façon non invasive et *in vivo*, ce qui a constitué une vrai révolution pour l'acquisition des données. Parmi ces techniques :

- TEP (Tomoscintigraphie par Émission de Positrons ou PET pour Position Emission Tomography en anglais) : l'appareil mesure en trois dimensions les variations de débit sanguin cérébral régional grâce à la localisation des photons générés par la désintégration de positrons issus d'un produit radioactif injecté au sujet. Cette technique peut être utilisée sur tout le corps, mais est souvent ciblée sur un organe en particulier. (voir la figure 1.1-a pour un exemple d'image obtenue avec cette technique)
- EEG (Électroencéphalographie) et MEG (Magnétoencéphalographie) : naturellement les neurones du cerveau émettent une activité électrique, que ces techniques mesurent. La technique EEG mesure l'activité électrique du cerveau à l'aide d'électrodes placées directement sur le cuir chevelu, tandis que les magnétomètres permettent la mesure des champs magnétiques induits par cette activité électrique.
- IRM (Imagerie par Résonance Magnétique, ou MRI pour Magnetic Resonance Imaging en anglais) : un aimant supra-conducteur produit un champ magnétique puissant. L'application de champs magnétiques plus faibles va exciter les atomes d'hydrogènes libres qui seront mesurés par le signal électromagnétique qu'ils émettent. Cette technique est souvent ciblée sur un organe en particulier. (voir la figure 1.1 pour des exemples d'images obtenues avec cette technique)

Les techniques d'imagerie ont beaucoup évolué depuis leur création, et des chercheurs continuent de travailler sur la mise au point de nouvelles techniques permettant d'apporter toujours plus de précision aux mesures effectuées. Elles nécessitent autant de protocoles de mesures que de matériels d'acquisitions, et davantage encore de méthodes d'analyses, ce qui complexifie le travail des chercheurs.

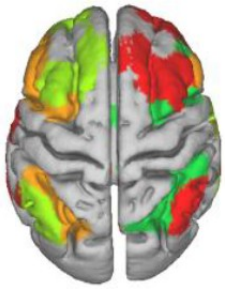
Le cerveau reste encore aujourd'hui un des organes les moins bien connus du corps humain, et les chercheurs sont loin d'avoir percé tous ses mystères. Le cerveau est principalement constitué de deux types de cellules : les neurones (voir figure 1.2-a) qui traitent les informations nerveuses et les cellules gliales qui assureraient le support métabolique. Les neurones sont reliés entre eux par des axones et permettent la propagation d'informations sur de longues distances.

La neuroimagerie *structurelle* examine et quantifie l'anatomie cérébrale, tandis que la neuroimagerie *fonctionnelle* enregistre les changements locaux de l'activité cérébrale quand les sujets effectuent des tâches cognitives, comme par exemple la génération de noms ou de verbes, l'exécution d'un calcul arithmétique, ou simplement restent à l'état de repos conscient, n'effectuant aucune tâche en particulier. Depuis une vingtaine d'années, la ségrégation est considérée comme principe de l'organisation fonctionnelle du cerveau (Friston *et al.*, 1994). Ce principe suggère que



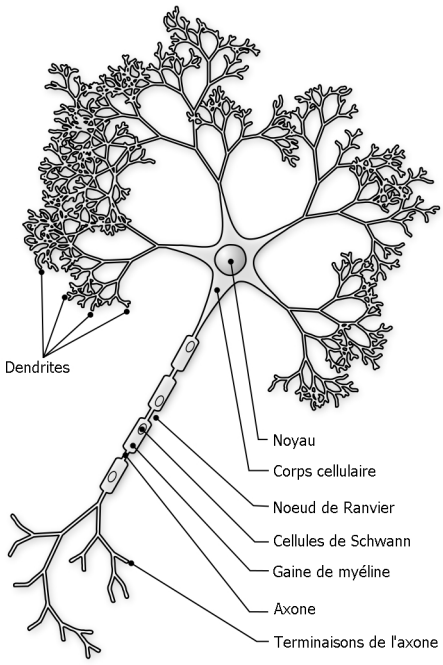
a) TEP

b) IRM structurelle

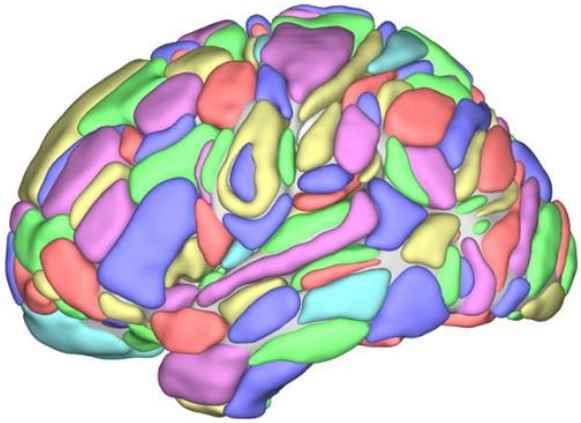


c) IRM fonctionnelle (Naveau, 2012)

FIGURE 1.1 – Visualisation du cerveau humain à l'aide de différentes techniques d'imagerie.



a) Neurone (Nicolas Rougier, GFDL 2007)



b) Atlas de cerveau (Joliot *et al.*, 2015)

FIGURE 1.2 – Deux échelles d'étude du cerveau humain.

certaines régions cérébrales sont spécialisées pour des processus cognitifs (moteurs ou sensoriels par exemple) et que leur collaboration, appelée intégration fonctionnelle, permet de supporter des fonctions cognitives particulières. L'étude de l'intégration fonctionnelle cérébrale a conduit à une représentation de l'organisation structurelle et fonctionnelle du cerveau sous la forme d'un système complexe (Bullmore & Sporns, 2009) : la segmentation du cerveau est définie sous la forme d'un atlas (voir figure 1.2-b) et les signaux fonctionnels enregistrés entre des zones distantes du cerveau sont analysés. L'étude de la connectivité cérébrale structurelle est appelée *connectomique* (*connectomics* en anglais) et inclut l'étude des réseaux cérébraux par la théorie des graphes.

La technique IRM est largement utilisé en neuroimagerie structurelle et fonctionnelle. Des exemples d'images IRM obtenues sur des cerveaux humains sont données en figure 1.1. Le nombre et la complexité des outils de traitement des données a augmenté de façon spectaculaire depuis les années 90 (Ferguson *et al.*, 2014; Gomez-Marin *et al.*, 2014; Van Horn & Toga, 2014). Les traitements des données de neuroimagerie sont organisés en workflow, le plus souvent exécutés sur des grilles de calcul à cause de leur coût computationnel élevé.

Les chaînes de traitement sont complexes et nécessitent une expertise métier conséquente qui est souvent répartie sur plusieurs membres d'un groupe de recherche. De façon générale, l'étude de la structure et du fonctionnement du cerveau de sujets, grâce aux méthodes d'imagerie actuelles, requière une collaboration active entre des spécialistes de différents domaines, tels que la physique, la médecine, la psychologie, l'informatique, les mathématiques et l'ingénierie, parmi d'autres. De plus en plus d'études en neuroimagerie combinent les données d'imagerie cérébrale avec les influences génomiques, les interventions pharmacologiques ou les conséquences d'un traitement médical (Van Horn & Toga, 2009). Pour Goble & Stevens (2008), la complexité inhérente aux données biologiques reste encore aujourd'hui un défi pour les chercheurs en imagerie biomédicale, car ils doivent gérer des données qui présentant des modalités différentes, des origines variées, et qui ont été traitées et analysées de différentes façons. Pour cette raison, Van Horn *et al.* (2001) qualifie la neuroimagerie appliquée à l'humain de domaine *pluridisciplinaire* "par sa nature même" [1].

La recherche en **neuroimagerie** étudie le cerveau à l'aide de multiples techniques d'imagerie médicale. En particulier, l'usage de la technique **IRM** est répandu pour l'**étude fonctionnelle du cerveau**.
La neuroimagerie est un domaine **pluridisciplinaire**, multi-méthodes et multi-plateformes.

1.1.2 Réutilisation des données en neuroimagerie

Des *cohortes* de sujets de grande dimension sont requises pour aborder des problématiques pluridisciplinaires telle que l'influence que les gènes exercent sur la structure et la fonction du cerveau – à la fois chez les sujets sains et malades, durant le vieillissement ou la croissance.

1. Traduction littérale depuis Van Horn *et al.* (2001) : "by its very nature, neuroimagery is a multidisciplinary endeavor".

De façon plus générale, de grandes cohortes sont nécessaires pour valider les résultats d'une recherche de façon statistique et dessiner ainsi des inférences fiables.

Les coûts financiers et humains, ainsi que les difficultés engendrées par des études de recherche en neuroimagerie sont élevés : les scanners d'acquisition IRM restent chers, et assurer le financement, la validation des protocoles expérimentaux et l'acquisition de données sur un grand nombre de sujets demandent du temps et des efforts. Par conséquent, seules les grandes structures de recherche et les projets collaboratifs au niveau national ou international peuvent accéder aux ressources et équipements nécessaires. Les analyses, à cheval sur plusieurs disciplines, deviennent de plus en plus diversifiées et complexes, et les groupes de recherche isolés ne détiennent pas toujours toutes les compétences nécessaires, ce qui les incite à collaborer. Les quinze dernières années ont vu l'émergence de ces projets à grande échelle qui impliquent plusieurs groupes de recherche sur différents sites géographiques et qui combinent plusieurs domaines comme l'imagerie, la psychologie et la génétique. Pour (Buckow *et al.*, 2014), les études de neuroimagerie à grande échelle posent non seulement des problèmes techniques, mais aussi sociaux, à cause du renouvellement des personnels et des problématiques de partage.

De nouvelles données sont collectées et publiées en permanence, sans être exploitées à leur potentiel maximal, et restent souvent inexploitées dans les systèmes de gestion des données des laboratoires. Pour Yarkoni *et al.* (2010), les chercheurs en neuroscience devraient orienter leur stratégie de recherche vers des études de synthèse à partir de données existantes, plutôt que de chercher à produire de nouvelles données. Ce type d'études est appelé *méta-analyses* et permet, toujours selon Yarkoni *et al.* (2010) de tester et de développer de nouvelles hypothèses, de vérifier la reproductibilité des résultats sur plusieurs laboratoires et de trouver des consensus par recoupement sur l'ensemble de la communauté, comme par exemple sur la segmentation du cerveau en *régions d'intérêt* (*ROI* pour *Regions Of Interest* en anglais) en neuroimagerie fonctionnelle par IRM.

Les technologies d'imagerie évoluent à un rythme qui garantit désormais que les jeux de données nouvellement acquis ne seront pas obsolètes avant plusieurs années. Cela permet la réutilisation de données pour des études longitudinales ou d'autres types d'analyses. Dans une *étude longitudinale*, une cohorte de sujets passe les examens deux fois - ou plus - à intervalles de plusieurs années, et le même protocole d'acquisition des données doit être utilisé. Les objectifs de ces études sont par exemple d'analyser l'évolution de fonctions cérébrales spécifiques ou de biomarqueurs structurels lorsque les sujets prennent de l'âge. De nombreuses études dans les pays occidentaux portent sur le vieillissement du cerveau et les dysfonctionnements associés, tels que la maladie d'Alzheimer (Weiner *et al.*, 2013).

Les chercheurs en neuroimagerie ont donc un intérêt à réutiliser les données et à faire en sorte que cette réutilisation soit facilitée. Savoir ce qui a été fait à chaque étape de traitement d'une donnée est la clé de sa compréhension par quiconque souhaiterait la *réutiliser* pour faire d'autres analyses. Cette information est appelée *provenance* et représente l'origine et l'historique d'une donnée (Simmhan *et al.*, 2005). Pour MacKenzie-Graham *et al.* (2008), la provenance des données est cruciale pour assurer la qualité, la précision, la reproductibilité et la réutilisation

de résultats d'études en neuroimagerie.

Par conséquent, les chercheurs doivent s'intéresser aux moyens de rendre leurs données accessibles et réutilisables par la communauté scientifique, permettant par exemple la méta-analyse précise de résultats antérieurs. Pour (Walter *et al.*, 2010) les chercheurs en neuroimagerie pourraient tirer un grand bénéfice d'outils innovants et de méthodes permettant la requête, l'analyse et le croisement de sources de données complexes, hétérogènes et à grande échelle. En d'autres termes, le domaine gagnerait à se doter d'un système intégré de gestion et de partage des données et de leur provenance pour réduire les coûts, gagner du temps et produire des analyses plus complexes.

Les traitements des données en neuroimagerie sont complexes et nécessitent de nombreuses étapes auxquelles sont associées autant de paramètres et de logiciels. La figure 1.3 présente un workflow Nipype [2] de traitement de données en neuroimagerie, qui permet de réaliser les premiers traitements sur des images nouvellement acquises, afin de les rendre exploitables. La représentation graphique dans le logiciel Tulip [3] facilite la compréhension des relations complexes entre les étapes du calcul.

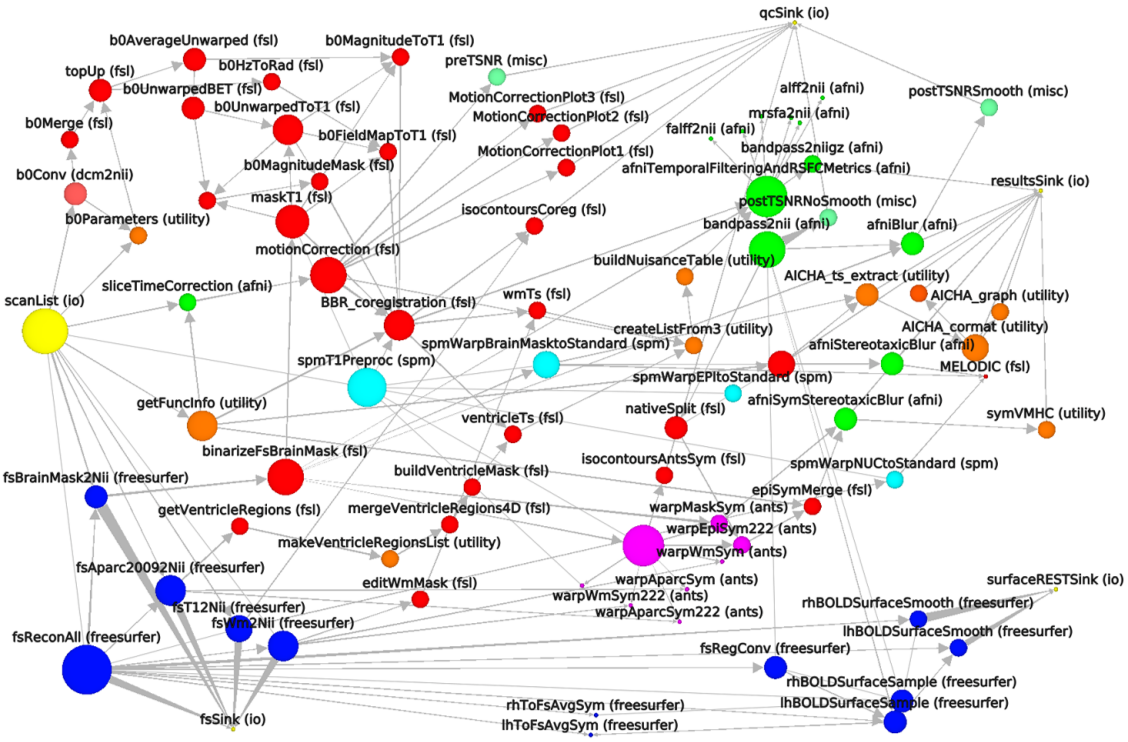


FIGURE 1.3 – Workflow Nipype d'un traitement d'images visualisé dans le logiciel Tulip. Les briques de calcul sont reliées entre elles par les flux entrants de données. Les couleurs des nœuds représentent les logiciels d'exécution de chaque brique : FSL=rouge, SPM=bleu, AFNI=vert, ANTS=rose, code isolé=orange, Freesurfer=bleu foncé, l'entrée Nipype=jaune. (réalisé par Pierre-Yves Hervé, 2015)

2. Framework extensible pour le traitement des données en neuroimagerie (Gorgolewski *et al.*, 2011).
3. Logiciel de visualisation de graphes développé par le Labri (Bordeaux, France) (Auber, 2004).

La **quantité de données à gérer** par les chercheurs pour mener à bien une étude a considérablement augmenté durant les vingt dernières années.
Pour réduire les coûts financiers et humains des études portant sur de larges cohortes de sujets, il apparait indispensable de **réutiliser au maximum les données d'études antérieures**.
Pour qu'une donnée soit réutilisable, **il faut que l'utilisateur puisse connaître sa provenance complète**.
Pour favoriser les nouvelles découvertes scientifiques, la neuroimagerie doit idéalement se doter d'un **système de gestion et de partage des données et de leur provenance** adapté.

1.1.3 Exploration de la connectivité fonctionnelle cérébrale

Bullmore & Sporns (2009) définissent la connectivité fonctionnelle comme la dépendance ou l'association statistique entre les éléments d'un réseau. Dans le cas de l'étude du cerveau, celui-ci est segmenté en régions dont les associations sont mesurées.

L'étude de la connectivité fonctionnelle permet de comprendre quelles régions sont actives en même temps lors de la réalisation d'une tâche. Cette méthode est utilisée pour améliorer la compréhension générale des mécanismes du cerveau, mais également pour l'étude et le diagnostique de pathologies neuro-dégénératives, telles qu'alzheimer ou la schizophrénie. Dans ce dernier cas, les données de connectivité fonctionnelle sont comparées entre les individus, comme l'illustre la figure 1.4.

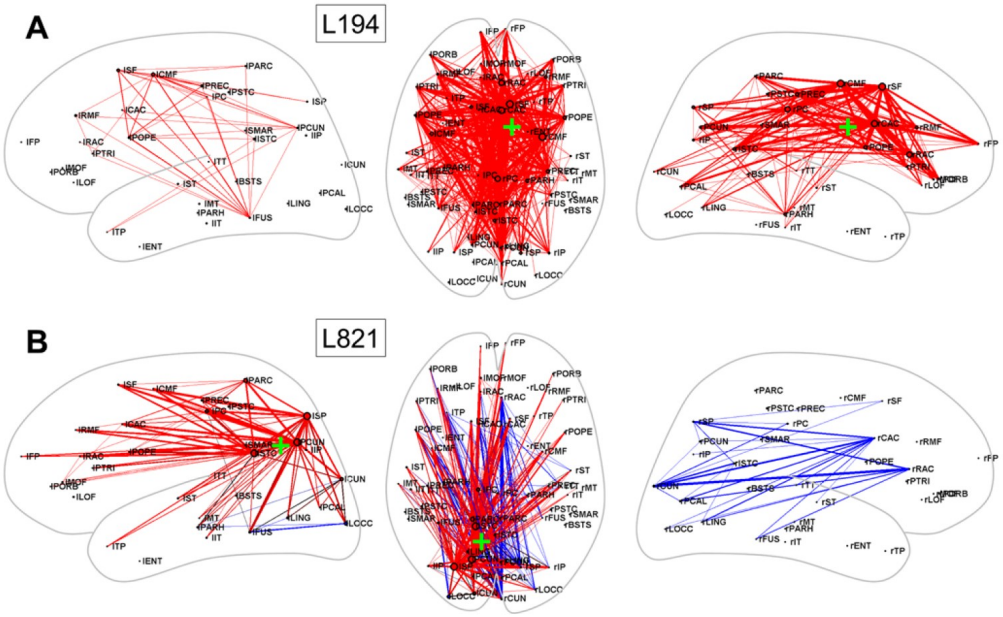


FIGURE 1.4 – Changements de la connectivité fonctionnelle à l'état de repos conscient suite à des lésions cérébrales simulées (Alstott *et al.*, 2009). Le centre de la région lésée est indiqué par le symbole "+" vert. Les arêtes rouges (bleues) indiquent une baisse (augmentation) de la corrélation entre les régions du cerveau lésé, vis à vis d'un cerveau normal.

La connectivité fonctionnelle peut être étudiée dans le cadre d'études longitudinales, pour comprendre l'évolution des relations entre les régions du cerveau. Par exemple dans des études dédiées à la compréhension de l'évolution de la maladie d'alzheimer avec l'âge (Weiner *et al.*, 2010), ou qui s'intéressent au développement cérébral chez des sujets pendant les deux premières années de la vie : à quatre semaines, un an et deux ans (Knickmeyer *et al.*, 2008). De plus en plus d'étude couplent les résultats de l'analyse de connectivité avec d'autres modalités, comme par exemple des variations génétiques (Papenberg *et al.*, 2015).

Malgré un intérêt croissant de la communauté pour l'exploration de la connectivité fonctionnelle cérébrale dynamique (Hutchison *et al.*, 2013), beaucoup des travaux ne propose qu'une statique de la connectivité fonctionnelle et il reste encore beaucoup à faire pour que la neuroimagerie dispose de méthodes et d'outils efficaces pour explorer les relations entre les régions du cerveau.

La connectivité fonctionnelle cérébrale peut être étudiée suivant plusieurs dimensions, telles que le temps d'acquisition, le temps entre les acquisitions ou encore les sujets.

1.2 Exploration de relations par la visualisation de graphes dynamiques

1.2.1 La visualisation, une aide cognitive

Les êtres humains n'ont pas attendu Napoléon et sa citation "un bon croquis vaut mieux qu'un long discours" pour s'apercevoir de l'intérêt de visualiser des informations. Pour Shneiderman (1996), explorer un ensemble d'information devient particulièrement difficile à mesure que son volume augmente : quand un texte prend la taille d'un livre ou d'une bibliothèque, comment localiser et naviguer entre toutes les entrées ?

Dans l'introduction du livre *Visualizing Knowledge and Information*, Keller & Tergan (2005) présentent les apports de la visualisation identifiés dans les travaux de recherche depuis les années 70. Il est communément montré que la visualisation est un dispositif de cognition externalisé, c'est-à-dire qu'elle permet d'améliorer les capacités cognitives humaines. La visualisation permettrait de surmonter les limites naturelles de la mémoire de travail, à la fois au niveau de la capacité et de la durée du stockage. Les représentations externes permettent de réduire la charge cognitive d'un individu (Sweller, 1994) et d'améliorer ses capacités à exécuter des tâches cognitives complexes (Larkin, 1981). Pour Ware (2012), "power of a visualization comes from the fact that it is possible to have a far more complex concept structure represented externally in a visual display than can be held in visual and verbal working memories" [4].

Par conséquent, la visualisation se montre particulièrement utile pour certains types de tâches :

4. Traduction : "le pouvoir de la visualisation vient du fait qu'il devient possible de représenter extérieurement une structure bien plus complexe que ce que peuvent contenir les mémoires de travail verbale et visuelle" (Ware, 2012).

– Rechercher et traiter des données structurées (Wiegmann *et al.*, 1992),
– Comprendre des relations abstraites entre éléments (Cox, 1999),
– Détecter, comprendre et identifier des motifs inattendus (Bezerianos *et al.*, 2010)

Pour Zhang & Norman (1994), les bénéfices de la visualisation proviennent de la coordination entre les représentation internes de l'individu et les représentations externes utilisées dans la visualisation, c'est-à-dire de l'utilisation d'une représentation distribuée.

La **visualisation est une aide cognitive** qui permet de détecter, comprendre et identifier des motifs inattendus dans des **jeux de données complexes de grande taille**.

1.2.2 Tâches de visualisation

Le domaine de la *visualisation d'information* (ou *infovis* pour *information visualisation*) étudie les moyens permettant une communication visuelle efficace. deux axes : les techniques de représentation de l'information, et les interfaces permettant la manipulation visuelle des données.

Pour visualiser efficacement de grands ensembles de données, il convient de procéder par étapes. Shneiderman (1996) est le premier à proposer un "mantra" [5] de l'exploration visuelle d'informations : "Overview first, zoom and filter, then details-on-demand", soit "D'abord une vue d'ensemble, zoomer et filtrer, puis détailler à la demande". Les techniques de visualisation doivent être utilisées en fonction des objectifs à atteindre, qui peuvent être distingués en trois grandes catégories : consultation (informatif), analyse (révélateur) et construction (découverte).

Le mantra de l'exploration visuelle de Shneiderman (1996) suppose l'*interaction* entre l'utilisateur et les données à visualiser. Un dispositif d'interaction permet à l'utilisateur d'agir manuellement sur la visualisation de façon directe en fonction de ses objectifs d'exploration, tandis qu'une *visualisation dynamique* est un changement automatique de la visualisation (Keim *et al.*, 2002). Pour Ware (2005), l'interaction améliore les performances en visualisation en permettant à la fois les aller-retours entre analyse qualitative et observation quantitative, et le positionnement à différentes échelles.

La **visualisation d'information** est le domaine qui étudie les techniques permettant une communication visuelle efficace.
L'interaction facilite l'exploration de données complexes.

1.2.3 Le graphe comme modèle

Un graphe est un objet mathématique qui permet de représenter des problèmes, simples ou complexes. Le mot graphe vient du Grec "écrit", un graphe est un moyen de représenter le langage par les signes.

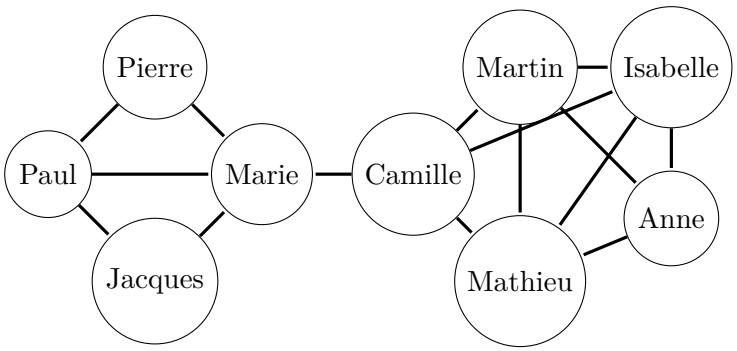
Dès les années 70, les bénéfices de la visualisation de données sous forme de graphes ont été mis en avant : Anscombe (1973) montre par exemple que des réseaux avec des mesures similaires

5. Terme utilisé par Shneiderman (1996) dans le sens d'une phrase répétitive qui reprend les concepts essentiels.

peuvent présenter des topologies différentes. La figure 1.5 présente des relations d'amitié (les arêtes) entre neuf personnes (les nœuds) sous les formes d'un tableau ou d'un graphe (représentation node-link). Au format tabulaire, il est difficile d'interpréter facilement les données, tandis qu'au format graphe, deux groupes distincts de personnes apparaissent, qui sont reliés grâce à l'amitié entre Marie et Camille.

Pierre	Paul
Anne	Martin
Marie	Camille
Camille	Martin
Paul	Jacques
Mathieu	Martin
Martin	Isabelle
Marie	Jacques
Pierre	Marie
Isabelle	Camille
Mathieu	Anne
Camille	Mathieu
Marie	Paul
Anne	Isabelle
Isabelle	Mathieu

a- format tabulaire



b- représentation sous forme de graphe (node-link)

FIGURE 1.5 – Relations d'amitié entre neuf personnes

Les graphes sont utilisés dans de nombreuses applications : l'analyse et l'optimisation de réseaux (sociaux, criminels, biologiques...), la modélisation conceptuelle (ontologies) ou encore les bases de données. Les *bases de données orientées graphe* (ou *bases graphes*) permettent de stocker des informations organisées dans une structure de données de type graphe (c'est-à-dire constituée de nœuds et d'arêtes). Un exemple connu de base graphe est Neo4j (Webber, 2012).

La théorie des graphes est l'étude mathématique des graphes en vue de répondre à des questions sur leur structure. Le domaine de la visualisation d'information développe des techniques spécifiques à la visualisation des graphes (*graph drawing*). La communauté est très active sur le sujet, comme en témoignent les publications dédiées à l'état de l'art au fil des années, telles que Di Battista *et al.* (1994) ou (Von Landesberger *et al.*, 2011).

Le graphe est un objet mathématique qui permet d'étudier des problèmes complexes.
Les graphes sont analysés à l'aide de la **théorie des graphes**.
Ils peuvent être visualisés grâce aux techniques développées dans le **domaine de la visualisation d'information**.

1.3 Synthèse du positionnement

1.3.1 Problématique

Le domaine de la neuroimagerie, multidisciplinaire par essence, doit faire face à de grandes quantités de données hétérogènes aux relations complexes et qui doivent être explorées pour permettre la réutilisation de données dans de nouveaux contextes. Ce constat nous amène à poser la problématique suivante :

Comment explorer les relations complexes entre ensembles de données hétérogènes ?

Cette problématique peut être découpée en deux principales sous-questions.

1.3.1.1 Questions de recherche

Pour permettre le partage et la réutilisation de données, il est nécessaire de conserver l'ensemble de leur provenance. L'exploration de la provenance implique que les données soient structurées.

1e question
Comment gérer les données hétérogènes et leur provenance ?

La visualisation est une aide cognitive à l'analyse de jeux de données. Les données multidimensionnelles et dynamiques du domaine de la neuroimagerie pourraient donc être explorées visuellement.

2e question
Comment visualiser les structures de données multidimensionnelles et dynamiques ?

Avant de concevoir une méthode de visualisation de données multidimensionnelles et dynamiques, il convient de définir leur représentation. Les problèmes soulevés sont donc déclinés en trois axes de recherche :

1. Faciliter la conservation de la provenance et structurer les données hétérogènes
2. Représenter des données multidimensionnelles dynamiques à explorer
3. Explorer visuellement des données

La figure 1.6 résume les articulations des problèmes et des axes de recherche présentées dans ce manuscrit.

1.3.1.2 Similarités entre les domaines de l'imagerie biomédicale et de l'industrie manufacturière

Nous avons choisi la neuroimagerie comme domaine d'application, cependant d'autres domaines présentent des besoins en exploration de données complexes. Nous avons vu dans l'intro-

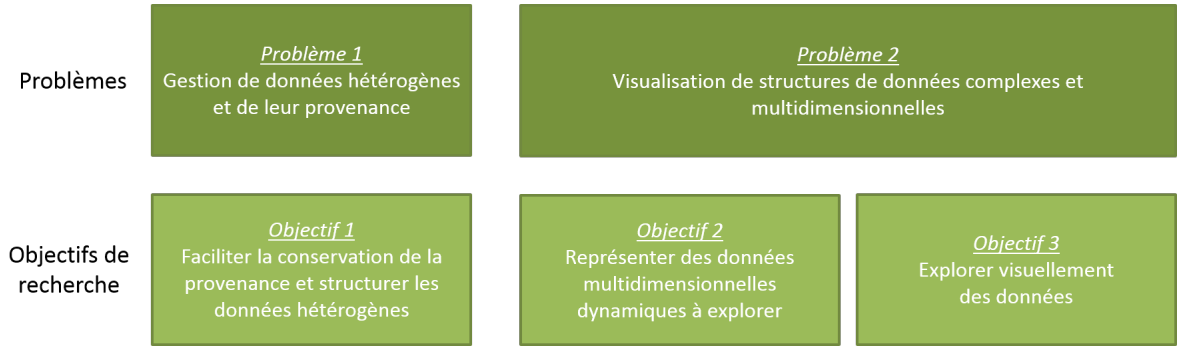


FIGURE 1.6 – Les deux problèmes déclinés en axes

duction que c'est le cas de l'industrie manufacturière. Dans cette sous-section, nous montrons que les domaines de l'imagerie médicale et de l'industrie manufacturière, a priori très éloignés, présentent des caractéristiques et des besoins similaires.

Avec la mondialisation, les entreprises manufacturières cherchent à développer des produits de meilleure qualité et avec un cycle de développement toujours plus court pour maitriser les coûts et rester compétitif. Les groupes de recherche évoluent également dans un environnement à la compétitivité croissante : les chercheurs doivent publier des articles scientifiques de la meilleure qualité afin d'obtenir des financements qui leur permettront de mener leurs recherches futures.

L'industrie manufacturière doit assurer le partage de données entre d'une part différents métiers comme la conception, la simulation ou la fabrication, et d'autre part des disciplines variées suivant la nature des produits : électricité, mécanique, fluides, informatique... Cette pluridisciplinarité est croissante avec l'augmentation des systèmes complexes (ou systèmes de systèmes), qui imposent davantage d'interactions entre les disciplines au moment de la conception du produit (Lefèvre *et al.*, 2014). Il a été montré dans la section 1.1.1 que la neuroimagerie est un domaine pluridisciplinaire, ce qui constitue un second point commun entre les deux domaines.

Dans un contexte d'ingénierie concurrente, il est nécessaire de faire coexister plusieurs sites géographiquement éloignés et plusieurs équipes dans chaque site, travaillant sur différentes étapes du cycle de vie du produit. Les équipes doivent pouvoir partager de façon efficace et en temps réel les données du produit, malgré un vocabulaire et des pratiques différentes. Les projets de recherche à grande échelle dans le domaine de la neuroimagerie présentent des caractéristiques similaires : plusieurs laboratoires, souvent éloignés géographiquement, doivent échanger des données et leur provenance avec précision pour mener à bien les analyses.

Un nouveau produit est rarement conçu de zéro : réutiliser des pièces ou des mécanismes de produits précédents est une nécessité pour réduire les temps développement d'un produit – et à plus forte raison de produits complexes. Il faut que les concepteurs puissent accéder facilement aux données de tout le catalogue de produits de l'entreprise pour éviter les redondances. Le domaine de la neuroimagerie est également fortement encouragé à réutiliser des données antérieures pour réduire les coûts et obtenir de meilleurs résultats statistiques.

Les domaines de l'imagerie biomédicale et de l'industrie manufacturière présentent des caractéristiques similaires : environnement compétitif, pluridisciplinarité, besoins en réutilisation et en partage des données.

Hypothèse

Les solutions de gestion des données développées pour l'industrie manufacturière peuvent être appliquées à la gestion des données en neuroimagerie.

Les propositions développées dans ce manuscrit pour le domaine de la neuroimagerie pourront être utiles pour le domaine de l'industrie manufacturière.

1.3.1.3 Intersection des domaines de recherche

Notre thèse s'inscrit dans des domaines de recherche pluridisciplinaires, autour de trois grands pôles : l'ingénierie de conception, l'informatique et l'imagerie biomédicale. L'objectif de cette thèse est de contribuer à améliorer l'exploration de données complexes entre ensembles de données hétérogènes dans le domaine de la neuroimagerie. L'originalité des travaux est de faire appel aux méthodes d'ingénierie collaborative qui existent déjà dans le domaine de l'industrie manufacturière pour faciliter la gestion des données de neuroimagerie.

Les propositions présentées dans ce manuscrit s'inscrivent dans les domaines de la gestion des données, de la visualisation d'information et en particulier de la visualisation des réseaux biologiques.

La figure 1.7 illustre les intersections des domaines de recherche et les domaines de contribution de la thèse par un diagramme ARC (pour Areas of Relevance and Contribution) (Blessing & Chakrabarti, 2009).

1.3.2 Environnement de la thèse

1.3.2.1 Un encadrement multi-partenaires

Notre démarche originale et multi-domaines nécessite la collaboration de plusieurs partenaires. La thèse s'inscrit dans un partenariat industrie-académie. Elle est financée par l'entreprise CADESIS et soutenue par l'ANRT (Agence Nationale de la Recherche et de la Technologie) dans le cadre d'une convention CIFRE. L'entreprise CADESIS apporte son expertise technique, tandis que l'encadrement scientifique est assuré par l'équipe SIM (Systèmes Intégrés en Mécanique) du laboratoire Roberval (UMR 7337 – CNRS Université de Technologie de Compiègne) et le laboratoire GIN (Groupe d'Imagerie Neurofonctionnelle, UMR 5296 – CNRS CEA Université de Bordeaux).

Chacun des partenaires contribue aux travaux présentés dans ce manuscrit du point de vue de leur domaines respectifs :

– CADESIS : ingénierie collaborative, conduite de projets de mise en place PLM, visualisation des informations et des connaissances.

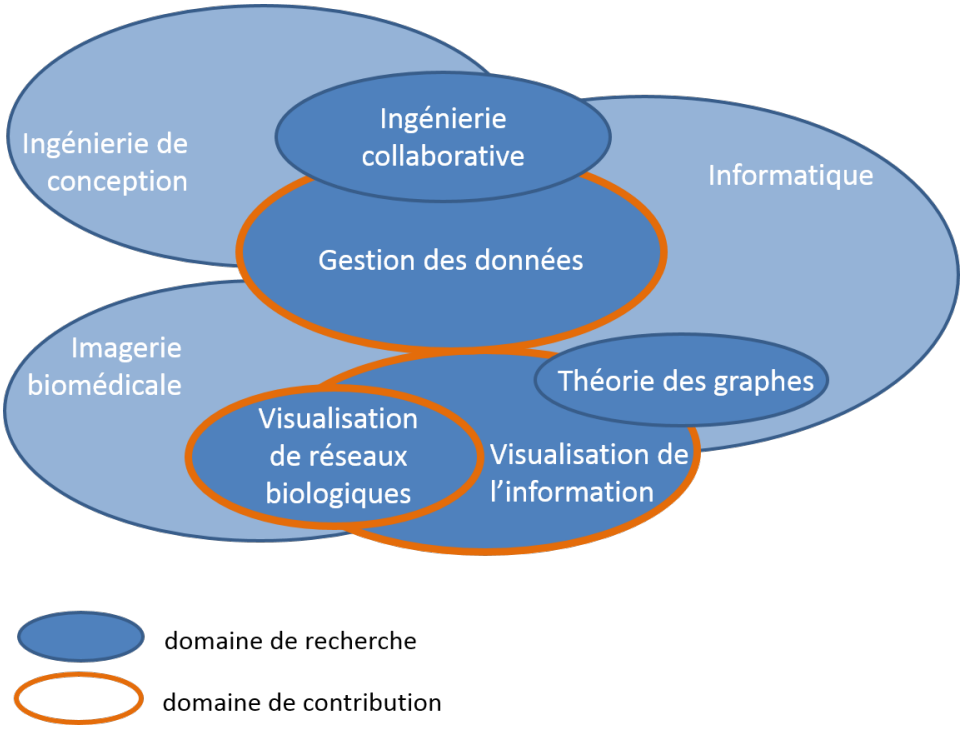


FIGURE 1.7 – Diagramme ARC des domaines de recherche de la thèse, d'après la méthode de Blessing & Chakrabarti (2009)

Les travaux sont menées au sein du pôle R&D de CADESIS qui entreprend un programme de recherche depuis dix ans dans le but d'améliorer l'offre auprès de ses clients en outils de capitalisation des connaissances autour des outils de gestion des données techniques. Les objectifs sont notamment de favoriser les échanges entre applications métier et les systèmes de gestion des données.

– Roberval : ingénierie collaborative (PLM), visualisation de graphes de relations.
  L'équipe SIM travaille sur la notion de système intégré, depuis le produit mécanique au système de production. Les principaux axes de recherche portent sur la définition de méthodes et d'outils d'ingénierie système avec un focus sur l'intégration entre expertises métiers et l'interopérabilité logicielle, et la conception de méthodes de développement de systèmes mécatroniques à l'aide de techniques innovantes.
– GIN : neuroimagerie, étude des réseaux biologiques
  Le GIN est une unité de recherche multidisciplinaire rassemblant des chercheurs des domaines de l'instrumentation pour l'imagerie médicale, de la médecine nucléaire et des neurosciences, du traitement du signal, de la psychiatrie et des neurosciences cognitives. Le GIN possède le savoir-faire pour toutes les techniques d'imagerie fonctionnelle (TEP, IRM, EEG, MEG) depuis l'élaboration des expériences jusqu'à l'analyse des données. Le GIN a aussi été pionner dans l'élaboration, la gestion et l'analyse de bases de données multimodales très grandes combinant des données d'imagerie, psychométriques et génétiques.

1.3.2.2 Le projet BIOMIST

En parallèle de cette thèse s'est déroulé le projet BIOMIST (BIO Medical research Imaging SemanTic data management, n° ANR-13-CORD-0007) qui est financé par l'ANR (Agence Nationale de la Recherche) dans le cadre de l'appel à projets CONTINT (Contenus Numériques et Interactions) dans la thématique "Des contenus aux connaissances et grandes masses de données".

Le consortium du projet est constitué de quatre partenaires : les trois partenaires déjà impliqués dans la thèse CIFRE et l'UTT avec l'Institut Charles Delaunay (ICD – UMR 6279). L'entreprise CADESIS est coordinateur du projet.

L'objectif est de fournir aux chercheurs utilisant l'imagerie biomédicale un système d'information efficace de façon à optimiser l'utilisation de leurs données dans le cadre d'activités de recherche incluant de larges groupes de sujets sur des périodes prolongées. Cela devra permettre la réutilisation de données produites en recherche clinique et fondamentale dans un contexte et pour un but autre que celui pour lequel elles avaient été acquises.

Le projet BIOMIST se concentre sur le domaine d'activité du laboratoire GIN de l'Université de Bordeaux : l'imagerie neurofonctionnelle. En dehors des données d'imagerie proprement dites (2D, 3D, 4D), l'objectif est de permettre la gestion de toutes les autres données nécessaires à la définition d'une étude, notamment les données démographiques, comportementales ainsi que des données génétiques. Le but n'est pas seulement de gérer et de tracer les documents d'une étude mais également les concepts utilisés par les chercheurs tels que paradigme de stimulations cognitives, tâches de traitements, définitions des études comportementales,...ainsi que toutes les relations qui peuvent exister entre ceux-ci.

Le projet BIOMIST propose des méthodologies et des outils qui permettront de gérer la complexité grandissante et la provenance des données de neuroimagerie, mais aussi leur usage, leurs différentes représentations et leur interprétation dans le domaine de la recherche neurofonctionnelle. Une infrastructure couramment utilisée en ingénierie industrielle pour couvrir les exigences de base de gestion des données BMI est mise en œuvre et adaptée aux besoins des chercheurs : il s'agit d'un outil PLM (Product Lifecycle Management). Afin de pallier aux limites de flexibilité que des systèmes PLM par rapport aux besoins en neuroimagerie, le projet BIOMIST propose d'utiliser des techniques de gestion des connaissances pour permettre une meilleure traçabilité et des possibilités de réutilisation des données dans un contexte d'avantage évolutif que celui de l'industrie, *i.e.* celui de la recherche. Par ailleurs, le projet BIOMIST développe et intègre des outils de visualisation et d'analyse qui permettront de faire des hypothèses, de découvrir intuitivement des motifs visuels et d'isoler des singularités structurelles, grâce à la modélisation des données sous forme de graphe. Ces graphes pourront être utilisées pour représenter les relations sémantiques ou les réseaux de connectivités du cerveau, qui sont une représentation spécifique aux études neurofonctionnelles.

La charge de travail au sein du projet est divisé en six lots :

- **WP1** : Gestion du projet et coordination
- **WP2** : PLM pour l'imagerie biomédicale
- **WP3** : Visualisation et comparaison de graphes
- **WP4** : Apports sémantiques pour la traçabilité et la réutilisation en recherche biomédicale
- **WP5** : Intégration et contrôle qualité
- **WP6** : Exploitation et dissémination des résultats

Les travaux réalisés dans le cadre de cette thèse sont validés au sein des lots WP2 et WP3 du projet BIOMIST.

Pour d'avantage d'informations, consulter le site du projet : `www.biomist.fr`.

1.3.3 Démarche d'élaboration des travaux de thèse

1.3.3.1 Méthode

Afin de répondre aux problèmes soulevés, nous organisons notre démarche en sept grandes étapes illustrées dans la figure 1.8 :

1. Étude des besoins en gestion des données dans le domaine de la neuroimagerie et état de l'art en matière de gestion des données du cycle de vie du produit. Étude du fonctionnement d'un laboratoire en gestion des données (interviews, étude du système de gestion des données en place), étude de cas et analyse des solutions de gestion des données du domaine.
2. État de l'art des techniques d'analyse topologique et de visualisation de graphes, en particulier les graphes multivariés et les graphes dynamiques.

   Nous segmentons la proposition en trois axes qui s'appuient sur les deux premières étapes :

3. Modélisation d'une structure de données pour la gestion des données en neuroimagerie qui répond aux trois axes principaux identifiés : provenance, hétérogénéité et flexibilité.
4. Définition d'une représentation sous forme de graphe pour stocker et manipuler des données multidimensionnelles et dynamiques.
5. Conception d'une méthode de visualisation interactives de graphes de données multidimensionnelles dynamiques qui alterne graphe composé réduit et small-multiples, tout en préservant la carte mentale de l'utilisateur.

   Nous finissons en validant les trois axes de la proposition :

6. Développement de prototypes – dans le cadre du groupe de recherche GIN et du projet BIOMIST – afin de prouver la faisabilité de nos propositions : il s'agira de mesurer l'écart entre la situation réelle atteinte et la situation idéale souhaitée au départ.
7. Une discussion des résultats et des pistes de recherches futures closent la thèse.

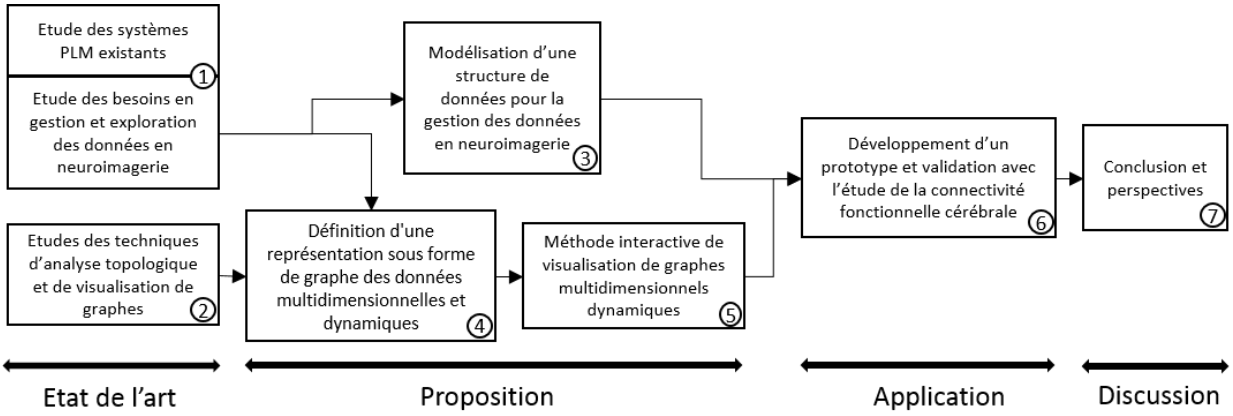


FIGURE 1.8 – Notre méthode de recherche organisée en sept grandes étapes

1.3.3.2 Structure du manuscrit

La structure du manuscrit, de l'introduction aux perspectives est présentée dans la figure 1.9. Dans un premier temps, nous présentons un état de l'art des domaines pour chacune de nos problématiques de recherche :

- la gestion de données dans l'industrie manufacturière, les concepts liés au Product Lifecycle Management et les systèmes PLM existants (chapitre 2).
- la gestion de données et l'analyse de réseaux cérébraux en neuroimagerie (chapitre 2).
- la théorie des graphes et les techniques de visualisation de graphes. Nous nous intéresserons particulièrement à la visualisation de graphes dynamiques (chapitre 3).

Ces deux chapitres nous permettent de nous positionner par rapport aux travaux existants et de formuler des hypothèses.

Les trois chapitres suivants présentent nos propositions pour résoudre les problèmes de recherche identifiés :

- le modèle de données BMI-LM (Bio-Medical Imaging – Lifecycle Management) qui structure les données en neuroimagerie autour de trois axes : provenance, hétérogénéité, flexibilité (chapitre 4) ;
- les graphes GMD (Graphe Multidimensionnel Dynamique) pour le stockage, l'analyse et la visualisation de données complexes multidimensionnelles et dynamiques (chapitre 5) ;
- la méthode OCL (Overview Constraint Layout) d'exploration de graphes GMD qui s'articule autour d'une réduction des données pour la mise en exergue des éléments constants du graphe, et d'une comparaison en contexte pour l'analyse des différences entre états du GMD (chapitre 6).

La mise en œuvre de nos propositions dans une prototype utilisé au sein du groupe de recherche GIN et dans le cadre du projet BIOMIST est détaillée dans le chapitre 7. Pour finir, une critique des travaux présentés dans ce manuscrit est proposée et des perspectives de recherche future sont données dans le chapitre 8.

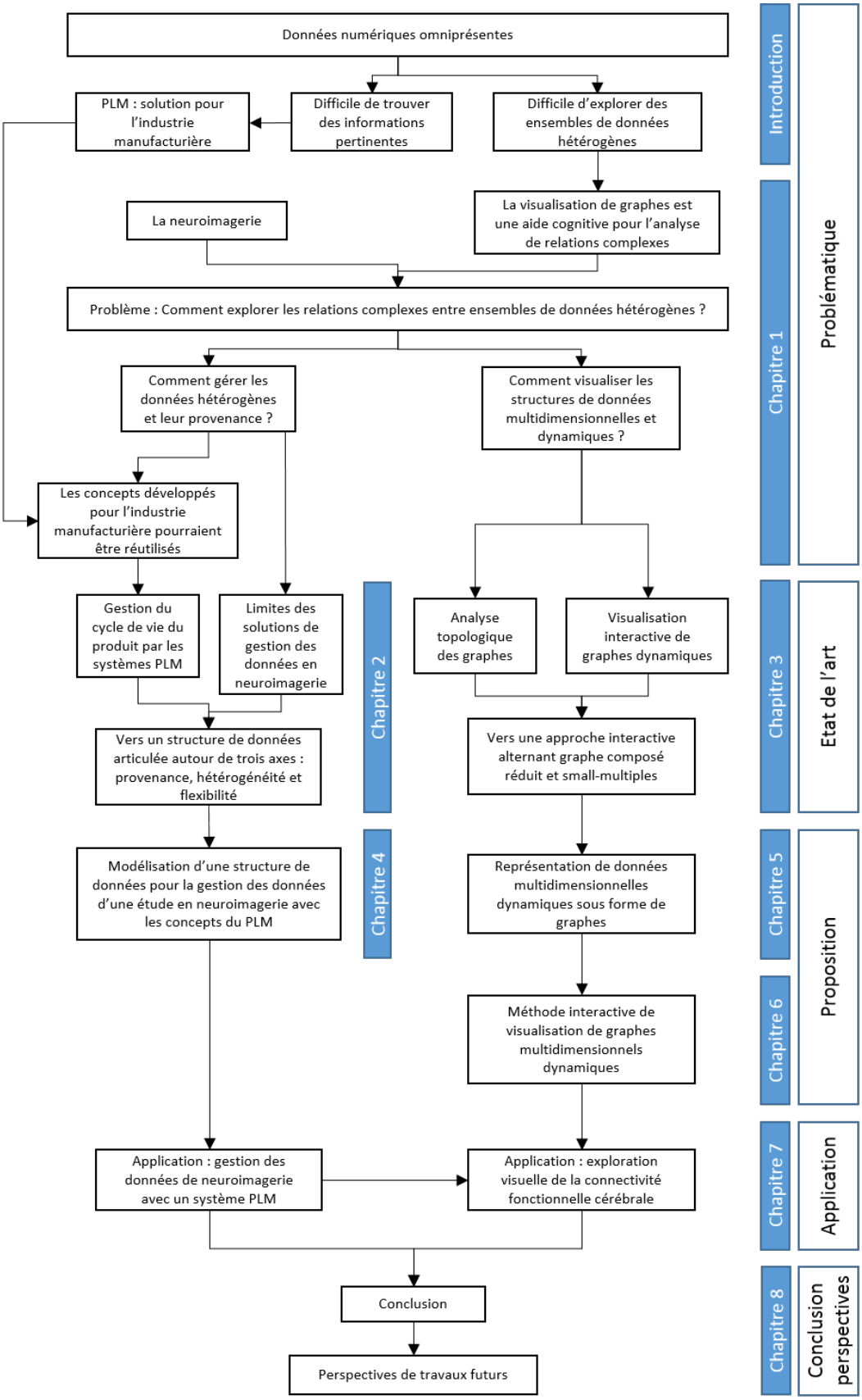


FIGURE 1.9 – Structure de la thèse

La figure 1.6 qui présente de façon synthétique les problèmes identifiés et les axes de recherche **sera reprise au fur et à mesure des chapitres pour guider le lecteur**.

Conclusion du chapitre 1

Ce chapitre a permis d'exposer les questions de recherche de la thèse, qui sont reprises dans la figure 1.6 :

- Comment gérer les données hétérogènes et leur provenance ?
- Comment visualiser les structures de données multidimensionnelles et dynamiques ?

De plus nous avons identifié les domaines de recherche et d'application de la thèse : ingénierie collaborative, gestion des données, visualisation de l'information, théorie des graphes et visualisation des réseaux biologiques (voir la figure 1.7).

Le prochain chapitre présente un état de l'art sur la gestion de données hétérogènes, appliquée au cycle de vie du produit et à la neuroimagerie.

Chapitre 2

Gestion de données hétérogènes

Le chapitre 1 a permis de mettre en évidence un premier problème "comment gérer des données hétérogènes pour garantir leur réutilisation et leur partage". Les caractéristiques et les besoins communs (partage, réutilisation, flexibilité) des domaines de la neuroimagerie et de l'industrie manufacturière ont été mis en évidence, ce qui nous a conduit à formuler l'hypothèse que les systèmes PLM, qui ont été conçus pour résoudre les problèmes en gestion des données de l'industrie manufacturière, pourraient aider à résoudre ceux en imagerie biomédicale.

Dans un premier temps, nous présentons les concepts liés à la gestion du cycle de vie des produits et l'état actuel des systèmes PLM (section 2.1). Ensuite nous étudions les caractéristiques des solutions de gestion des données développées pour le domaine de la neuroimagerie (section 2.2).

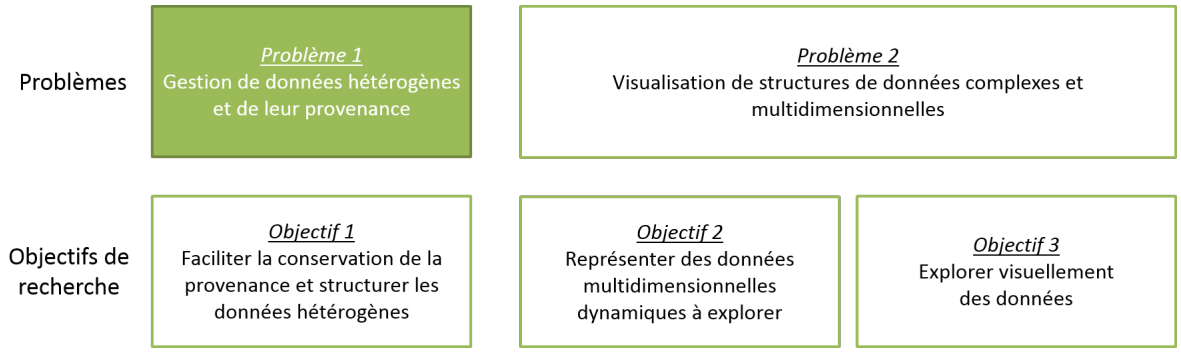


FIGURE 2.1 – Le problème 1 : Gestion de données hétérogènes pour la réutilisation et le partage.

Sommaire

2.1 Gestion des données produit

Dans un contexte de compétitivité mondiale croissante, les entreprises de l'industrie manufacturière sont confrontées à une multiplication des acteurs intervenants tout au long du cycle de vie des produits, répartis dans des équipes et sur des sites géographiquement éloignés. Au sein d'une *entreprise étendue* (*extended enterprise*), les fonctionnalités principales du produit sont fournies séparément par différents groupes (appartenant ou non à la même entreprise) qui travaillent ensemble pour fournir un produit ou un service (Belkadi *et al.*, 2010). Le produit n'est alors plus l'œuvre d'un groupe de travail, mais il résulte d'actions collectives coordonnées autour d'objectifs partagés (Nguyen Van, 2006). Les entreprises étendues doivent permettre une collaboration entre les équipes et avec les différents partenaires économiques (clients, sous-traitants, fournisseurs...), si bien que les technologies d'échange de données leur sont devenues indispensables (Sackett, 1990).

Dans un premier temps, les étapes du cycle de vie du produit sont présentées (section 2.1.1), puis les technologies PLM permettant la gestion des données et des concepts générées à chaque étape du cycle de vie du produit sont introduites (section 2.1.2).

2.1.1 Cycle de vie du produit

2.1.1.1 Étapes de la vie d'un produit

Selon Boujut & Blanco (2003) et Grebici (2007), le produit évolue considérablement et acquiert de la maturité tout au long de son cycle de vie. Pour Ducellier (2008), cette maturité se résume en quatre étapes : le produit virtuel (définition des exigences du produit), le produit évoluant (conception et fabrication du produit), le produit mature (distribution et utilisation du produit) et le produit en déclin (arrêt de l'utilisation du produit).

Dans (Kiritsis *et al.*, 2003), le cycle de vie du produit est composé de trois phases principales :

1. Début-de-vie (Beginning-of-life – BOL) : inclut la conception et la fabrication du produit. Les équipes de conception et de fabrication collaborent pour définir le produit et également pour planifier les outils de production et les approvisionnements auprès des fournisseurs.
2. Milieu-de-vie (Middle-of-life – MOL) : phase de distribution, d'utilisation et de maintenance du produit. L'historique du produit relatif à son circuit de distribution, à ses conditions d'utilisation, de disfonctionnement et de maintenance peut être collecté, afin de mettre en place un référentiel de suivi du produit mature.
3. Fin-de-vie (End-of-life – EOL) : le produit est retiré du service en vue d'être recyclé ou remplacé. Des informations sur le produit en déclin (composants, matériaux...) sont collectées pour être communiquées aux acteurs assurant son recyclage ou sa réutilisation.

A chaque phase, des quantités d'informations sont générées et doivent être partagées entre les différentes équipes qui interviennent sur le produit : chaque partie prenante doit pouvoir accéder à l'information dont elle a besoin au moment où elle en a besoin.

2.1.1.2 Informations échangées au cours du cycle de vie d'un produit

Boothroyd (1994) constate que environ 80% du coût total de développement du produit est fixée à la conception, si bien que les efforts des entreprises se concentrent sur la phase BOL du cycle de vie du produit, laissant très souvent de côté les phases MOL et EOL. La conception d'un produit est généralement définie comme étant l'ensemble des tâches conduisant à obtenir la représentation de l'objet physique grâce à un modèle. Gardan (2005) présente la conception de produits manufacturés comme le processus d'obtention d'un produit fini répondant à l'expression d'un besoin tel que décrit dans le Cahier des Charges Fonctionnelles (CdCF).

Il existe de nombreuses méthodes et approches de conception. L'*ingénierie simultanée* (ou ingénierie concourante) est définie comme une approche de conception où les activités du processus de développement de produit et les moyens de production sont intégrés et exécutés le plus possible en parallèle, afin de réduire le temps des cycles de développement (Sohlenius, 1992; Jagou, 1993). Cette approche consiste à fédérer autour d'un produit, plusieurs équipes multidisciplinaires, travaillant de façon simultanée pour la réduction des coûts et du temps de mise en service du produit.

L'*ingénierie collaborative* est un cas particulier de l'ingénierie simultanée intervenant dans les types de processus distribué et collaboratif. Selon (Wang *et al.*, 2005), dans un contexte de processus collaboratif, des personnes de différents métiers et différentes entreprises coopèrent pour définir un produit, spécifier sa production, son assemblage ou d'autres processus à travers une coordination, une communication et un contrôle à distance. L'ingénierie collaborative vise donc à une interaction multidisciplinaire des métiers intervenants dans le processus et impliqués dans chaque phase du cycle de vie du produit (Shen *et al.*, 2008).

La figure 2.2 présente les données produites tout au long du cycle de vie d'une voiture : des arbres de spécification, des *BOM* (*Bill-Of-Material*) pour les données techniques de conception et de production, des processus, etc. Les informations générées dans un contexte d'ingénierie collaboratives sont de trois natures : (1) les informations techniques, (2) les informations de standardisation et de normalisation et (3) les informations administratives (Ducellier, 2008). Ces informations et données sont échangées entre acteurs de différentes activités le long du projet, dans un contexte d'ingénierie collaborative et d'entreprise étendue (Belkadi *et al.*, 2010). Ce partage est indispensable pour que les parties prenantes du développement du produit puissent accéder aux bonnes informations au bon moment, ce qui permettra de réduire les temps de mise sur le marché des produits.

Les entreprises doivent se doter de systèmes de gestion de l'information pour faciliter la communication entre entre les acteurs intervenant tout au long du cycle et vie du et assurer la traçabilité des données du produit (Terzi *et al.*, 2007).

2.1.2 Les systèmes PLM pour la gestion du cycle de vie des produits

La complexité croissante des produits et la distribution multi-sites des activités de l'entreprise étendue contribuent à mettre en évidence l'importance du travail collaboratif dans ces types d'organisation (Shen, 2003).

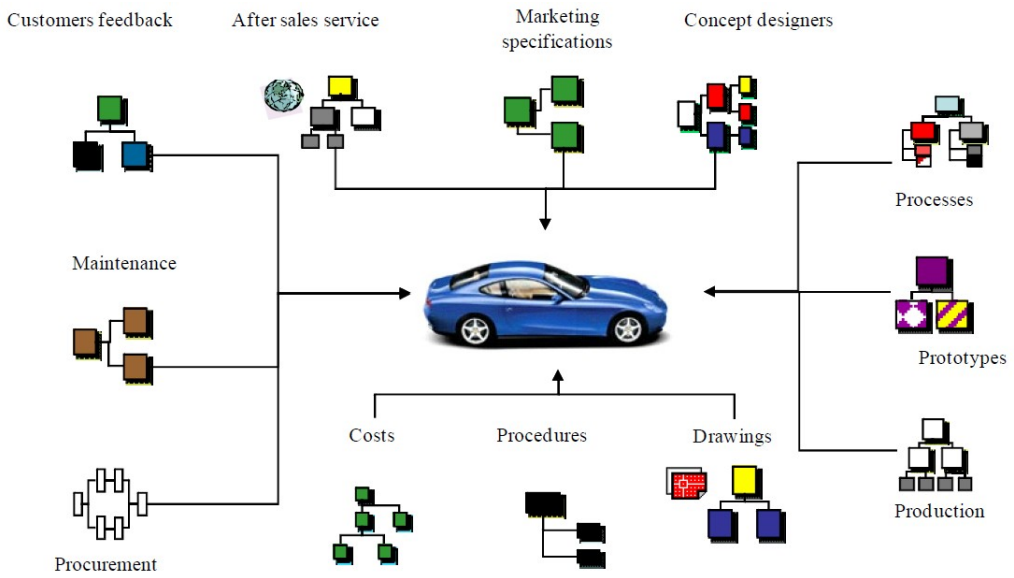


FIGURE 2.2 – Données produit d'une voiture (Terzi *et al.*, 2010)

Les *Systèmes de Gestion des Données Techniques* (*SGDT*) ont été créés au milieu des années 80 pour répondre aux besoins de traçabilité et d'échanges des données d'ingénierie du produit (en particulier les données de Conception Assistée par Ordinateur (CAO)). Dans Randoing (1995), les SGDT sont définis comme étant des "outils intégrés, permettant de consolider et redistribuer l'ensemble du patrimoine informationnel d'un produit, à définir, concevoir, fabriquer et maintenir, et d'en structurer et contrôler les données techniques, leur évolution et leur distribution". Leurs objectifs principaux sont de fournir les information en cohérence avec le statut de développement du produit, c'est-à-dire à la la bonne personne et au bon moment (Rosenman & Gero, 1999; Chen & Jan, 2000). Les SGDT permettent également d'interagir avec différents processus d'entreprise (Eynard *et al.*, 2005), et ils ont par la suite évolués pour s'orienter vers des solutions de gestion du cycle de vie de produits appelées *PLM* (pour *Product Lifecycle Management*) (Debaecker, 2004; Stark, 2004; Saaksvuori & Immonen, 2008).

2.1.2.1 Définition du PLM

Les systèmes PLM ont commencé à apparaitre à la fin des années 90 en tant qu'approche intégrée pour la gestion de la conception de produits dans les industries automobiles et aéronautiques (Konstantinov, 1988). Le complexité des produits développés dans ces domaines, ainsi que la compétition croissante causée par la mondialisation, ont rendu nécessaires l'utilisation d'un système de gestion des données produit efficace (Ming *et al.*, 2005). Afin de rester compétitif en réduisant les coûts, l'objectif principal est de fournir la bonne information à la bonne personne dans le bon contexte et au bon moment tout au long du cycle de vie du produit (Ameri & Dutta, 2005). Pour (Stark, 2004), le PLM est devenu un nouveau paradigme pour la conception et la fabrication, car l'enjeu n'est plus seulement de gérer les données techniques du produit, mais les concepts associés au produit.

Le PLM peut être défini comme un modèle de développement orienté sur le cycle de vie du

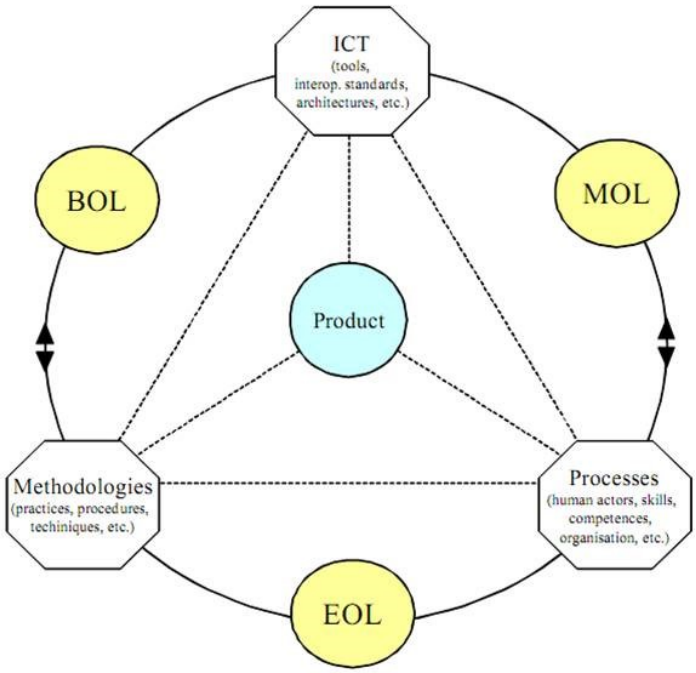


FIGURE 2.3 – Éléments fondamentaux du PLM (Terzi *et al.*, 2010)

produit, et au sein duquel les données produit sont partagées entre les acteurs, les processus et les organisations, à différents stages du cycle de vie du produit (Terzi *et al.*, 2010). Les systèmes PLM intègrent la modélisation, l'ingénierie, la fabrication et la gestion de projet au sein d'une unique plateforme collaborative (Ming *et al.*, 2005). Les informations auxquelles les utilisateurs doivent accéder sont différentes tout au long du cycle de vie, et les systèmes PLM doivent gérer les données produit à plusieurs niveaux de granularité. Par exemple Bill Of Material (BOM) décrit les assemblages et les pièces qui constituent le produit, engineering BOM (eBOM) décompose le produit tel qu'il est conçu, et manufacturing BOM (mBOM) décrit le produit tel qu'il est fabriqué.

Les systèmes PLM répondent à un besoin croissant d'échange, de partage et de gestion de données techniques, lié à l'utilisation intense des outils informatiques (TIC), et à la complexité organisationnelle du cycle de vie des produits manufacturés (Troussier, 2010). Selon (Grieves, 2005), le PLM est une question de traitement numérique de données où les Technologie de l'Information et de la Communication (TIC) jouent un rôle fondamental, et Terzi *et al.* (2010) définit les trois fondamentaux du PLM comme les méthodologies (pratiques, procédures, techniques...), les processus (acteurs, connaissances, organisations...) et les TIC (outils, standards...). Ils sont illustrées dans la figure 2.3.

2.1.2.2 Systèmes PLM actuels

Les sytèmes PLM sont des technologies matures, connues pour augmenter la productivité, maximiser la valeur du produit et réduire les coûts des organisations (Stark, 2004). Les approches orientées-objet ont montré leur utilité en tant que modèles pour intégrer les produits,

les processus et les ressources à travers des modèles UML [1] (Eynard *et al.*, 2004). Le noyau du modèle produit (CPM pour Core Model Product) définit la forme, la fonction et le comportement du produit ; il a été étendu (Fenves *et al.*, 2008) et permet la conception d'un cadre de modélisation des informations produit pour supporter l'ensemble des exigences du PLM (Sudarsan *et al.*, 2005). Les modèles de données du cycle de vie du produit ont été éprouvés dans les systèmes PLM, principalement pour la phase BOL du cycle de vie.

Interfaces des systèmes PLM Les interfaces des systèmes PLM existants n'ont pas fait l'objet d'études jusqu'à présent, bien qu'elles soient utilisées quotidiennement dans l'industrie manufacturière. Les interfaces logicielles des systèmes PLM actuels ne sont ni ergonomiques ni intuitives. De façon générale, les entreprises perdent beaucoup de temps et d'argent à former leurs employés à l'utilisation des systèmes d'information, et le PLM ne fait pas exception.

La principale critique qui puisse être adressée à l'interface est la surcharge de fenêtres, de menus et d'icônes. Les instances de données produit sont présentées sous la forme de liste et seulement un niveau d'information du produit peut être affiché à la fois. Par ailleurs, le vocabulaire de l'interface est beaucoup trop orienté sur les concepts propres à l'industrie, ce qui limite l'extensibilité des systèmes PLM à d'autres domaines.

L'annexe B présente deux études de cas montrant les limites des interfaces des systèmes PLM actuels : (1) la migration du système PLM d'une entreprise de conception de systèmes thermiques et (2) la gestion de données d'imagerie biomédicale. Ces limites se déclinent en trois axes :

- La recherche de données : il est difficile pour un utilisateur non expert de former des requêtes complexes sur les données stockées dans un système PLM, ce qui pose des problèmes notamment dans le domaine de l'imagerie biomédicale.
- L'affichage des données : les informations sont affichées exclusivement sous la forme de listes, les relations comme les concepts sont représentés sous la même forme sans distinction de leur type et fonction, ce qui rend difficile l'identification des différents niveaux d'information.
- La navigation des relations : les relations ascendantes et descendantes sont affichées dans des fenêtres différentes et l'utilisateur doit alterner entre plusieurs fonctionnalités pour analyser des relations complexes.

Les systèmes PLM sont principalement accessibles via des clients riches installés en applications autonomes au niveau local sur les ordinateurs, mais aussi via des clients web comme le PLM Windchill. Les clients web deviennent peu à peu une évolution inévitable de tous les systèmes PLM dans l'industrie manufacturière, afin de faciliter l'accès aux données partout dans le monde et sur de nouveaux supports comme les tablettes tactiles. Plusieurs clients web PLM proposent des fonctionnalités intéressantes (ARAS, Windchill), mais leur interface est très surchargée. Récemment le client Active Workspace a été développé avec des paradigmes de navigation similaires à ceux des sites web commerçants (boutique en ligne), mais la navigation reste complexe.

1. Unified Modeling Langage

2.1.2.3 Applications du PLM en dehors de l'industrie manufacturière

Bien qu'à l'origine les systèmes PLM ont été conçus pour les domaines de l'automobile et de l'aéronautique, ils sont désormais adoptés par l'ensemble de l'industrie manufacturière, dont le secteur pharmaceutique (Fielding *et al.*, 2014). De plus en plus de travaux de recherche s'intéressent à d'autres domaines que l'industrie manufacturière, tels que l'ingénierie assistée par ordinateur (IAO), la mécatronique, l'ingénierie et l'architecture en bâtiment, les services ou encore l'imagerie biomédicale (BMI). Ces domaines sont multidisciplinaires et doivent gérer des données hétérogènes liées par des relations complexes qu'il est difficile de parcourir.

Malgré les similarités entre l'industrie manufacturière et le domaine de l'imagerie biomédicale, le PLM n'est pas encore utilisé dans ce dernier, à quelques exceptions près :

- Conception et fabrication de prothèses : chaque prothèse est adaptée aux spécificités du patients et conçues à partir de reconstructions 3D basées sur des images scannées du patient (Lantada & Morgado, 2013). Tornier, Groupe Lepine ou Mount Kisco Medical Group sont des exemples d'entreprises qui utilisent le PLM pour gérer le cycle de vie de chaque produit, quels que soient les types de documents : images, CAO ou texte.
- Entreprises des services de santé : le PLM est couplé avec le matériel médical d'acquisition des données pour éviter les entrées manuelles de données et les erreurs qui pourraient en résulter.

Les difficultés spécifiques rencontrées par les nouveaux domaines d'application du PLM n'ont pas été questionnées jusqu'à présent, bien que l'ensemble de la communauté PLM pourrait en profiter. Étant donné la complexité croissante des relations entre les données autant dans l'industrie manufacturière que dans les nouveaux domaines d'application du PLM, il apparait important de mener des travaux de recherche en vue d'améliorer la navigation et l'analyse des relations.

Dans un contexte de compétitivité accru, **l'ingénierie collaborative est devenu incontournable** dans l'industrie manufacturière.
Les problématiques de partage des données issues de l'apparition de l'ingénierie collaborative ont été résolues par la création des **SGDT puis des systèmes PLM qui permettent de tracer les données d'un produit tout au long de son cycle de vie**.
Les systèmes PLM actuels présentent des **limites en terme d'interfaces** qui rendent difficile l'exploration des relations entre les données générées à toutes les étapes du cycle de vie du produit.

2.2 Gestion des données en neuroimagerie

La neuroimagerie présente des caractéristiques qui nous intéressent : une forte hétérogénéité des données renforcée par la pluridisciplinarité des études, la nécessité du partage et de la réutilisation des données dont la génération a un coût financier non négligeable, ainsi que

l'exploration visuelle des données. Depuis vingt-cinq ans, les chercheurs en neuroimagerie ont pris progressivement conscience de l'importance d'une gestion durable de ses données pour en faciliter le partage et la réutilisation, et ils se sont dotés d'outils informatiques pour faire face à la complexité grandissante de la provenance des données.

La section 2.2.1 introduit les problématiques liées au partage des données dans le domaine neuroimagerie, puis les systèmes de gestion des données pour la neuroimagerie sont comparés dans la section 2.2.2.

2.2.1 Le partage des données en neuroimagerie : une nécessité

Plusieurs publications ont mis en avant ces dernières années les problématiques associées au partage de données en neuroimagerie, exposant notamment les raisons de la nécessité d'un partage des données à grande échelle (Poline *et al.*, 2012).

Un chercheur en neurosciences ne peut pas être compétent dans toutes les étapes d'une étude (connaître les détails de la physique par résonance magnétique, les analyses statistiques avancées...), c'est pourquoi le partage des données avec des pairs, à la fois à l'intérieur et entre les disciplines scientifiques, est une composante inhérente et nécessaire à la neuroimagerie.

Les coûts financiers et humains, ainsi que les difficultés engendrées par des études de recherche en neuroimagerie sont élevés : les scanners d'acquisition IRM restent chers, et assurer le financement, la validation des protocoles expérimentaux et l'acquisition de données sur un grand nombre de sujets demandent du temps et des efforts. Par conséquent, seules les grandes structures de recherche et les projets collaboratifs au niveau national ou international peuvent accéder aux ressources et équipements nécessaires. Réutiliser les données d'études existantes permet non seulement à de petites équipes de recherche de profiter de jeu de données qu'ils leur seraient impossible à acquérir, mais aussi d'exploiter au maximum les ressources existantes pour optimiser les coûts et favoriser un plus grand nombre d'avancées scientifiques pour un même investissement. D'autres domaines, comme la génétique (Benson *et al.*, 2010; Kaye *et al.*, 2009), la médecine ou l'histoire naturelle, ont démontré que le partage de données à grande échelle permet des avancées scientifiques rapides (Yarkoni *et al.*, 2010).

Associer à une publication les données qui ont conduit au résultat scientifique est un gage de la qualité de la publication, puisque les résultats peuvent être vérifiés par les pairs qui valident la publication (Birney *et al.*, 2009) et reproduits par n'importe quel membre de la communauté scientifique. La reproduction des résultats de recherche est considéré depuis longtemps comme un pré-requis fondamental à la démarche scientifique.

Si en 2001 le partage des données est encore une idée émergente dans le domaine des neurosciences (Van Horn *et al.*, 2001), le partage des données est devenu une question centrale ces dernières années, autant pour les équipes de recherche que pour les financeurs des études. Nichols & Pohl (2015) a mis en avant qu'aux États-Unis, les organismes nationaux ne financent plus les projets de recherche biomédicale de grande envergure qu'à la condition du partage d'une partie des données générées dans le projet (Collins & Tabak, 2014).

La réponse à une question *a priori* anodine – quelles données doivent être partagées ? – ne fait pas encore l'unanimité. Avec un accès aux données brutes, une reproductibilité totale des résultats est possible, et les données peuvent être réutilisées pour calculer des données dérivées qui n'étaient pas initialement prévues dans le cadre de l'étude qui les a générées (Adamson & Wood, 2010). Cependant les données dérivées sont déjà filtrées et peuvent être utilisées par un plus grand nombre de personnes, puisque l'expertise pour les générer depuis des données brutes n'est pas nécessaire. Ces données raffinées présentent donc une valeur ajoutée, cependant l'interprétation de leur provenance est plus complexe. Pour Fox & Lancaster (2002), seules les données publiées peuvent être partagées car ce sont des données vérifiées et donc fiables. De nombreuses méta-analyses sont menées sur des données publiées mises à la disposition des chercheurs dans des bases de données publiques sur Internet. Pourquoi choisir, pour s'assurer d'une provenance complète, il faut garder la description des antécédents d'une donnée.

Dans un premier temps nous nous intéressons aux données générées tout au long d'une étude, puis dans un second temps les limites conceptuelles et techniques qui empêchent un partage des données à grande échelle en neuroimagerie sont discutées.

2.2.1.1 Étapes de gestion des données d'une étude de recherche

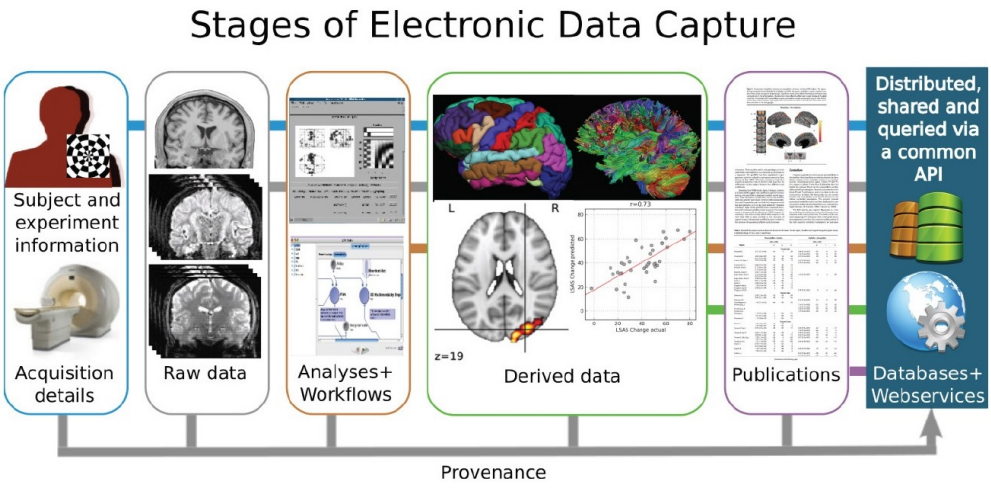


FIGURE 2.4 – Étapes de la capture électronique des données (Electronic Data Capture – EDC) pour le partage des données en neuroimagerie (Poline *et al.*, 2012)

La *capture électronique des données* (*EDC* pour *Electronic Data Capture*) est un terme générique qui prend un sens spécifique dans le domaine biomédical : utiliser des logiciels pour collecter les données d'une étude sous une forme électronique au lieu de les collecter sur un support papier. Poline *et al.* (2012) propose les étapes de l'EDC pour la neuroimagerie (en vue du partage des données), qui sont illustrées dans la figure 2.4 : (1) les détails des expérimentations menées (données du sujets, matériel utilisé, paramètres...) sont collectées, (2) les données brutes sont stockées, (3) des chaînes de traitement sont exécutées pour permettre l'analyse des données, (4) les données dérivées sont obtenues et analysées, (5) les résultats de l'étude sont publiés.

L'ensemble des données est géré dans un système accessible via une interface ou un web service. Le système gère en particulier la *provenance* associée aux données : ce que sont les données, comment, où et quand elles ont été produites, pourquoi et dans quel but (Simmhan *et al.*, 2005). La provenance des données assure la qualité, l'exactitude, la reproductibilité et la réutilisation des résultats d'une étude de recherche.

Pour comprendre quelles données sont générées tout au long d'une étude de recherche, et la nature de la provenance entre les données de chaque étape. A la suite de l'analyse de la bibliographie et d'entretiens menés au sein du laboratoire GIN sur les données générées lors d'une étude, quatre étapes de déroulement d'une étude sont identifiées et illustrées dans la figure 2.5. Avant de commencer une étude, la nature des acquisitions et de leur utilisation future doit être détaillée et approuvée par un comité d'éthique. Cette information est appelée *spécifications d'une étude* (*study specifications*), et constitue l'étape (1) d'une étude de recherche. Les *données brutes* (*Raw data*) comme les images, les données cliniques, les résultats d'un test psychologique ou encore les résultats génétiques, sont acquises à partir des spécifications d'une étude et sont annotées. Elles constituent l'étape (2) d'une étude. Les *données dérivées* (*derived data* peuvent être calculées à partir des données brutes et d'autre données dérivées. Il existe différents types de données dérivées : données d'un seul type ou combinant des types hétérogènes, provenant d'un seul sujet ou de multiples sujets. Ces données constituent l'étape (3) d'une étude. Pour finir, les données dérivées sont préparées (mises au format) pour la publication et proposées à la communauté. Ces *données publiées* (*published results*) constituent l'étape (4) d'une étude et sont qualifiées de données finalisées et de qualité, qui sont validées par les pairs de la communauté.

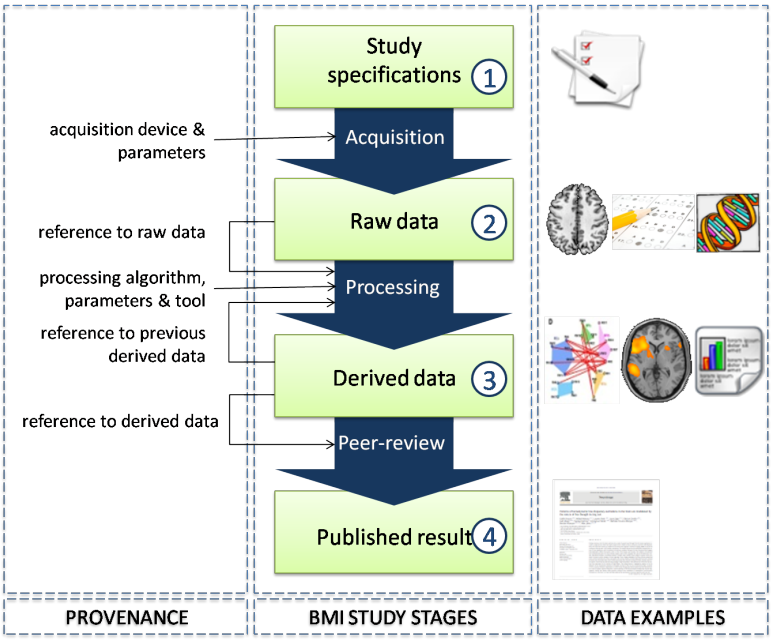


FIGURE 2.5 – Les quatre étapes d'une étude de recherche en neuroimagerie

Acquisition, *traitement* (*processing*) et *revue par les pairs* (*peer-review*) représentent les processus entre deux étapes d'une étude de recherche, et c'est la trace de ces processus qui constitue

la provenance permettant de retrouver comment une donnée a été obtenue.

Une représentation de la provenance est proposée par le consortium World Wide Web (W3), qui développe des standards pour soutenir le développement du web. La norme PROV-DM (PROVenance Data Model) [2] définit la provenance comme un enregistrement qui décrit les personnes, les institutions, les entités et les activités impliqués dans la production, et la distribution d'une données ou d'un objet. Une *entité* peut être physique, numérique ou conceptuelle. Une *activité* a lieu sur une période temps et agit avec ou sur une ou plusieurs entités. Cela inclut la consommation, le traitement, la transformation, la modification, l'utilisation ou la génération d'entités. Un *agent* a une responsabilité dans l'exécution d'une activité. Les entités, les activités et les agents sont modélisés par sept relations :

- WasDerivedFrom : une entité peut être dérivée d'une autre entité.
- Used : une activité utilise une entité pour sa réalisation
- WasGeneratedBy : une entité peut avoir été générée par la réalisation d'une activité.
- WasInformedBy : le lancement ou les paramètres d'une activité peuvent avoir été communiqués par une autre activité.
- WasAssociatedWith : une activité est associée à agent qui est donneur d'ordre ou exécutant.
- WasAttributedTo : une entité est attribuée à un agent.
- ActedOnBehalfOf : un agent peut agir sur ordre d'un autre agent.

La figure 2.6.a présente le schéma UML de la norme PROV-DM. Pour illustrer l'utilisation de la norme générique PROV-DM dans le domaine de la neuroimagerie, nous modélisons dans la figure 2.6.b la transformation d'une donnée brute en une donnée dérivée : un chercheur (agent) applique un traitement de recalage (activité) sur la donnée brute IRMf-r (entité) et obtient la donnée dérivée IRMf-r recalée (entité).

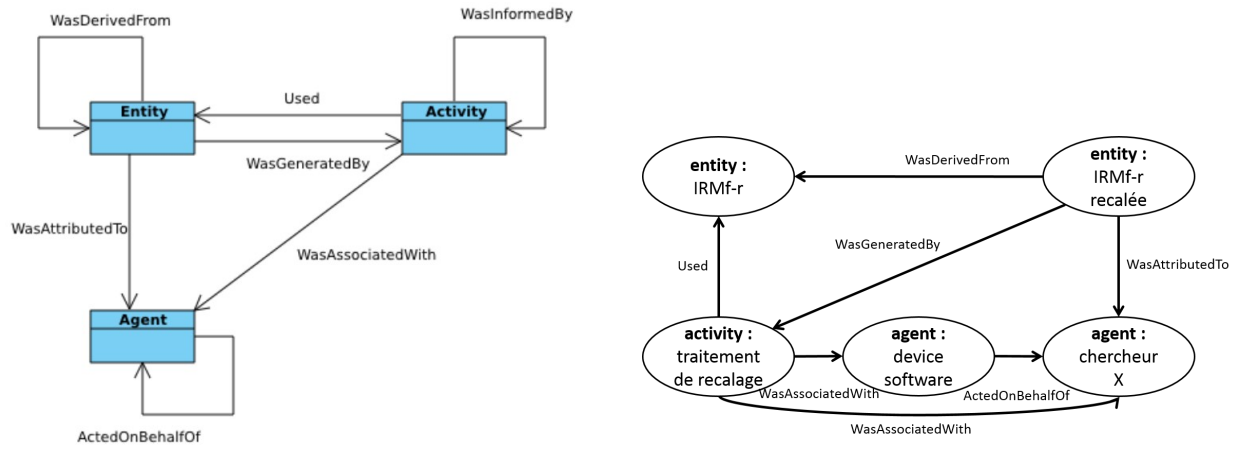
Récemment Keator *et al.* (2013) a développé un modèle de données pour la gestion des données de neuroimagerie qui inclut la norme PROV-DM pour la gestion de la provenance.

2.2.1.2 Limites au partage des données

Bien qu'un consensus ait émergé dans le domaine de la neuroimagerie sur la nécessité du partage des données générées par la communauté, des difficultés techniques et sociales entravent les efforts fournis.

La provenance en neuroimagerie est complexe à décrire – rien que d'acquérir des jeux de données équivalants sur des machines de marques différentes requière des précautions particulières dans la conception du protocole (Kruggel *et al.*, 2010) –, ce qui complique d'entrée le partage des données. Par ailleurs, les bases de données existantes utilisent chacune leur propre terminologie et leurs propres modèles de données, ce qui empêche d'identifier des données à la provenance similaires et aussi de requêter des métadonnées à grande échelle. Pour (Teeters *et al.*, 2008), c'est surtout le manque d'un méta modèle standardisé qui réduit les échanges entre les bases de données pour la neuroimagerie. En effet, les laboratoires utilisent des paradigmes

2. http://www.w3.org/TR/2013/REC-prov-dm-20130430/



a b

FIGURE 2.6 – Représentation de la provenance : a) Structure du standard PROV développé par W3 (http://www.w3.org/TR/2013/REC-prov-dm-20130430/), b) Exemple d'un chercheur qui applique un traitement à des données

expérimentaux variés, et les méthodes utilisées pour générer les données dans le cadre de ces paradigmes sont sensiblement différents. Lorsqu'un laboratoire implémente une base de données, le premier réflexe est souvent de concevoir le modèle de données sur les processus utilisés en interne.

Cependant, même si les chercheurs utilisaient un modèle de données unifié dans toutes les bases de données utilisées, le caractère intrinsèquement évolutif de la recherche pose problème. Les protocoles expérimentaux, l'acquisition des données et les processus de traitement peuvent être sujets à de grandes variations entre deux études, et par ailleurs personne ne peut pas espérer décrire *a priori* tous les protocoles expérimentaux que des chercheurs en neuroscience vont avoir besoin d'incorporer dans une structure de base de données rigide Van Horn *et al.* (2001). Tout modèle de gestion des données à destination de la neuroimagerie doit donc pouvoir être facilement modifiable sans causer des modifications majeures dans la structure globale du système pour s'adapter aux nouveau protocoles et aux nouveaux concepts développés quotidiennement par la recherche.

Pour Buckow *et al.* (2014), les études de neuroimagerie à grande échelle posent non seulement des problèmes techniques, mais aussi sociaux. Ces limites avaient déjà été mises en exergue par Poline *et al.* (2012) : d'une part le partage des données prend du temps et de l'argent, et les chercheurs sont réticents à partager avec d'autres des données durement acquises ; d'autre part les chercheurs craignent de voir le résultat de leurs analyses en compétition avec d'autres chercheurs.

Enfin, certaines considérations éthiques et légales peuvent freiner la partage des données (Ascoli, 2006; Poline *et al.*, 2012). La perspective de voir leurs données personnelles diffusées sur des bases de données publiques – bien que la préservation de l'anonymat soit obligatoire

vis à vis de la législation – pourrait freiner la participation des sujets aux études. Par ailleurs tous les pays n'ont pas la même législation en matière d'éthique et de protection des données humaines, ce qui limite les programmes de partage internationaux de données.

Pour résoudre les problèmes techniques qui limitent le partage des données, l'utilisation d'ontologies pour faciliter la médiation entre bases de données est une piste intéressante qui a déjà été expérimentée (Ashish *et al.*, 2010). Il existe plusieurs ontologies développées pour la biologie (OBO [3]) et plus spécifiquement pour les neurosciences (Cognitive Atlas Project [4], CogPO [5], Neurolex [6], ONTONeurolog [7]...).

Plusieurs initiatives proposent des modèles à utiliser pour partager les données avec leur provenance. Poldrack *et al.* (2008) a décrit comment les résultats d'une étude de recherche devraient être communiqués et indique aux auteurs quelques informations fournir avec une publication et comment les présenter. XCEDE (XML-based Clinical and Experimental Data Exchange) est un schéma XML développé par le réseau BIRN (Biomedical Informatics Research Network). Il fournit une hiérarchie de métadonnées qui permet de stocker, de décrire et de documenter des données générées par des études de recherche à différents niveaux : projets, sujets, protocoles, acquisitions... L'historique des traitements et des changements subis par les données est également décrit dans le format XCEDE. Ce format extensible a été développé pour faciliter les échanges entre bases de données, outils et services web (Gadde *et al.*, 2012). Keator *et al.* (2013) propose un schéma pour le partage structuré de données brutes et dérivées entre bases de données. Il s'appuie sur le modèle de la provenance présenté dans la sous-section 2.2.1.1 (figure 2.6).

2.2.2 Les systèmes de gestion des données pour la neuroimagerie

De nombreux logiciels de gestion des données pour la neuroimagerie ont été développés. Nous avons comparé dix-huit d'entre eux. Dans un premier temps nous présentons notre méthode et les critères de comparaison choisis, puis une synthèse des principales stratégies de gestion des données est présenté. Pour finir, les limites des systèmes de gestion des données actuels en neuroimagerie sont identifiées.

2.2.2.1 Critères de comparaison

Nichols & Pohl (2015) a comparé sept systèmes de gestion des données conçu spécifiquement pour le domaine de la neuroimagerie (COINS, HID, IDA, LORIS, NiDB, REDCap et XNAT), en s'intéressant aux fonctionnalités principales et à la complexité de déploiement des systèmes au sein d'un laboratoire ou en vue d'un projet multi-sites. Cependant cette étude comparative ne s'est pas intéressée à la gestion intégrée des données et de leur provenance depuis les

3. http://www.obofoundry.org/
4. http://www.cognitiveatlas.org/
5. http://www.cogpo.org/
6. www.neurolex.org/
7. http://neurolog.i3s.unice.fr/public_namespace/ontology

spécifications d'une étude aux données publiées.

Lors de la conception du système NiDB (Neuroinformatics Database), Book *et al.* (2013) ont identifié six fonctionnalités que toute base de données pour la neuroimagerie devrait proposer : (1) l'association de toute donnée à un sujet, (2) un stockage des données centré sur le sujet, (3) un contrôle qualité en temps réel pour les acquisitions d'imagerie, (4) des chaînes de traitement automatisées, (5) un design simple, et (6) un import et un export facilité.

Dans notre comparaison, nous nous intéressons moins aux caractéristiques techniques (technologies utilisées, infrastructures des bases de données) qu'aux fonctionnalités offertes par les systèmes de gestion des bases de données et à l'hétérogénéité des données gérées. Les critères de comparaison retenus sont répartis en quatre catégories :

- **Disciplines gérées :**
  - Clinique : données cliniques et démographiques du sujet.
  - Imagerie : quelques que soient les modalités supportées.
  - Psychologie : résultats aux tests comportementaux.

    Nous n'avons pas retenu la discipline "génétique", car un seul système de gestion des données le permet.
- **Types de données gérées :**
  - Étude : gestion de données de définition d'une étude, comme les spécifications de l'étude.
  - Acquisitions : gestion de données brutes.
  - Traitements : gestion de données dérivées, issues de l'application d'un ou plusieurs traitements sur des données brutes.
  - Publications scientifiques : gestion des articles scientifiques publiés et des données qui ont permis d'obtenir les résultats présentés dans les publications.
- **Processus supportés :**
  - Conception de formulaires de recueil de données : les utilisateurs peuvent créer leurs propres formulaires d'acquisition de donnée dans l'interface.
  - Vérification des données : un processus de vérification de la qualité des données – brutes et/ou dérivées – est disponible à l'import ou après l'application d'un traitement.
  - Workflows de traitement de données : des chaînes de traitement peuvent être lancées depuis l'interface du système (accès à des librairies et/ou des pipes de calcul prédéfinis).
  - Visualisation intégrée des données : le système permet de visualiser directement des données dans l'interface, en particulier des images ou des graphes.
- **Autres fonctionnalités :**
  - Contrôle d'accès : les utilisateurs doivent se connecter à la base de données avec un compte auquel des droits spécifiques sont associés, déterminant les données accessibles.
  - Partage public : les données sont accessibles et utilisables par n'importe quelle personne qui en fait la demande.
  - Multi-sites : la base de données est partagée entre plusieurs partenaires (laboratoires, institutions...) qui peuvent être géographiquement éloignés.
  - Standards d'échanges : le système utilise des standards de la communauté.
  - Intégration d'ontologies : le système utilise des ontologies du domaine qui peuvent être

utiles pour la médiation avec d'autres bases de données.
- Modularité du logiciel : le système est personnalisable et son modèle de données extensible.

2.2.2.2 Tableau comparatif

Le résultat de la veille effectuée est présentée selon les critères définis précédemment dans le tableau 2.1. Les dix-huit systèmes comparés sont : REDCap (Research Electronic Data CAPture), HID (Human clinical Imaging Database), UMCD (UCLA Multimodal Connectivity Database), COINS (Collaborative Informatics and Neuroimaging Suite), XNAT (eXtensible Neuroimaging Archive Toolkit), CVT (Connectome Viewer Toolkit), NiDB (Neuroinformatics DataBase), LORIS (LongitudinalOnlineResearchandImagingSystem), IDA (LONI Image & Data Archive), Shanoir (SHAring NeurOImaging Resources), BIL&GIN (Brain Imaging Laterality & Groupe d'Imagaerie Neurofonctionnelle), SuMSdb (Surface Management System database), BrainMap, openfMRI, fMRIDC (fMRI Data Center), Brede, DFBIdb, Neurolog.

Le système de gestion des données HID n'est plus développé actuellement, mais il constitue une preuve de concept qui a été l'inspiration de beaucoup d'autres systèmes (Nichols & Pohl, 2015). XNAT est le système le plus déployé [8], et dispose d'une grande communauté d'utilisateurs.

La plupart des systèmes comparés sont libres (open source). Hormis pour les systèmes COINS et IDA qui proposent un service de base de données de type cloud, les bases de données doivent être installées au sein des laboratoires qui les mettent en place.

L'import des données peut être facilité par la connexion de la base à des web services de saisie des données sous forme de formulaires. Ces formulaires sont parfois personnalisables par les utilisateurs, pour s'adapter aux spécificités de chaque étude. De nombreux systèmes de gestion des données proposent des workflows de contrôle de la qualité des données à l'import. L'accès à des librairies et à des moteurs de chaînes de traitement est parfois intégré directement dans les systèmes de gestion des données, comme par exemple le Pipeline Software Package (IDA) ou des librairies Python (CVT).

8. Une liste des installations de XNAT dans le monde est disponible à l'adresse : http://www.xnat.org/about/xnat-implementations.php

Outil	**Référence**	2 *Imagerie*	3 *Psychologie*	4 *Clinique*	5 *Etude*	7 *Acquisitions*	8 *Traitements*	9 *Publications*	10 *Formulaires*	11 *Contrôles*	12 *Traitements*	13 *Visualisation*	15 *Contrôle d'accès*	16 *Partage public*	17 *Multi-sites*	19 *Standards*	20 *Ontologies*	21 *Modularité*
REDCap	(Harris *et al.*, 2009)		■	■	■	■	■		■	■			■		■		■	■
HID	(Keator *et al.*, 2008)	■	■	■		■	■		■						■	■		■
UMCD	(Brown *et al.*, 2012)	■					■				■	■	■	■				
COINS	(Scott *et al.*, 2011)	■		■	■				■						■	■		■
XNAT	(Marcus *et al.*, 2007)	■	■		■	■	■			■	■		■		■		■	■
CVT	(Gerhard *et al.*, 2011)	■				■	■				■	■			■			■
NiDB	(Book *et al.*, 2013)	■			■	■	■			■	■	■	■		■			■
LORIS	(Das *et al.*, 2011)	■	■			■	■		■	■		■	■		■			
IDA	(Van Horn & Toga, 2009)	■		■		■	■			■	■	■	■		■			
Shanoir	(Barillot *et al.*, 2015)	■		■		■	■			■	■	■	■		■			
BIL&GIN	(Joliot *et al.*, 2010)	■	■			■	■				■							
SuMSdb	(Dickson *et al.*, 2001)	■					■	■						■				
BrainMap	(Fox & Lancaster, 2002)	■					■	■						■				
openfMRI	(Poldrack *et al.*, 2013)	■	■	■		■	■	■						■				
fMRIDC	(Van Horn *et al.*, 2001)	■					■	■						■				
Brede	(Nielsen, 2009)	■					■	■			■							
DFBIdb	(Adamson & Wood, 2010)	■				■			■	■		■	■			■		■
Neurolog	(Dojat *et al.*, 2011)	■				■	■				■	■					■	■

TABLE 2.1 – Comparaison des caractéristiques des principales solutions existantes de gestion des données en neuroimagerie.

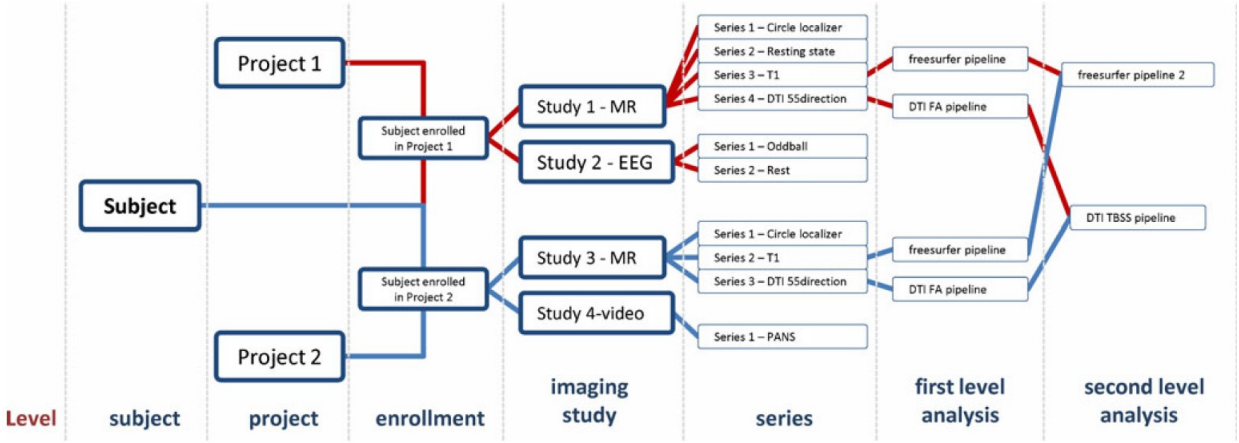


FIGURE 2.7 – Modèle de données centré sur le sujet du système de gestion des données NiDB (Book *et al.*, 2013)

Quatre des systèmes présentés organisent la gestion des données autour du sujet (REDCap, COINS, XNAT et NiDB). Une illustration du modèle de données du système NiDB est présenté dans la figure 2.7 : le sujet est le premier niveau d'entrée des données dans le système, suivi des études, puis des acquisitions et enfin des données dérivées.

L'étude du tableau de comparaison des dix-huit systèmes de gestion des données en neuroimagerie nous permet d'identifier deux catégories :

- La première catégorie de bases de données permet aux chercheurs de gérer leurs données localement et de partager entre sites et entre laboratoires dans le cadre d'études à grande échelle. Ces bases de données permettent de stocker des données brutes et dérivées (étapes (2) et (3) d'une étude) ainsi que des métadonnées pour la gestion des études et la récupération des données (requêtes complexes). Les types de données prises en compte sont des données démographiques et d'imagerie principalement, avec parfois des résultats de tests psychologiques ou d'autres tests hors-imagerie. Des exemples de tels outils sont : XNAT [9] (Marcus *et al.*, 2007), LORIS (Das *et al.*, 2011), COINS (Scott *et al.*, 2011) et IDA (Van Horn & Toga, 2009). Ils sont utilisés à l'échelle locale et à l'échelle multi-sites pour stocker et partager facilement des données entre acteurs d'un projet.
- La seconde catégorie se focalise sur les résultats provenant d'article scientifiques à comité de lecture (étape (4) d'une étude). Les coordonnées stéréotaxiques de pics d'activation – l'une des données dérivées les plus recherchées en neuroimagerie fonctionnelle actuellement – et leurs métadonnées sont les données que l'on trouve le plus fréquemment dans ces bases de données. A certains égards, les bases de données de résultats publiés, en particulier BrainMap (Fox & Lancaster, 2002), augmente le contenu de la littérature en fournissant une description de la provenance des données très complète : il apparait que les informations que doivent apporter les auteurs des articles pour satisfaire aux exigences de ces bases de données sont parfois plus complètes et précises que ce qui est rédigé dans

9. http ://xnat.org/about/xnat-implementations.php

l'article initial (Fox *et al.*, 2005). Des exemples de ces bases de données sont : BrainMap, SumsDB (Dickson *et al.*, 2001).

Le système de gestion des données de UMCD se situe à cheval sur les deux catégories, puisqu'il s'agit d'une base de données ouverte au public et contenant uniquement des données de connectivité fonctionnelle cérébrale dans le but de les explorer grâce aux graphes (matrices de connectivité fonctionnelle). UMCD intègre des librairies de traitement et visualisation de graphes. Une limite du système concerne la provenance des données : les matrices ne sont pas toutes calculées à partir des mêmes atlas (nombre et nomenclature des régions), et cela pose parfois des problèmes de traçabilité pour comparer les réseaux (Brown *et al.*, 2012).

2.2.2.3 Critique des solutions existantes

Gestion globale Actuellement aucun système de gestion des données permet la gestion de données hétérogènes depuis les spécifications d'une étude (1) aux résultats publiés (4). Seul NiDB prend en compte de façon complète l'étape (1). Beaucoup de systèmes intègrent des processus de contrôle qualité à l'import des données brutes, mais encore peu d'entre eux en propose pour le calcul de traitements sur les données (soit entre les étapes (2) et (3) d'une étude). Un nombre réduit de systèmes de gestion des données, XNAT et IDA, intègrent des workflows entre les étapes (2) et (3), avec une étape de contrôle qualité.

Gestion de données hétérogènes Aucun système de gestion des données ne permet d'intégrer les données de nouvelles disciplines.

Gestion des fichiers de données intégrée La gestion des fichiers de données est le point faible de tous les types de systèmes de bases de données en neuroimagerie. Certaines bases de données locales ou collaboratives ne contiennent que des métadonnées, et les fichiers de données sont accessibles via leur adresse locale sur les disques. Cela constitue un problème majeur pour la cohérence et la récupération des données, car rien ne garantit que les fichiers sont toujours à la même adresse. Dans le cas de bases de données de résultats publiés, seules les données dérivées contenues dans les publications sont gérées en plus des métadonnées, et très rarement les données brutes qui ont permis aux auteurs de lance leurs analyses. C'est un frein à la reproduction des résultats scientifiques et à la réutilisation des données pour de futures analyses.

Stratégies de réutilisation Peu de systèmes de gestion des données proposent une stratégie de réutilisation qui s'appuie sur la sauvegarde complète de la provenance des données. Cohérence et complétude de la description des données est une bonne stratégie pour la récupération des données (Poldrack *et al.*, 2008), mais elle n'est la plupart du temps que partiellement implémentée et il n'existe actuellement aucun standard. La plupart des systèmes de gestion des données appartenant à la 2e catégorie et qui gèrent les résultats publiés utilisent des taxonomies ou des ontologies pour améliorer la récupération des données et les requêtes depuis d'autres bases de données, ce qui permet la manipulation des concepts liés aux données, mais qui n'est pas suffisant pour capturer la provenance des données.

Flexibilité de l'organisation des données Plusieurs systèmes proposent des interfaces pour concevoir les formulaires d'acquisition et d'import des données selon les besoins de l'étude, ce qui permet une plus grande flexibilité. Le système XNAT autorise l'ajout de plugins pour étendre son modèle et ses fonctionnalités, mais il faut savoir développer. A cause des technologies utilisées pour développer les systèmes de gestion des données, il est difficile d'obtenir un modèle de données flexible, bien que les traitements et méthodes utilisées pour la recherche en neuroimagerie évoluent de façon récurrente. La plupart des systèmes sont des bases de données relationnelles (basées sur du SQL), ce qui permet de modifier des attributs, mais difficilement la structure du modèle de données.

Malgré l'existence une vingtaine de systèmes de gestion des données pour la neuroimagerie développées par la communauté, des limites techniques et conceptuelles empêchent la gestion, le partage et la réutilisation des données générées pendant les études de recherche.

Les verrous à lever pour concevoir un système de gestion des données de neuroimagerie idéal sont :

- **Provenance** – gérer l'intégralité des données d'une étude depuis ses spécifications jusqu'aux publications et capturer la provenance associée, pour un partage et une réutilisation optimale des données.
- **Hétérogénéité** – accepter tous les formats de données et gérer les concepts de plusieurs disciplines.
- **Flexibilité** –permettre les évolutions du modèle de données sans conséquences sur les données déjà présentes dans la base de données.

Une étude de recherche peut être décrite en quatre étapes : (1) spécifications de l'étude, (2) données brutes, (3) données dérivées, (4) publications.

La pluridisciplinarité des études, le temps et les coûts d'acquisition et la reproduction des résultats incitent **les chercheurs en neuroimagerie à partager les données existantes, cependant des limites techniques et sociales empêchent encore aujourd'hui le partage et la réutilisation des données à grande échelle**.

Les systèmes de gestion des données développés pour la neuroimagerie présentent des limites en terme de gestion des données hétérogènes tout au long d'une étude, de gestion de la provenance et de flexibilité des modèles de données.

Les verrous à lever pour concevoir un système de gestion des données de neuroimagerie idéal sont : provenance, hétérogénéité et flexibilité.

Conclusion du chapitre 2

Dans ce chapitre nous avons présenté les concepts liés à la gestion du cycle de vie du produit dans l'industrie manufacturière, ainsi que les apports et limites des systèmes PLM (Product

Lifecycle Management). Ces derniers permettent d'apporter la bonne information à la bonne personne et au bon moment tout au long du cycle de vie du produit. Les interfaces des systèmes PLM rendent cependant difficile l'exploration de la provenance des données.

Dans un second temps, les problématiques du domaine de la neuroimagerie en terme de partage des données ont été introduites, et les solutions de gestion des données développées pour la neuroimagerie ont été comparées. La pluridisciplinarité des études, le temps et les coûts d'acquisition et la reproduction des résultats incitent les chercheurs en neuroimagerie à partager les données existantes, cependant des limites techniques et sociales empêchent encore aujourd'hui le partage et la réutilisation des données à grande échelle. Les systèmes de gestion des données développés pour la neuroimagerie présentent des limites en terme de gestion des données hétérogènes tout au long d'une étude, de gestion de la provenance et de flexibilité des modèles de données.

Afin d'orienter notre réponse au problème 1 "comment gérer les données hétérogènes et leur provenance ?" défini dans le chapitre 1, nous avons identifié **trois verrous à lever** pour une gestion adaptée des données de neuroimagerie tout au long d'une étude pour faciliter leur partage et leur réutilisation :

- **Provenance** – gérer l'intégralité des données d'une étude depuis ses spécifications jusqu'aux publications et capturer la provenance associée, pour un partage et une réutilisation optimale des données.
- **Hétérogénéité** – accepter tous les formats de données et gérer les concepts de plusieurs disciplines.
- **Flexibilité** –permettre les évolutions du modèle de données sans conséquences sur les données déjà présentes dans la base de données.

Nous avons formulé dans le chapitre 1 l'hypothèse que les solutions de gestion des données développées pour l'industrie manufacturière pourraient être appliquées à la gestion des données en neuroimagerie. Les systèmes PLM développés pour la gestion des données et des concepts associés au cycle de vie du produit en ingénierie collaborative permettent notamment de tracer l'historique des actions menées sur une donnée, de gérer le contrôle des accès, de sauvegarder les fichiers dans le coffre-fort des données qui assure leur intégrité, de concevoir et d'appliquer des workflows d'action, d'associer aux données des statuts pour s'assurer de leur situation dans le processus de contrôle qualité, de gérer un ensemble de données au sein d'un projet avec ses règles spécifiques. Une partie de la gestion de la provenance (traçabilité et contexte) est naturellement gérée par les fonctionnalités des systèmes PLM. Les modèles de données des systèmes PLM sont personnalisables, ce qui va nous permettre d'intégrer les spécificités de la neuroimagerie. Notre proposition de modèle de données orienté PLM pour la gestion des données hétérogènes en neuroimagerie est présentée dans le chapitre 4.

Le chapitre suivant traite de la visualisation de graphe. Il présente les concepts de la théorie des graphes, ainsi que les techniques de visualisation statiques et dynamiques existantes, ce qui nous permettra d'orienter notre proposition concernant le problème 2 identifié dans cette thèse :

"comment visualiser les structures de données multidimensionnelles et dynamiques ?".

Chapitre 3

Visualisation de graphe : un état de l'art

Ce chapitre présente l'état de l'art associé au problème 2 de cette thèse : comment visualiser les structures de données complexes et multidimensionnelles ? (voir figure 3.1) Nous avons établi dans le chapitre 1 que la visualisation est une aide cognitive pour l'appréhension de données complexes, et que les graphes permettent de représenter mathématiquement ces données. L'état de l'art s'intéressera aux techniques d'étude de la structure d'un graphe et aux techniques de visualisation existantes, afin d'identifier les verrous scientifiques à lever.

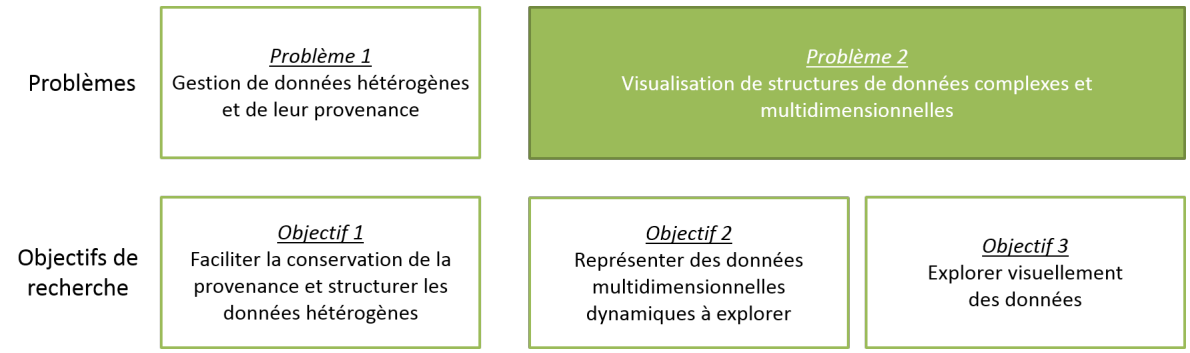


FIGURE 3.1 – Problème 2 : Visualisation de structures de données complexes et multidimensionnelles.

Dans la section 3.1, nous définissons les graphes et leur caractérisation selon la théorie des graphes. Selon Von Landesberger *et al.* (2011), l'analyse visuelle repose sur trois piliers qui sont fortement liés : la représentation visuelle du graphe, l'analyse algorithmique du graphe et l'interaction utilisateur (voir figure 3.2). Aigner *et al.* (2011) définit quant à lui de façon similaire trois questions à se poser pour résoudre un problème de visualisation :

1. Quoi : qu'est-ce qui est présenté ? – caractérisation des données
2. Pourquoi : qu'est-ce que l'utilisateur cherche à voir ? – caractérisation des tâches utilisateur
3. Comment : quelle(s) représentation(s) et quelle(s) technique(s) utiliser ? – caractérisation de la solution de visualisation

Les sections 3.2 et 3.2.3 présentent respectivement les techniques de visualisation actuelles, mêlant représentation visuelle et analyse algorithmique, des graphes statiques et dynamiques – qui est un champ de recherche plus récent, bien que les besoins existent depuis longtemps. La section 3.3 développe les interactions que peut avoir un utilisateur avec les graphes des points de vue formel (tâches de visualisation) et pratique (logiciels de visualisation), ce qui conditionne l'application des techniques de visualisation présentées dans les deux sections précédentes.

Pour finir, l'exploration de la connectivité fonctionnelle cérébrale est présentée dans la section 3.4.

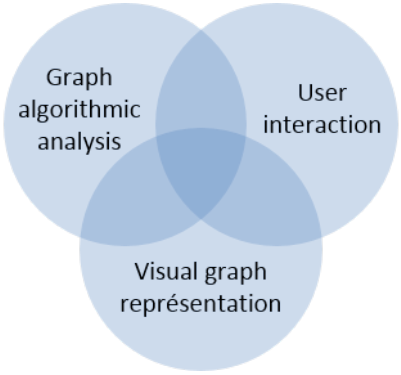


FIGURE 3.2 – Les trois composants de l'analyse visuelle de graphe (Von Landesberger *et al.*, 2011)

Sommaire

3.1 Théorie des graphes

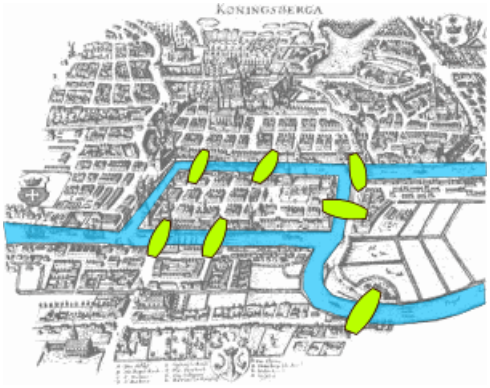
Les graphes sont un outil mathématique puissant pour représenter des problèmes dans de nombreux domaines tels que l'étude des réseaux (électriques, informatiques, sociaux...), la biologie, la chimie ou encore les sciences sociales. La théorie des graphes propose des algorithmes de résolution de ces problèmes d'un point de vue informatique.

Dans cette section sont présentés la définition formelle des graphes et du vocabulaire associé, ainsi que les propriétés topologiques qui sont couramment calculées pour servir de support à l'analyse des graphes.

3.1.1 Les graphes

3.1.1.1 Petite histoire de la théorie des graphes

Le premier article de l'histoire de la théorie des graphes a été publié par Leonhard Euler en 1741 (Euler, 1741). Il exposait un problème lié à la ville de Königsberg, qui possédait à l'époque sept ponts : trouver une promenade qui, partant d'une rive, emprunte une seule fois chacun des ponts et revient au point de départ. Grâce à la modélisation mathématique du problème, Euler a pu montrer qu'un tel chemin était impossible à Königsberg. Ce problème est schématisé dans la figure 3.3 : les nœuds représentent les rives et les arêtes les ponts de la ville. En testant toutes les combinaisons de chemin possibles, il apparait que le chemin passant par toutes les nœuds une seule fois n'existe pas. Depuis lors, le chemin qui passe par toutes les arêtes exactement une fois est appelé chemin *eulérien*.

a) 

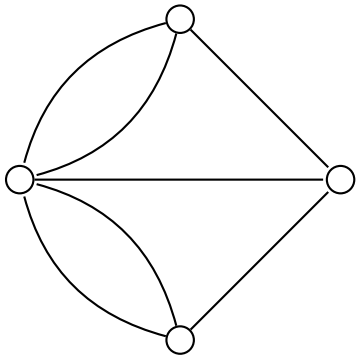
b) 

FIGURE 3.3 – Problème des sept ponts de Königsberg. a) Carte de Königsberg à l'époque d'Euler, avec le fleuve Pregel et les sept ponts (Bogdan Giusca, GFDL, licence CC paternité - partage à l'identique). b) Modélisation du problème sous la forme d'un graphe : chaque nœud représente une rive, et chaque arête un pont.

Cependant, le terme "graphe" n'est introduit pour la première fois qu'en 1878 par James Joseph Sylvester dans un article du journal scientifique *Nature* (Sylvester, 1878), dans le domaine de la chimie. Les mathématiciens se sont intéressés à de nombreux problèmes liés aux graphes au cours des deux siècles précédents : les problèmes de *factorisation de graphe*, les recherches

sur la théorie des *graphes extrémaux* ou encore des *graphes parfaits*[1]. Un des plus connus est le *problème des quatre couleurs*, qui consiste à trouver comment colorier une carte de façon à ce que les zones voisines aient des couleurs différentes.

3.1.1.2 Les différents types de graphes

L'illustration des concepts présentés dans cette sous-section est donnée en figure 3.4.

Un **graphe** est constitué d'un ensemble fini de *nœuds* (ou sommets, *nodes* ou *vertices* en anglais) V et d'un ensemble fini d'*arêtes* (ou arcs, *edges* en anglais) E, tels que :

$$G = (V, E) \text{ avec } E \subseteq \{(u, v) | u, v \in V, u \neq v\} \tag{3.1}$$

où u et v sont deux nœuds quelconques du graphe

Les arêtes connectent les nœuds entre eux. Elles peuvent être *orientées* (ou dirigées, *directed* en anglais) ou *non-orientées* (non-dirigées). Un graphe qui possède des arêtes orientées est appelé *graphe orienté*, respectivement pour des arêtes non-orientées. Dans le cas d'un graphe orienté, les nœuds sont ordonnés par paire pour définir les arêtes : l'arête qui part du nœud u au nœud v est écrit (u, v), tandis que la paire (v, u) représente l'arête qui part du nœud v au nœud u. Un graphe non-orienté dont tous les nœuds son reliés deux à deux par une arête est dit *complet*.

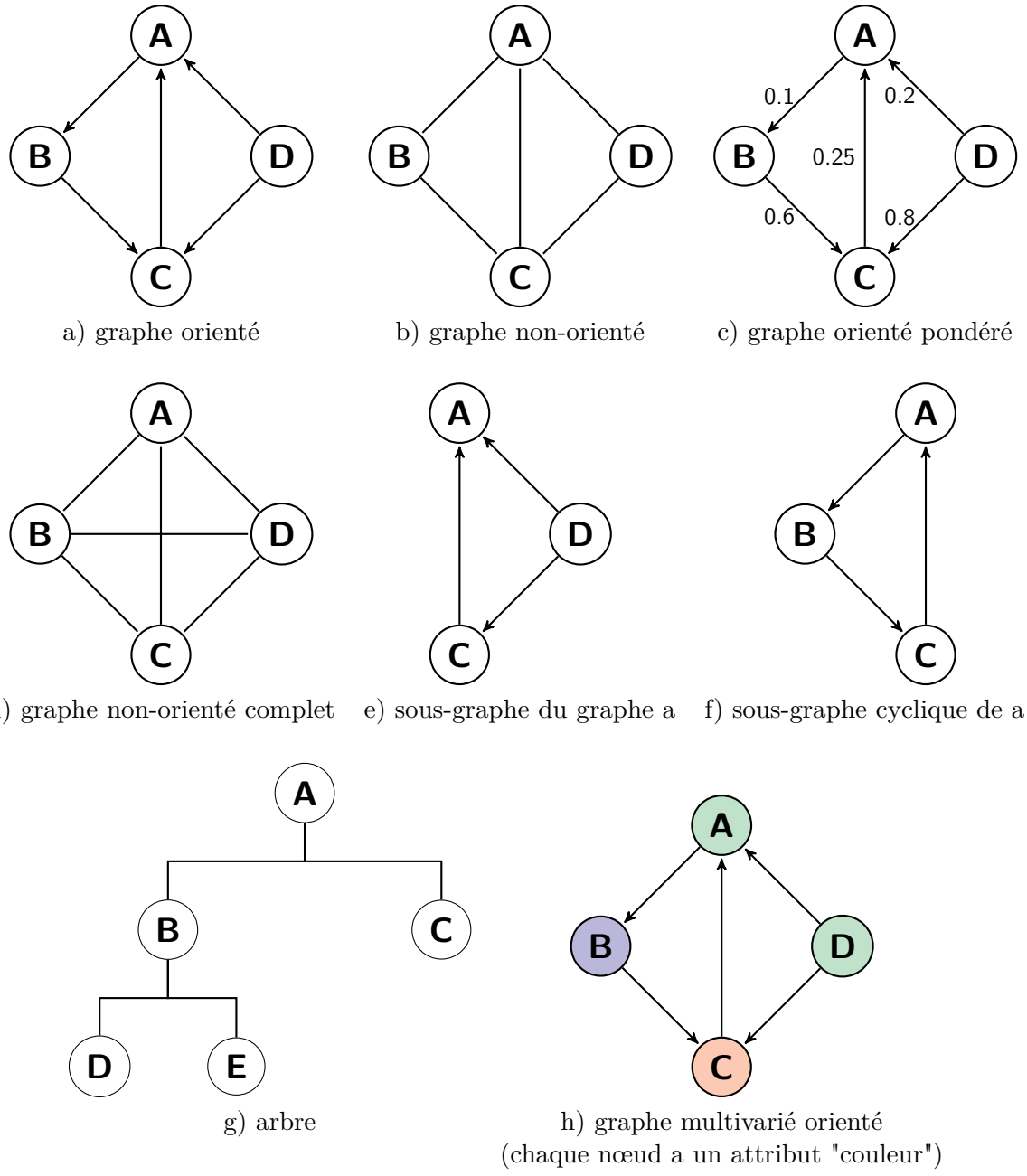
Un **graphe cyclique** présente un *circuit* qui permet de passer par tous les nœuds du graphe et de revenir au point de départ. Quand le graphe n'est pas orienté, le circuit est appelé *cycle* et un graphe qui ne contient aucun cycle est appelé *acyclique*. Un graphe non-orienté qui ne présente pas de cycle est appelé *forêt*, c'est-à-dire qu'il est constitué d'un ensemble d'arbres.

Un **arbre** (ou graphe *hiérarchique*) est un graphe non-orienté et acyclique. Le nœud initial de l'arbre est appelé *racine*, tandis qu'un nœud situé à l'extrémité d'une ramification d'un arbre est appelé *feuille*. Le niveau d'un nœud dans la structure hiérarchique de l'arbre est appelé *profondeur*. Toute une branche de la théorie des graphes est dédiée à l'étude des arbres.

Les **graphes composés** sont des graphes hiérarchiques pour lesquels sont autorisés des arêtes entre des nœuds qui ne sont pas contenus par un parent commun. Formellement, un graphe composé $C = (G, T)$ est défini par un graphe $G = (V, E_G)$ et un arbre $T = (V, E_T, r)$ où r est le nœud racine de l'arbre qui partagent un ensemble d'arêtes, tel que : $\forall e = (v_1, v_2) \in E_G$, $v_1 \notin path_T(r, v_2)$ and $v_2 \notin path_T(r, v_1)$. Les nœuds qui partagent un parent commun dans T appartiennent au même groupe, tandis que les nœuds qui sont reliés à un parent commun dans G partagent une relation générique. Une illustration de graphe composé est donnée dans la figure 3.5.

Une valeur, appelée **poids**, peut être associée à une arête que l'on qualifie alors de *pondérée*

1. Pour plus d'informations sur les grands problèmes de la théorie des graphes, se référer aux ouvrages (Bondy & Murty, 1976) et (West *et al.*, 2001).



a) graphe orienté b) graphe non-orienté c) graphe orienté pondéré

d) graphe non-orienté complet e) sous-graphe du graphe a f) sous-graphe cyclique de a

g) arbre

h) graphe multivarié orienté
(chaque nœud a un attribut "couleur")

FIGURE 3.4 – Illustration des types de graphes (représentation node-link)

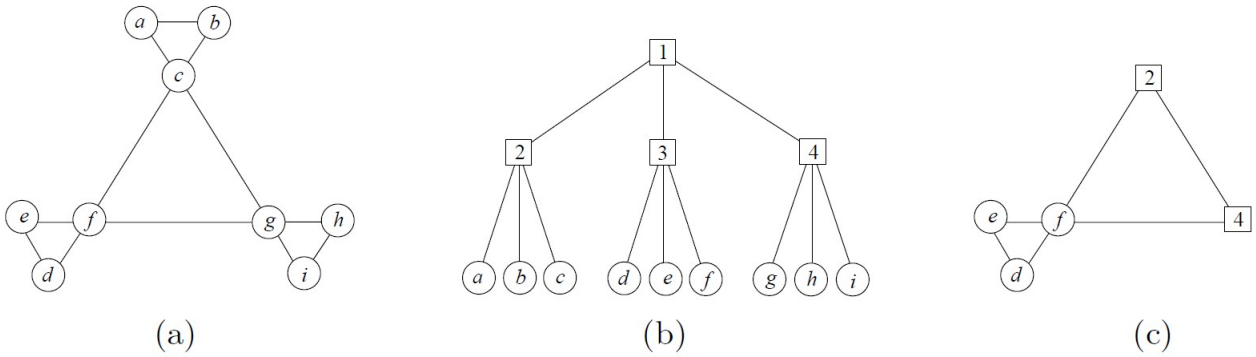


FIGURE 3.5 – (a) Graphe d'entrée G, (b) Arbre après clustering T : les noeuds avec une lettre représentent les noeuds du graphe d'entrée, tandis que les noeuds numérotés représente les noeuds des clusters, (c) Vue fish-eye du graphe composé $C = (G, T)$. (Abello *et al.*, 2005)

(*weighted* en anglais). Par extension, un graphe qui accepte des arêtes pondérées est appelé *graphe pondéré*.

Dans la littérature de la théorie des graphes, un graphe orienté pondéré est généralement appelé **réseau**, tandis que dans le domaine de la visualisation d'information, le terme réseau est généralement associé à un graphe dont des attributs sont associés aux nœuds et aux arêtes. *Dans cette thèse nous choisissons d'utiliser le terme* graphe *quelles que soient ses caractéristiques.*

Un **sous-graphe** g est défini par un ensemble de nœuds appartenant à G :

$$g = (V', E') \text{ avec } E' \subseteq \{(u', v') | u', v' \in (V' \cap V), u' \neq v'\} \tag{3.2}$$

Un **graphe multivarié** M peut être défini comme un graphe G auquel n attributs A sont ajoutés sur les nœuds et les arêtes, tels que (pour les nœuds, la définition est similaire pour les arêtes) :

$$A = \{A_1, ..., A_n\} = (a_{ij})(j = 1...|V|; i = 1...n) \tag{3.3}$$

A_i représente une colonne de la table des attributs (nœud ou arête), et $a^u = (a_{u1}, ..., aun)$ décrit toutes les valeurs des attributs pour le nœud u, en supposant qu'il n'y a pas de donnée manquante (Jusufi, 2013).

Dans un scénario multivarié statique, les valeurs des attributs restent fixes et le challenge consiste à visualiser les interactions entre le ou les graphes et ces attributs. Les graphes statiques multivariés peuvent être vus comme des graphes auxquels sont associés des jeux de données dimensionnels reliés à ses éléments.

Un **graphe dynamique** est défini par une séquence de graphes :

$$\Gamma = (G_1, G_2, ..., G_n) \tag{3.4}$$

où les $G_i = (V_i, E_i)$ sont des graphes statiques dont l'indice réfère à moment temporel $\tau =$

$(t_1, t_2, ..., t_n)$ (Beck *et al.*, 2014).

Les *graphes multivariés dynamiques* (aussi appelés *graphes multivariés temporels*) ont plusieurs attributs des nœuds et/ou des arêtes qui évoluent avec le temps (Abello *et al.*, 2014).

3.1.2 Propriétés topologiques

3.1.2.1 Propriétés des graphes statiques

Les nœuds, les arêtes et le graphe lui-même peuvent être caractérisés par des mesures liées à leur situation dans le graphe. Un panel non exhaustif des propriétés utiles dans la suite du manuscrit est présenté dans cette sous-section. La figure 3.6 illustre certaines de ces propriétés.

Degré d'un nœud : nombre de connexions qui relie le nœud au reste du graphe. Cette mesure est souvent considérée comme fondamentale, et de nombreuses autres mesures s'appuient sur elles. L'ensemble des degrés de tous les nœuds du graphe est appelée distribution des degrés. Dans les graphes générés aléatoirement, les connections sont équiprobables et la distribution des degrés présente une gaussienne centrée symétriquement. Les graphes complexes ne présentent généralement pas une distribution gaussienne, mais une "queue" qui s'étend au niveau des hautes valeurs de degrés. La mesure d'*assortativité* (*assortativity* en anglais) est la corrélation entre les degrés de nœuds connectés. Une assortativité positive indique que des nœuds présentant un fort degré ont tendance à se connecter entre eux.

Distance entre deux nœuds : nombre minimum d'arêtes qui doivent être traversées pour se rendre d'un nœud à un autre. L'ensemble ordonné des arêtes parcourues est appelé **chemin**.

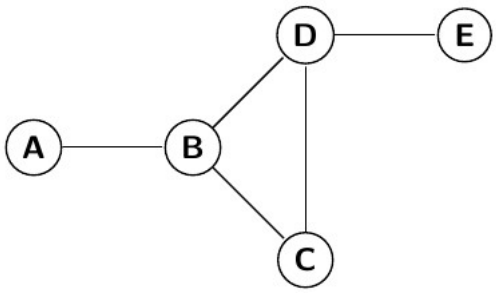
Distance géodésique : nombre d'arêtes formant le chemin le plus court entre deux nœuds.

Excentricité : plus grande distance géodésique entre un nœud i et n'importe quel autre nœud du graphe, autrement dit à quelle distance est le nœud le plus éloigné de i dans le graphe.

Rayon d'un graphe : excentricité minimale de l'ensemble des nœuds, $r = \min_{v \in V} \in (v)$

Diamètre d'un graphe : excentricité maximale de l'ensemble des nœuds, $d = \max_{v \in V} \in (v)$

Distance et efficacité algorithmique : les graphes aléatoires et complexes présentent globalement des distances courtes, tandis que des maillages réguliers présentent des distances plus importantes. L'*efficacité* est inversement reliée à la distance, cependant elle est numériquement plus aisé à utiliser pour estimer les distances topologiques entre des éléments déconnectés d'un graphe. De nombreux algorithmes ont été développés pour déterminer les plus courts chemins entre deux nœuds. La plupart du temps, ces algorithmes émettent l'hypothèse qu'il n'y a pas de boucle dans le graphe. Nous pouvons citer en exemple l'algorithme de Dijkstra – ne fonctionne pas pour les poids négatifs –, l'algorithme de Floyd – le poids d'un chemin orienté



géodésique(A,E) = 3
géodésique(B,E) = 2
excentricité(A) = 3
excentricité(B) = 2
excentricité(C) = 2
excentricité(D) = 2
excentricité(E) = 3

rayon(graphe) = 2
diamètre(graphe) = 3

FIGURE 3.6 – Illustration des propriétés relatives à la distance entre les nœuds d'un graphe (représentation node-link)

est la somme des poids des arêtes constituant ce chemin – et les algorithmes de Fruskal et de Prim qui permettent de trouver pour un graphe l'arbre recouvrant (STP) avec le poids le plus faible.

Densité ou coût de connexion La densité de connexion D est le nombre actuel d'arêtes du graphe $|E|$ au regard de la proportion totale du nombre d'arêtes possible (qui dépend du nombre de nœuds $|V|$) : $D = \frac{2|E|}{|V|(|V|-1)}$. Un graphe complet a une densité de 1. La densité de connexion est un indicateur simple pour estimer le coût physique d'un graphe, par exemple l'énergie ou une ressource nécessaire.

Hubs, centralité and robustesse : la *centralité* (*centrality* en anglais) d'un nœud mesure combien de plus court chemins entre les autres paires de nœuds dans le graphe passe par lui. Les nœuds présentant un fort degré ou une forte centralité sont appelés des *hubs*. Un hub impacte l'efficacité de la communication globale entre les nœuds du graphe, et sa suppression entraine des conséquences majeures sur la topologie du graphe. L'importance d'un nœud individuel sur l'efficacité d'un graphe peut ainsi être évaluée en le supprimant et en étudiant les propriétés du graphe résultant "amputé". La *robustesse* d'un graphe réfère à la fois à l'intégrité structurelle du graphe à la suite de la suppression des nœuds ou des arêtes, et aux effets des perturbations à des niveaux locaux ou globaux du graphe.

Modularité De nombreux graphes complexes sont constitués de modules. Il existe des algorithmes qui permettent d'estimer la modularité d'un graphe, dont la plupart sont basés sur le clustering hiérarchique. Chaque module contient des nœuds densément connectés, et il y a relativement peu de connexions entre des nœuds de différents modules. Les hubs peuvent être décrits en terme de rôle dans cette structure de communautés. Les hubs provinciaux sont principalement connectés aux nœuds de leur module d'appartenance, tandis que les hubs de connexion sont reliés aux nœuds des autres modules.

Cluster : ensemble de nœuds reliés, c'est-à-dire que si les voisins directs d'un nœud sont également connectés entre eux, ils forment un cluster. Le coefficient de clustering quantifie le

nombre de connexions qui existent entre les voisins les plus proches d'un nœud et le nombre possible maximal de connexions. Pour Chen *et al.* (2012), utiliser un algorithme de détection de cluster de voisinage est une façon d'abstraire la complexité d'un graphe. Les réseaux aléatoires ont un faible clustering, tandis que les graphes complexes présentent un fort clustering (associé à une efficacité de transfert d'information et de robustesse forte). Les interactions entre des nœuds voisins peuvent également être quantifiées en comptant les occurrences de petits motifs de nœuds inter-connectés. La distribution de différentes classes de motifs au sein d'un graphe fournit une information à propos du type d'interactions locales que le graphe peut supporter. De façon locale, une **clique** (pour un graphe non-orienté) est un ensemble de nœuds d'un graphe dont le sous-graphe induit est complet. Deux sommets quelconques de la clique sont donc toujours adjacents, c'est-à-dire connectés par une arête. Un état de l'art des techniques de clustering est présenté dans Schaeffer (2007).

Small-world (aussi appelé *effet du petit monde*) : les nœuds du graphe ne sont pas tous reliés entre eux et pourtant tous les nœuds peuvent être atteints depuis n'importe quel autre nœud en une distance *faible*. Cette distance est proportionnelle au logarithme du nombre de nœuds dans le graphe.

Un graphe est dit **complexe** lorsqu'il présente certaines caractéristiques topologiques telles qu'un haut clustering, un petit-monde, la présence de nœuds à haut degré et de hubs, l'assortativité – la tendance des nœuds du réseau à être reliés avec d'autres nœuds qui ont des caractéristiques similaires –, la modularité ou encore la hiérarchie. Les graphes générés aléatoirement ou les treillis réguliers ne présentent jamais ces propriétés, à l'inverse de la plupart des réseaux réels qui sont complexes par cette définition.

3.1.2.2 Propriétés des graphes dynamiques

Bien que la théorie des graphes ait principalement développé des mesures pour les graphes statiques, certaines propriétés statiques ont été étendues à la dynamique, et d'autres créés pour décrire les changements de structure entre états temporels.

Comme la plupart des métriques sont basées sur l'adjacence des nœuds, qui varie dans les graphes dynamiques, il apparait naturel d'adapter ces premières. L'extension des mesures classiques (statiques) de la théorie des graphes aux graphes dynamiques est discuté dans Nicosia *et al.* (2013). Les définitions des mesures découlent La **longueur temporelle** ou durée est mesurée comme l'intervalle entre le premier et le dernier contact entre deux nœuds. Le **plus court chemin temporel** entre entre deux nœuds i et j est défini comme le chemin temporel connectant i et j qui a la longueur temporelle la plus petite. Par suite la **distance temporelle** $d_{i,j}$ entre deux nœuds i et j est la longueur temporelle du plus court chemin temporel entre i et j. Pour l'intégralité de la définition des mesures, se référer à Nicosia *et al.* (2013). L'application des mesures temporelles sur des systèmes réels est présentée dans Tang *et al.* (2013).

Uddin *et al.* (2014) introduisent une nouvelle façon d'étudier le degré de centralité d'un nœud au sein d'un graphe dynamique. Ils appellent cette mesure **Time Scale Degree Centrality** (TSDC), qui est valable pour des réseaux orientés et à la taille variable. Le TSDC *sortant* (*out_degree*) pour un nœud i est donné dans l'équation ci-dessous, où N est le nombre de nœud du réseau, T la durée totale du réseau, x_{ij} de la valeur de la cellule de la matrice d'adjacence X du réseau et t_i le moment à partir duquel le nœud i peut être pris en considération.

$$TSDC_i(out\_degree) = \frac{\sum_j x_{ij}(T - t_i)}{(N-1) * T} \tag{3.5}$$

Les auteurs ont montré que la mesure avait intérêt d'un point de vue microscopique par rapport à une analyse simple de l'évolution du degré de centralité des nœuds au cours du temps. La principale limite de cette mesure est qu'elle prend en compte la durée de connexion qu'à partir du moment de son apparition, et le cas d'une disparition simple ou multiple (apparition-disparition successives) n'est pas pris en compte. La mesure n'est donc applicable que pour des réseaux orientés à la dynamique additives.

La **centrality dynamique** peut être calculée pour identifier les nœuds qui ont le plus d'influence sur une période temporelle donnée, en s'appuyant sur (Lerman *et al.*, 2010).

Federico *et al.* (2012) ont développé la mesure du **Change Centrality** (CC). Cette mesure permet la comparaison deux à deux d'états successifs d'un réseau dynamique. Son objectif est de fournir pour chaque nœud une mesure des changements qui interviennent à la fois au niveau microscopique et macroscopique.

$$CC_{t_1,t_2}(i) = \sum_{n=0}^{e_i} \frac{1}{2^{n+1}} r^n_{t_1,t_2}(i) \tag{3.6}$$

Entre deux instants t_1 et t_2, avec $e_i = max_{t \in t1,t2} e_t(i)$ est l'excentricité maximale du nœud i et $r^n_{t_1,t_2} = \frac{|N^n_{t1}(i) \Delta N^n_{t2}(i)|}{|N^n_{t1}(i) \cup N^n_{t2}(i)|}$ le ratio du changement à une distance n (avec $N_t(i) = \{j \in V : d_t(i,j) = 1\}$ l'ensemble des nœuds i à une distance donnée au temps t) La mesure obtenue est normalisée et positive.

Pour les auteurs, s'il est important de mettre en relief les changements d'un état sur l'autre, il est également important de minimiser les changements non nécessaires pour aider à la préservation de la carte mentale de l'utilisateur. Le CC doit aider au calcul du layout de présentation des résultats.

Il existe un vocabulaire et des **propriétés topologiques** associés aux graphes.
Les réseaux réels présentent souvent les propriétés des graphes complexes : haut clustering, petit-monde, présence de hubs, assortativité, modularité et hiérarchie.
Les propriétés topologiques statiques peuvent être étendues aux graphes dynamiques afin de **décrire les changements de structure entre états temporels**.

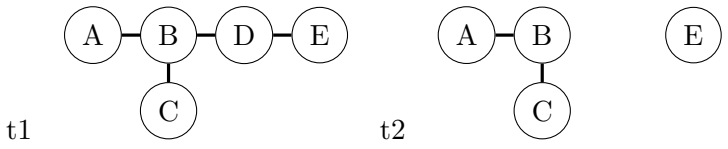


FIGURE 3.7 – Exemple d'un réseau dynamique à deux états temporels successifs (t1, t2) (Federico *et al.*, 2012)

3.2 Techniques de visualisation de graphes

Le domaine de l'information visualisation a produit beaucoup de techniques de visualisation pour les graphes statiques. Des états de l'art très complets ont été publiés sur la visualisation de graphes en général (Herman *et al.*, 2000; Von Landesberger *et al.*, 2011) et des graphes dynamiques en particulier (Beck *et al.*, 2014). Cette section ne présente pas de façon exhaustive les techniques existantes, mais expose celles qui présentent un intérêt dans le cadre de cette thèse.

Les différentes représentations d'un graphe sont présentées, suivies d'une introduction aux algorithmes de layout basés sur la topologie et sur les attributs d'un graphe. Pour finir, les techniques de visualisation de graphes dynamiques sont présentées.

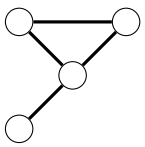
3.2.1 Représentation d'un graphe

Si Euler n'a pas utilisé de représentation de graphe pour poser et résoudre son problème des sept ponts de Königsberg (voir 3.1.1.1), l'intérêt de la communauté pour la représentation visuelle des graphes n'a cessé de croitre depuis la fin du XVIIIe siècle. La discipline du *dessin de graphe* (ou *graph drawing* en anglais) s'intéresse à cette question. Initialement les représentations visuelles sous forme de graphes étaient surtout utilisées à des fins pédagogiques ou lors de présentations. Une histoire de l'évolution du dessin de graphe depuis ses premières apparitions jusqu'aux représentations modernes est proposée par Kruja *et al.* (2002). Il existe aujourd'hui de nombreuses façons de représenter les nœuds et les arêtes d'un graphe, leurs attributs et leurs propriétés, qui sont présentées de façon non-exhaustive dans cette sous-section.

3.2.1.1 Node-link vs matrices

Les deux représentations de graphe les plus utilisées sont le diagramme node-link et les matrices, illustrées dans la figure 3.8. Dans un **diagramme node-link**, les nœuds sont représentés par des disques et les arêtes par des traits. Selon Henry *et al.* (2007), les diagrammes node-link ont pour avantage :

- d'être intuitifs pour l'utilisateur ;
- d'être efficaces sur des petits graphes ;
- de permettre d'effectuer facilement des tâches complexes, comme par exemple suivre un chemin entre deux nœuds non adjacents ;
- d'être relativement compacts.



$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

	1	2	3	4
1				
2				
3				
4				

a- node-link b- matrice d'adjacence b- tableau matriciel (pondéré)

FIGURE 3.8 – Différentes représentations d'un graphe à quatre nœuds et quatre arêtes

Cependant, pour les graphes denses, le chevauchement des nœuds et le croisement des arêtes posent des problèmes de lecture du graphe. De façon générale l'usage de la représentation node-link requière l'application d'un layout, ce qui signifie qu'un calcul doit être effectué avant la visualisation. Certains types de graphes disposent naturellement d'un layout et ne nécessitent pas obligatoirement d'en calculer un nouveau : par exemple les graphes possédant des références géographiques ou bien anatomiques comme dans le cas de graphes de connectivité fonctionnelle cérébrale. Ces layouts "naturels" peuvent néanmoins poser des problèmes comme le croisement inévitable des arêtes, et empêcher d'identifier visuellement des caractéristiques topologiques du graphe.

Une **matrice** (appelée parfois *matrice d'adjacence* ou *matrice de proximité*) indique les liaisons entre chaque paire de nœuds dans le graphe. Elle permet de mettre en évidence comment les nœuds sont adjacents, c'est-à-dire la façon dont ils sont connectés entre eux. Cette représentation a été introduite dans les années 60 par Bertin (1973).

Le matrice d'adjacence d'un graphe $G = (V, E)$ est définie par une matrice nxn $D = (d_{ij})$ où n est le nombre de nœuds dans G, $V = \{v_1, ..., v_n\}$ et d_{ij} est le nombre d'arêtes entre v_i et v_j. S'il n'existe aucune arête (v_i, v_j) dans G, alors $d_{ij} = 0$. Si le graphe est non-orienté, la matrice d'adjacence est symétrique : $D_T = D$.

Intuitivement, les matrices d'adjacence sont utiles pour visualiser efficacement des relations, mais pas pour étudier des corrélations entre entre les relations et les propriétés des nœuds du graphe : seuls les attributs sur les arêtes peuvent être représentés visuellement. Les avantages principaux des matrices sont (Heymann, 2013) :

- pas d'occlusion visuelle, car les nœuds et les arêtes ne se recouvrent pas.
- efficacité de réalisation de tâches simples comme identifier le nœud le plus connecté, un lien entre deux nœuds ou un voisin commun entre deux nœuds. Cette efficacité est supérieure pour les matrices vis à vis des diagrammes node-link à partir de vingt nœuds (Ghoniem *et al.*, 2004).
- lisible pour des graphes grands et denses.

Les matrices sont peu adaptées aux graphes clairsemés, et surtout elles permettent difficilement de réaliser des tâches complexes comme le suivi d'un chemin entre deux nœuds (Ghoniem *et al.*, 2004). Il est nécessaire de disposer d'un espace de visualisation suffisamment large pour les afficher en entier. Par ailleurs, une organisation aléatoire des nœuds dans la matrice ne permet

pas toujours la mise en évidence de motifs – par exemple des clusters –, et il est donc quasiment indispensable d'appliquer un algorithme permettent de réordonner par permutations successives les lignes et colonnes de la matrice avant de la visualiser. Ces inconvénients sont probablement la cause de la faible utilisation de la représentation matricielle par rapport aux diagrammes node-link.

Des représentations **hybrides** ont été développées pour essayer de bénéficier des avantages combinés des matrices et des diagrammes node-link. NodeTrix (Henry *et al.*, 2007) répond à un problème spécifique de la visualisation des réseaux sociaux : les graphes sont localement clairsemés mais localement très denses, ce qui rend difficile leur analyse. La grande majorité des systèmes qui permettent la visualisation des réseaux sociaux proposent une représentation node-link, bien qu'elle soit uniquement adaptée à la visualisation de graphes clairsemés. NodeTrix permet d'intégrer le meilleur des deux représentations traditionnelles des graphes en utilisant les diagrammes node-link pour visualiser la structure globale du réseau, à l'intérieur duquel les matrices d'adjacence montrent les communautés. Un exemple de visualisation avec NodeTrix est donné dans la figure 3.9 (Henry *et al.*, 2007).

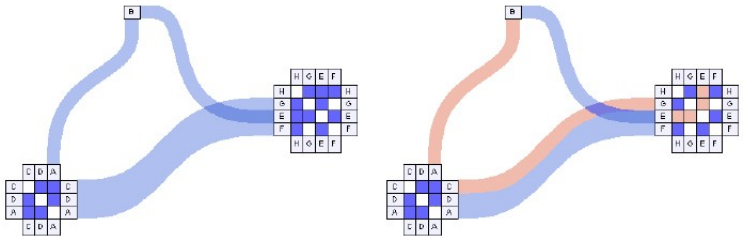


FIGURE 3.9 – Deux façons de visualiser les connexions entre nœuds dans une représentation combinant les matrices et les diagrammes node-link, avec le logiciel NodeTrix (Henry *et al.*, 2007)

3.2.1.2 Représentation des attributs

Cette thèse s'intéresse à la visualisation de données hétérogènes et multidimensionnelles, ce sont donc à la fois la structures et les attributs du graphe qui doivent être explorés. Il existe différents types d'attributs à visualiser (Jusufi, 2013) :

- Les attributs appartenant aux éléments du graphe : nœuds et arêtes,
- Les données dérivées calculées à partir des attributs sur les éléments du graphe,
- Les propriétés topologiques du graphe, calculées à partir de la structure du graphe.

Les correspondances couramment utilisées entre les variables de données d'un graphe et les variables visuelles appliquées sur les diagrammes node-link sont présentées dans le tableau 3.1 (Heymann, 2013). Pour Chen *et al.* (2012), l'utilisateur doit pouvoir encoder n'importe quel attribut des données comme une propriété visuelle, telle que la couleur, la taille, la transparence, le type de police... Un ensemble de ces propriétés encodées est appelé *style visuel*.

Variables de données	Variables visuelles
nœud	point
label du nœud	texte à proximité du point correspondant
arête	segment de ligne ou de courbe
attribut qualitatif	couleur du point
attribut quantitatif	taille du point

TABLE 3.1 – Correspondances usuelles entre les variables de données d'un graphe et les variables visuelles appliquées (Heymann, 2013)

Pour les graphes multivariés, les arêtes peuvent également être le support de la visualisation des attributs. Les glyphes permettent aussi de visualiser des attributs supplémentaires. Par exemple le logiciel PivotGraph (Wattenberg, 2006) a été conçu pour les graphes multivariés. Il ne s'appuie pas sur l'étude de la topologie globale du graphe, mais positionne les nœuds sur une grille afin de se focaliser sur les relations les attributs des nœuds et leurs connexions, en jouant sur les tailles et les couleurs des éléments du graphes (nœuds et arêtes). Une visualisation avec le logiciel PivotGraph est présenté dans la figure 3.10 ; les arêtes sont représentées par différentes formes, tailles et couleurs.

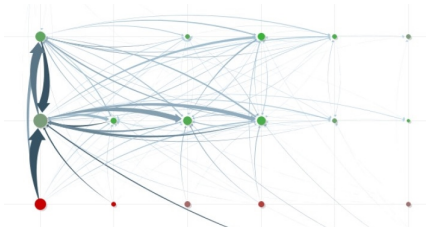


FIGURE 3.10 – Réseau de communication de salariés d'une entreprise. L'axe X représente les divisions de l'entreprise, et l'axe Y les bureaux géographiques. La division de la colonne la plus à gauche présente beaucoup plus de communications entre les bureaux que les autres. Visualisé avec PivotGraph (Wattenberg, 2006)

3.2.1.3 Préparation des graphes

Des techniques de réduction de graphe permettent de simplifier les données en taille et en complexité. Il existe deux principales approches pour réduire un graphe (Von Landesberger *et al.*, 2011) :

- Filtrage : une méthode stochastique (sélection aléatoire) ou un algorithme déterministe (sélection sur les attributs ou les propriétés) peuvent être appliqués pour déterminer les éléments du graphe à enlever (nœuds ou arêtes).
- Agrégation : les éléments du graphe (nœuds ou arêtes) sont fusionnés afin de réduire la taille et/ou la complexité du graphe. L'agrégation va permettre de jouer sur les échelles de visualisation, et permettre une plus grande interactivité. La figure 3.11 illustre les niveaux d'agrégation d'un graphe qui peuvent être représentés par un graphe hiérarchique. Parfois l'agrégation peut simplement être une délimitation visuelle pour aider à appréhender la compréhension du graphe, comme montré sur la figure 3.15 (voir la sous-section 3.2.2

portant sur l'esthétique des graphes).

Le pré-traitement implique l'application d'un layout, qui indique l'agencement des nœuds dans l'espace. Les layouts existant pour la représentation node-link sont présentés dans la sous-section suivante.

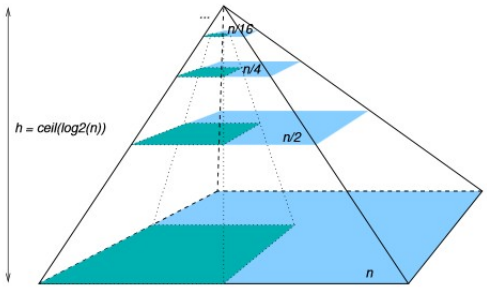


FIGURE 3.11 – Structure conceptuelle d'un graphe ayant plusieurs niveaux d'agrégation (Elmqvist *et al.*, 2008)

3.2.2 Agencement des diagrammes node-link

Cette sous-section introduit les familles de layouts utilisés pour la représentation node-link. Di Battista *et al.* (1994) est considéré comme la référence introduisant une approche algorithmique au dessin de graphe. Un état de l'art très complet présentant les caractéristiques des principaux algorithmes de layout est proposé par (Gibson *et al.*, 2013) ; il constitue un document de référence pour le choix d'un algorithme adapté à un problème donné : taille et propriétés du graphe. Il n'y a pas de meilleure façon de représenter un graphe, car la technique la plus adaptée dépend grandement des informations recherchées (Blythe *et al.*, 1996).

Layouts de force Les *layouts de force* (ou layouts énergétiques) ont été parmi les premiers à être développés pour un calcul automatique de l'agencement des nœuds d'un graphe et sont parmi les plus couramment utilisés aujourd'hui (Gibson *et al.*, 2013). Basés sur un modèle physique d'attraction et de répulsion, le but est d'agencer les nœuds du graphe de façon optimale. Le cas d'un graphe modélisé par un système de ressort est présenté dans la figure 3.12 ; les flèches indiquent les directions d'attraction des noeuds qui vont permettre d'obtenir la configuration stable finale.

Kobourov (2012) propose une étude comparative concentrée sur les algorithmes de force existants. Dans Gibson *et al.* (2013), un tableau présente les caractéristiques – performance, esthétique, distribution, taille – des algorithmes pour les diagrammes node-link (incluant les algorithmes de force), ce qui permet de comparer facilement ces derniers par rapport aux autres approches. Il existe deux familles d'algorithmes de force :

– Basés sur le principe de l'algorithme de Eades (1984), l'objectif est d'atteindre un équilibre. Fruchterman & Reingold (1991) ont proposé une adaptation de l'algorithme de Eades pour prendre d'avantage en compte l'esthétique du graphe : réduction du croisement des arêtes, uniformisation de la taille des arêtes, recherche de la symétrie...

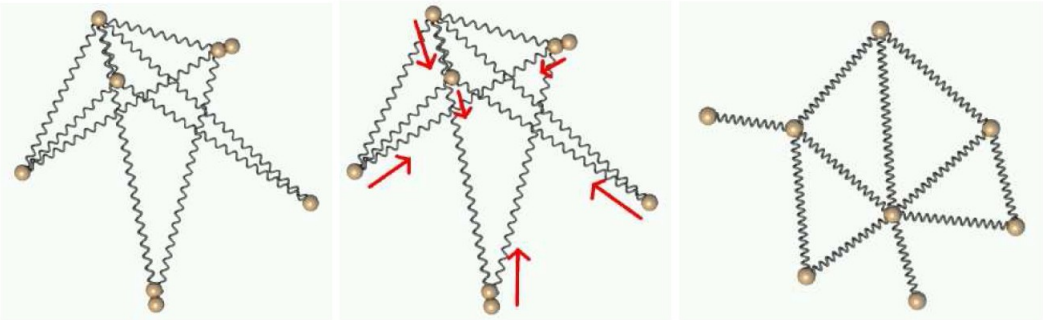
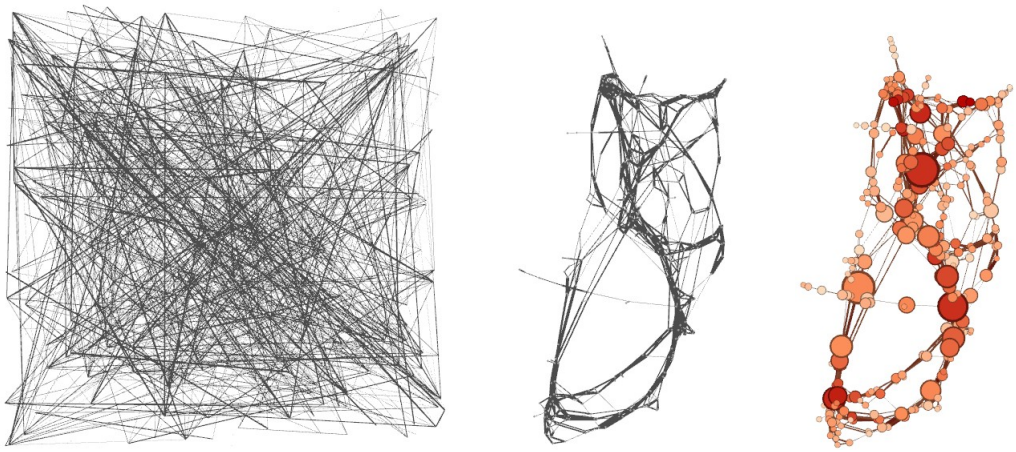


FIGURE 3.12 – Illustration du principe de système de ressorts : en partant de positions aléatoires, les ressorts du système vont chercher à retourner dans une configuration stable (Kobourov, 2012)



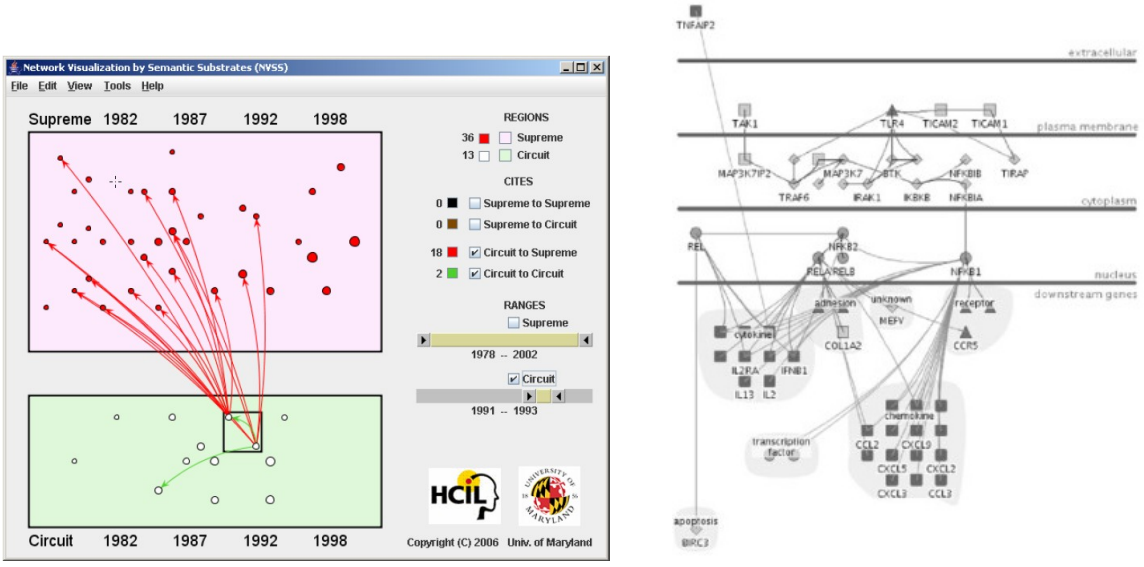
(a) (b) (c)

FIGURE 3.13 – Layout de force 400 nœuds, 400 arêtes. (a) Layout aléatoire, (b) Layout de force, (c) Layout de force et affichage de propriétés sur les noeuds du graphe : taille=degré, couleur=centralité betweeness.

– Problèmes d'optimisation qui cherchent à minimiser une fonction. Les premiers de ces algorithmes sont ceux de Kamada & Kawai (1989) et Davidson & Harel (1996).

Les algorithmes de force ont été améliorés au cours des années (voir Gibson *et al.* (2013) pour un état de l'art exhaustif), mais ils présentent encore des faiblesses au-delà de cinquante nœuds en termes de résultat et de performances : la plupart de ces algorithmes ont été développés pour leurs fonctionnalités et non pas dans une optique de performance, si bien que l'exécution d'un algorithme de force est souvent gourmand en temps.

Les layouts *multi-échelle* (ou multi-niveau, *multi-scale* en anglais) sont une technique qui permet de rendre les algorithmes de force plus efficaces sur les graphes de plus de quarante nœuds (Noack, 2004). Le principe est de commencer par optimiser le layout sur un graphe "grossier", puis de propager le layout obtenu dans le graphe original. Le graphe "grossier" est obtenu par des techniques de réduction.



(a) (b)

FIGURE 3.14 – (a) Affaires judiciaires entre 1991 et 1993 visualisées avec la méthode des substrats sémantiques. (rouge) cours suprêmes des Etats-Unis, (vert) cours fédérales (Shneiderman & Aris, 2006). (b) Visualisation de molécules avec le layout Cerebral : petit réseau TLR4 de 57 nœuds et 74 arêtes (Barsky *et al.*, 2007).

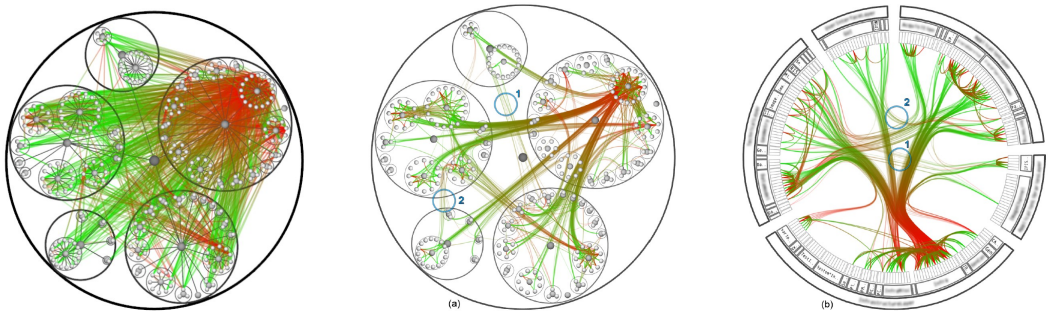
Layouts à partir des attributs des nœuds Il est courant d'utiliser des attributs de nœuds pour calculer le layout du graphe, que ce soit pour imposer un ensemble de restrictions sur le placement des nœuds, pour utiliser l'appartenance à un groupe pour positionner les nœuds, ou encore pour établir une correspondance directe entre la position du nœud et un de ses attributs.

Les techniques de **layout à contraintes** imposent des critères de placement définis par l'utilisateur pour une partie ou l'ensemble des nœuds, en plus d'utiliser un algorithme de layout plus classique. Les contraintes sont utilisées pour améliorer un layout existant tout en préservant sa topologie (Dwyer *et al.*, 2009). Elles peuvent consister à la fixation de la position d'un nœud, la séparation de certains groupes de nœuds ou le maintien du layout d'un sous-graphe. Elles proviennent souvent des attributs ou des propriétés des nœuds.

Les premières méthodes de layout à contraintes ont été proposées par Sugiyama *et al.* (1981) et permettent d'agencer les graphes hiérarchiques en fonction des attributs des nœuds. De nombreuses techniques s'en inspirent. Une autre technique est celle des substrats sémantiques : les substrats sont des zones indépendantes dans lesquelles les nœuds sont positionnés en fonction de leurs attributs, comme illustré dans la figure 3.14.a (Shneiderman & Aris, 2006). Barsky *et al.* (2007) visualise des molécules qui sont séparées en couches, en fonction de leur localisation cellulaire ; cette technique est illustrée dans la figure 3.14.b avec un petit et un grand graphe. Une approche classique en visualisation est de proposer à l'utilisateur une vue d'ensemble du graphe et des vues locales de parties du graphe, ce qui pose des problèmes de correspondance dans entre les layouts des vues respectives ; Dwyer *et al.* (2008) utilise pose des contraintes pour le calcul des vues locales à partir du calcul du layout de la vue générale.

Layouts pour le clustering et graphes composés Le clustering est souvent représenté comme un arbre, soit à l'aide de couches qui indiquent les niveaux de hiérarchie, soit comme une carte proportionnelle abrégée (treemap) (Van Wijk & Van de Wetering, 1999) qui permet de visualiser le graphe dans un espace rectangulaire réduit. Les diagrammes node-link classiques peuvent être adaptés avec des contraintes pour représenter les groupes hiérarchiques de nœuds avec des *super-nœuds* ou des sections de layout radial, comme présenté dans la figure 3.15. Les arêtes peuvent également être représentées à différents niveaux hiérarchiques, et la technique de regroupement des arêtes en faisceau assure une bonne lisibilité du graphe (figure 3.15.b-c).

Les graphes composés peuvent être représentés en deux graphes (générique et hiérarchique) par des diagrammes node-link classiques, mais aussi sous la forme de layouts multi-niveaux. Abello *et al.* (2005) propose la possibilité de parcourir indépendamment les niveaux du graphe hiérarchique, ainsi que de créer des vues réduites de l'intersection des vues et des relations hiérarchiques , qui sont appelée vues composées fisheye. Le principe de la technique est illustré dans la figure 3.16.



a b c

FIGURE 3.15 – Diagramme node-link hiérarchique : (a) layout à contraintes hiérarchique, (b) réduction des arêtes en faisceaux, (c) layout radial hiérarchique avec réduction des arêtes en faisceaux (Holten, 2006)

Layouts en trois dimensions Les algorithmes développés pour un affichage en deux dimensions ne sont pas toujours généralisables à un affichage en trois dimensions. Les problématiques d'occlusion des nœuds et de croisement des arêtes ne sont pas les mêmes en trois dimensions (Herman *et al.*, 2000; Teyseyre & Campo, 2009). Par ailleurs, l'usage d'un espace en trois dimensions implique une navigation interactive de la part de l'utilisateur, et le résultat de la visualisation peut difficilement être partagé dans un mode statique.

Esthétique des graphes L'un des objectifs principaux des layouts est de trouver des positions spatiales aux nœuds qui maximisent certaines mesures de *désirabilité* (Coleman & Parker, 1996). Quand le layout va servir à un être humain, la mesure de la désirabilité est appelée une *esthétique* (*aesthetic*) . Pour Bennett *et al.* (2007), la création de graphes esthétiquement attractifs est plus qu'une quête pour la beauté, elle a pour but de révéler la structure et la

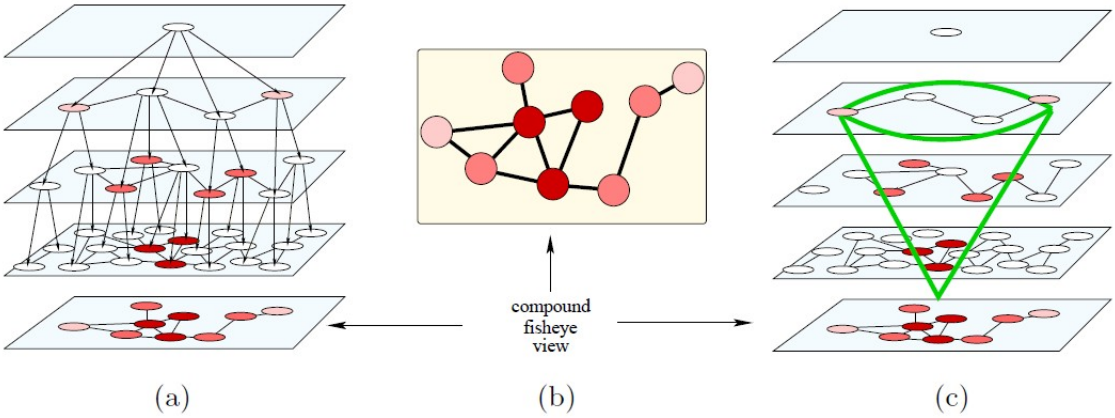


FIGURE 3.16 – Vue fisheye-composée d'un graphe clusterisé : (a) Vue multi-niveaux du graphe clusterisé, le nœuds colorés forment la vue du bas ; (b) Vue fisheye-composée obtenue à partir de trois niveaux de hiérarchie ; (c) Vue conceptuelle : intersection d'une vue multi-niveaux avec un cône inversé (Abello *et al.*, 2005)

signification sous-jacente du graphe, ce qui signifie que l'esthétique d'un graphe sert avant tout et surtout à la lisibilité des graphes.

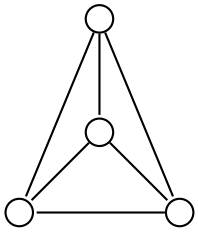
Simplement dessiner un graphe n'est pas suffisant, car la façon dont le graphe est présenté a un impact sur la façon dont le graph est compris ; en accord avec les principes de la théorie Gestalt de la proximité [2] (Smith, 1988), les développeurs d'algorithmes de layouts doivent être conscients que des nœuds placés l'un à côté de l'autre seront interprétés par l'utilisateur comme une réelle relation, que cette relation existe effectivement ou pas (Gibson *et al.*, 2013). Cela signifie que le layout et l'arrangement des nœuds influence fortement comment l'utilisateur perçoit les relations dans le graphe.

La lisibilité d'un graphe est définie par la propriété intrinsèque d'un affichage de graphe à être compréhensible par un être humain Bennett *et al.* (2007). Les *heuristiques esthétiques* (*aesthetic heuristics* en anglais) pour le dessin des graphes sont conçues pour améliorer la lisibilité et la compréhension des graphes. Quand les graphes sont larges et densément connectés, il y a deux difficulté majeures pour rendre un layout lisible (Purchase, 2002) :

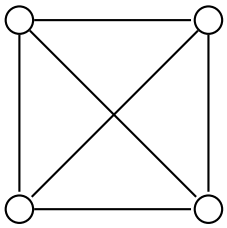
- Problèmes syntaxiques (ou structurels) : éviter l'occlusion d'éléments à cause du chevauchement des arêtes et des et empêcher les arêtes de devenir trop longues et tordues.
- Problèmes sémantiques (ou spécifiques à un domaine) : mettre en exergue les caractéristiques les plus importantes du modèle sous-jacent. Ces problèmes sont dépendant des tâches effectuées par l'utilisateur.

Des critères peuvent parfois être en concurrence et il n'est pas possible de les satisfaire tous à la fois, comme l'illustre la figure 3.17 : le graphe (a) prend en compte le critère esthétique de non-chevauchement des arêtes, tandis que le graphe (b) présente une symétrie parfaite, au détriment du croisement de deux arêtes.

2. Aussi appelée psychologie de la forme. Les formes sont perçues comme des ensembles structurés et non comme une juxtaposition d'éléments.



a- Optimisation du chevauchement des arêtes



Optimisation de la symétrie

FIGURE 3.17 – Les critères d'optimisations de l'esthétique d'un graphe sont parfois en concurrence

Bennett *et al.* (2007) propose in 2007 une intéressante étude sur les heuristiques esthétiques et leur évaluation, centré sur les graphes à représentation node-link. Selon les auteurs, les formes, la taille, la texture et la couleur des nœuds et des arêtes sont encore peu exploités dans la conception d'heuristiques esthétiques, bien qu'ils pourraient jouer un rôle important dans la lisibilité des graphes. Récemment, Becker *et al.* (2014) s'est intéressé à l'influence de la forme et de la couleur sur les performances de recherche visuelle, ce qui dépend des tâches effectuées par l'utilisateur et de son entrainement.

3.2.3 Techniques de visualisation des graphes dynamiques

Le monde réel évolue constamment. Pour des raisons de simplification, les données sont étudiées de façon statique : les graphes statiques sont souvent une réduction des données et représentent soit une partie (un point dans le temps), soit une agrégation de tous les moments temporels possibles.

Dans les graphes dynamiques (ou temporels), les nœuds et les arêtes apparaissent, disparaissent et parfois réapparaissent ; les valeurs des attributs évoluent. Ces changements bas-niveau sont responsables de changements haut-niveau : l'émergence de nœuds centraux, la fusion ou la division de deux clusters, le diamètre du graphe, etc (Bach *et al.*, 2014b).

Dans la suite de la thèse, nous ne faisons pas de distinctions entre un instant t_i et un intervalle (agrégation autour de l'instant t_i), tel que le décrit Aigner *et al.* (2011) : un graphe dynamique est une succession de moments temporels.

Le domaine de la visualisation de graphes dynamiques est relativement récent, car l'étude des graphes statiques a longtemps été privilégiée à cause de limites computationnelles et de raisonnement. Il connait cependant un fort engouement dans la communauté de la visualisation de l'information : les publications scientifiques relatives à la visualisation de graphes dynamique a quadruplé entre 2006 et 2012 (Beck *et al.*, 2014). Aujourd'hui, de nombreux travaux de recherche sont effectués dans le but de dépasser les limitations des graphes statiques par l'étude des problèmes dynamiques dans leur ensemble, ce qui permet de comprendre pourquoi et comment les graphes statiques sont atteints, mais aussi éventuellement de prévoir des événements futurs.

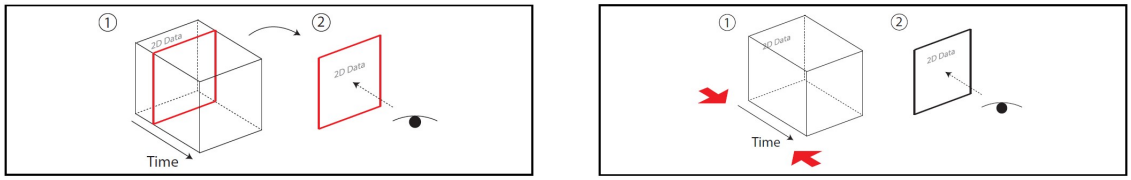
Dans un premier temps sont présentées les principales techniques de visualisation de graphes dynamiques identifiées dans la littérature. Ensuite les aspects de préservation de la carte mentale pour l'exploration des graphes dynamiques sont développés.

3.2.3.1 Représentation des graphes dynamiques

Comme pour les graphes statiques, les représentations principales sont les diagrammes node-link et matrices. Ces dernières sont encore moins couramment utilisées pour les graphes dynamiques que pour les graphes dynamiques, comme le montre l'état-de-l'art mené par Beck *et al.* (2014) : moins de dix publications sur les graphes dynamiques concernent la visualisation de matrices depuis les années 90, quand les digrammes node-link sont présents dans plusieurs dizaines de publications (voir la figure 3.20).

Pour simplifier les données à visualiser, de même que pour les graphes statiques, des techniques de réduction peuvent être appliquées aux graphes dynamiques. Bach *et al.* (2014b) en a dressé une liste :

- Time cutting (découpage temporel) : un moment dans le temps est extrait du graphe dynamique (voir figure 3.18).
- Time flattening (aplatissement temporel) : agrégation linéaire ou discrétisée, de la totalité ou d'une partie des événements du graphe dynamique (voir figure 3.18).
- Colored time flattening (aplatissement temporel coloré) : des couleurs sont assignées à chaque moment dans le temps, ce qui apparait sur le graphe agrégé.
- Time juxtaposing (juxtaposition temporelle) : les moments temporels sont affichés côte à côte.
- Space cutting and flattening (découpage et aplatissement spatial) : les opérations sont similaires aux découpage et aplatissement temporels, mais ont lieu dans l'espace, comme par exemple l'évolution d'un nœud dans le temps.
- Sampling (échantillonnage) : une partie des données est prélevée pour être visualisée.



a. Opération time cutting b. Opération time flattening

FIGURE 3.18 – Exemples d'opérations de réduction effectuées sur des données dynamiques (Bach *et al.*, 2014b)

3.2.3.2 Comparaison de graphes

La *correspondance de graphes* (ou *graph matching*) consiste à trouver les correspondances entre les nœuds et les arêtes de deux graphes selon des critères plus ou moins stricts, ce qui permet d'identifier des structures similaires.

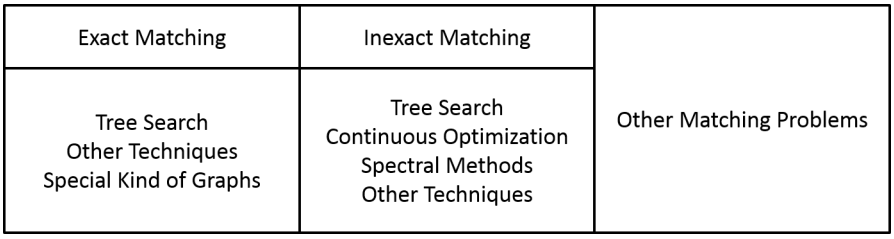


FIGURE 3.19 – Taxonomie des méthodes de correspondance de graphes (Graph Matching), adapté de Conte *et al.* (2004)

Une taxonomie des méthodes de correspondance de graphes a été proposée par Conte *et al.* (2004) dans le domaine de la reconnaissance de motifs (Pattern Recognition), qui est illustrée dans la figure 3.19. Les méthodes sont divisés en deux catégories principales : la recherche de la correspondance stricte ou exacte (Exact Matching), et celle d'une correspondance souple ou inexacte (Inexact Matching). Un isomorphisme entre deux objets est un morphisme – c'est-à-dire un ensemble de règles permettant de passer d'une structure mathématique à une autre – admettant un morphisme inverse. Un isomorphisme entre deux sous-graphes indique donc une correspondance exacte entre eux. La détermination des sous-graphes communs entre deux graphes est devenu le problème le plus fréquent. Un exemple d'application est la comparaison de deux graphes de provenance (Chen *et al.*, 2012) : l'algorithme Direct Classification of node Attendance (DCA) recherche les sous-graphes les plus proches et les nœuds qui ne sont pas inclus dans ces sous-graphes, afin d'indiquer les différences entre deux provenances.

Les contraintes strictes imposées par la correspondance exacte sont trop rigides pour la comparaison de deux graphes, en particulier les grands graphes. En particulier pour les graphes issus de données expérimentales qui sont sujets à des déformations – bruit d'acquisition, variabilité naturelle entre deux objets d'expérience, etc –, les correspondances exactes ne sont pas adaptées et amènent souvent à de mauvaises interprétations (Conte *et al.*, 2004). Il est donc préférable que la correspondance soit souple, en permettant le relâchement des contraintes. Par ailleurs, les coûts computationnels des algorithmes de correspondance de graphes est élevé, en particulier pour la correspondance exacte dont la complexité est exponentielle dans le pire des cas. Avec les algorithmes de correspondance inexacte, il est possible de calculer des solutions approximatives en un temps réduit, avant de décider si un résultat plus précis est souhaité.

3.2.3.3 Techniques de visualisation dynamique

En 2014, un état de l'art très complet a été réalisé dans le cadre de la conférence annuelle EuroVis (Beck *et al.*, 2014). La taxonomie des techniques de visualisation des graphes dynamiques – pour la représentation node-link – proposée par les auteurs dans cet article est donné dans la figure 3.20, avec une indication du nombre de publications par technique. Elle est composée de deux grandes catégories, l'*animation* et la *timeline* (ligne de temps), qui sont présentées dans la suite du document.

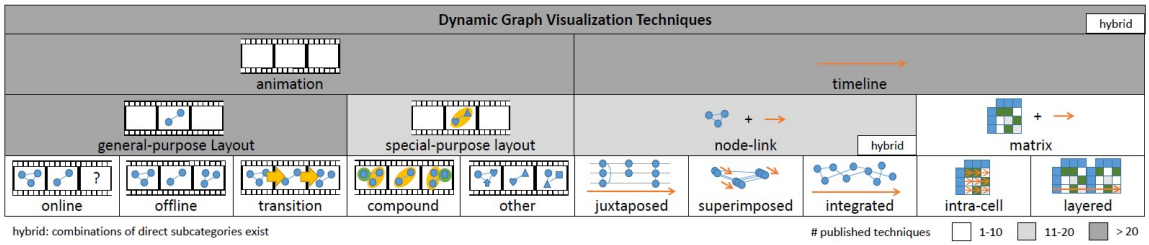


FIGURE 3.20 – Taxonomie hiérarchique illustrée des techniques de visualisation dynamique de graphes. La couleur de fond des cellules du tableau indique le nombre de techniques publiées par catégorie (Beck *et al.*, 2014)

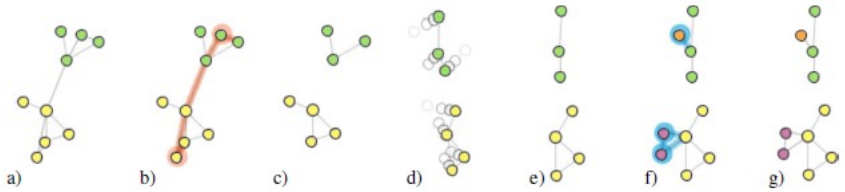


FIGURE 3.21 – Étapes d'une animation pendant laquelle les changements sont mis en relief : a) État initial du graphe, b) Éléments qui vont être supprimés (contour rouge), c) Éléments restants, d) Layout d'adaptation de la position des nœuds aux nouvelles caractéristiques topologiques du graphe, e) Nouvelle position des nœuds, f) Ajout des nouveaux éléments (contour bleu) et g) État final du graphe (Bach *et al.*, 2014a)

Animation Dans une séquence de graphe dynamique animé, les nœuds et les arêtes qui sont ajoutées et enlevées de l'image sont effacées et affichés progressivement. Les trajectoires des mouvements des nœuds sont interpolées afin que les utilisateurs puisse suivre plus facilement les changements du graphe. La figure 3.21 montre un exemple d'animation, étapes par étapes, appliqués à un layout node-link (Bach *et al.*, 2014a).

L'animation peut être appliquée à de nombreux layouts, par exemple un layout radial pour les arbres ou des graphes à la densité faible (les nœuds sont positionnés sur des cercles concentriques), comme illustré dans la figure 3.22(Yee *et al.*, 2001). Tekušová & Schreck (2008) propose d'animer un graphe hiérarchique agencé en trois dimensions, comme présenté dans la figure 3.23.

Timeline Dans les interfaces *small multiples*, les moments temporels sont tous représentés à l'écran de façon individuelle (Tufte, 1991). Les approches small multiples permettent à l'utilisateur une vue globale du temps à travers la séquence complète des moments statiques (Tversky *et al.*, 2002). Pour les diagrammes node-link, Beck *et al.* (2014) identifie trois sous-ensembles de techniques : la juxtaposition de nœuds, la superposition de moments temporels (figure 3.24.a), la timeline intégrée (figure 3.24.a) et les techniques hybrides.

Pohl *et al.* (2008) ont développé le système DGD, un outil de visualisation et d'analyse de réseaux dynamiques qui permet par exemple de visualiser l'évolution du degré (degree centrality) normalisé d'un nœud sur une sélection de nœuds, au cours du temps. Le logiciel StoryFlow permet de visualiser les parcours et les interactions des personnages d'un film avec les événements importants qui s'y rapportent (Liu *et al.*, 2013) ; un exemple est présenté en figure 3.25.

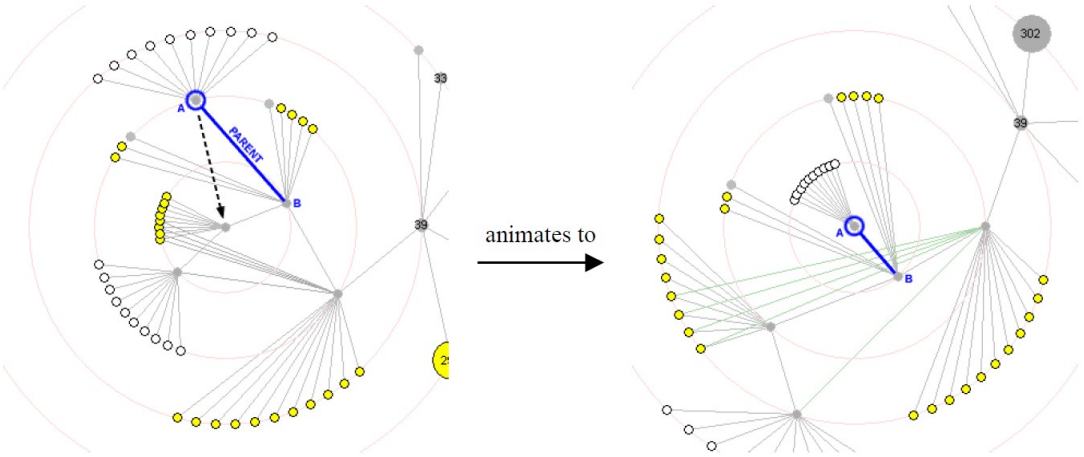


FIGURE 3.22 – Le nœud est sélectionné pour devenir le nouveau focus du layout radial. Les changements sont animés. (Yee *et al.*, 2001)

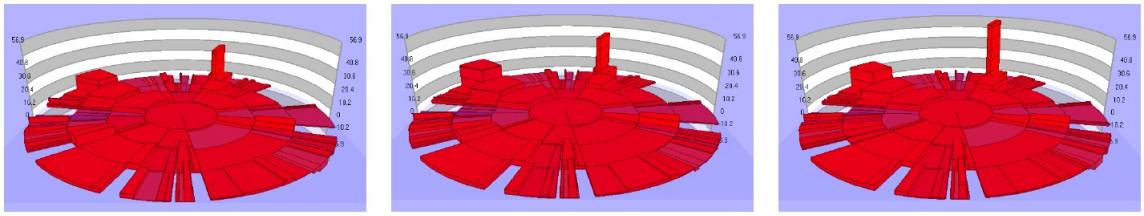
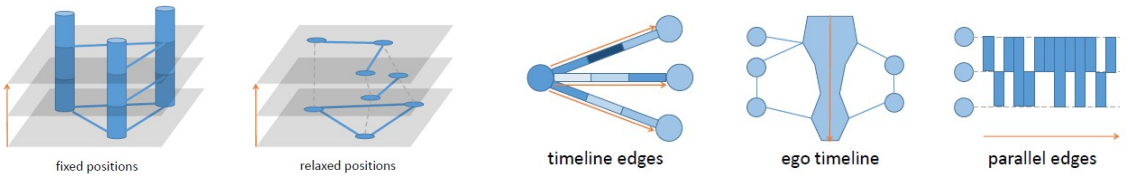


FIGURE 3.23 – Animation temporelle montrant les transitions entre deux moments temporel sur un layout hiérarchique circulaire (Tekušová & Schreck, 2008)



(a) (b)

FIGURE 3.24 – (a) Approche node-link de superposition, chaque couche représentant un moment temporel (b) Approches node-link de timelines (Beck *et al.*, 2014)

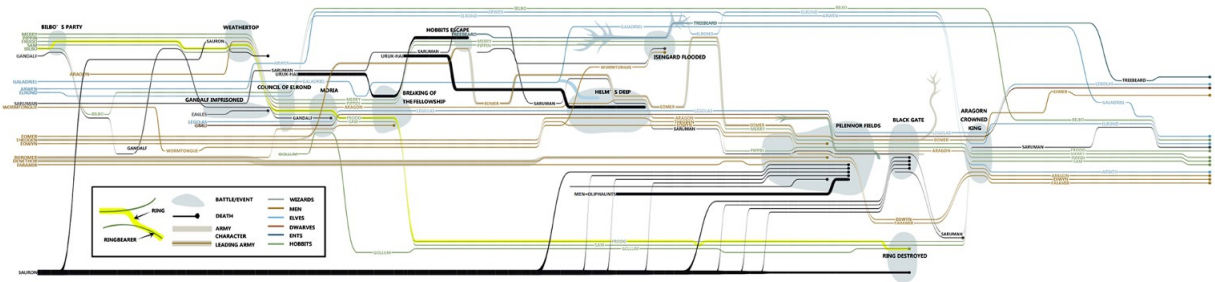


FIGURE 3.25 – Visualisation du film Le Seigneur des Anneaux avec StoryFlow (Liu *et al.*, 2013).

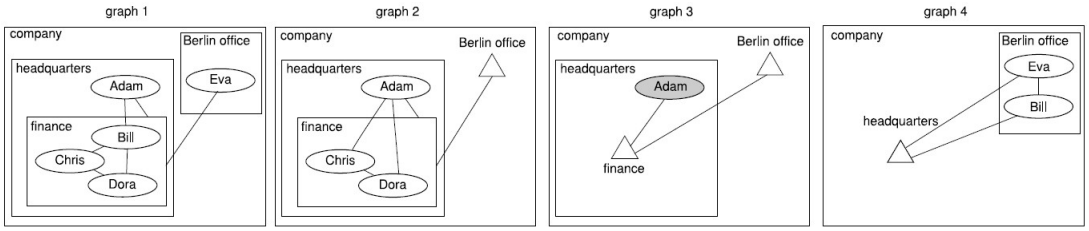


FIGURE 3.26 – Séquence avec application de la méthode de pliage (Reitz *et al.*, 2009)

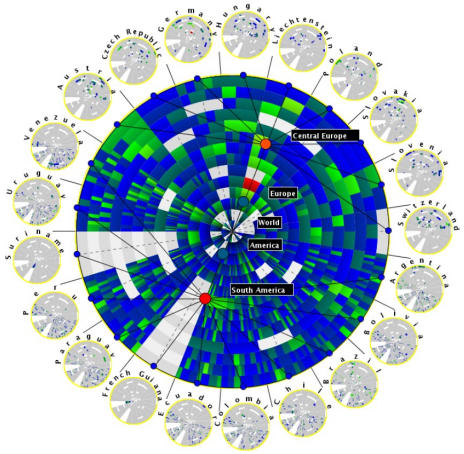


FIGURE 3.27 – Comparaison de matchs de football entre les équipes nationales de l'Europe centrale et de l'Amérique du sud (Burch & Diehl, 2008)

Des techniques particulières ont été développées pour les graphes composés à partir du principe de small multiples. La méthode de pliage de Reitz *et al.* (2009) permet par exemple de plier et déplier des groupes de nœuds en fonction des événements, comme illustré dans la figure 3.26 : si un groupe est identique d'un moment temporel sur l'autre, alors une vue réduite du groupe (symbole) est utilisée pour les moments suivants, et inversement si un changement intervient dans un groupe qui était représenté par une vue réduite. Burch & Diehl (2008) ont développé TimeRadar pour visualiser des graphes composés pondérés : un layout radial présente la hiérarchie et des sections de cercle représentent les changements temporels au niveau des arêtes. Un exemple est présenté dans la figure 3.27.

Cartes des différences Les *cartes des différences* (*difference map*) montrent les différences en termes de nœuds et d'arêtes, entre deux graphes. L'union des nœuds et des arêtes est souvent mis en exergue par rapport aux autres nœuds, grâce à une couleur par exemple. Un exemple de carte des différences est présenté en figure 3.28.

Les cartes de différences sont encore peu utilisées pour visualiser les graphes dynamiques (Archambault *et al.*, 2011b). La combinaison des cartes des différences et des hiérarchies des graphes permettent de montrer les zones de changements sur des grands graphes (Archambault, 2009). Les cartes de différences ont également été utilisées pour accentuer les zones communes entre des réseaux biologiques (Bourqui & Jourdan, 2008). Bach *et al.* (2014a) propose Graph-Diaries qui utilise les cartes de différences pour attirer l'attention de l'utilisateur avant, pendant,

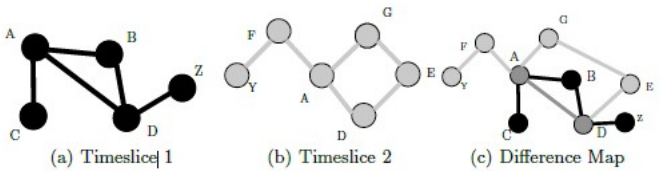


FIGURE 3.28 – Deux moments temporels (a) et (b), et la carte des différences associée (c) (Archambault *et al.*, 2011b)

et après l'animation entre deux moments temporels.

Esthétique des graphes Dynamiques Trois types de critères peuvent être utilisés pour évaluer les qualités d'une visualisation dynamique de graphes et trouver la bonne technique de visualisation pour un jeu de tâches et de données (Beck *et al.*, 2009, 2013) :

- Critères esthétiques généraux : réductions de la pagaille, réduction des confusions spatiales, cohérence de la représentation de motifs similaires, minimisation de l'espace utilisé.
- Critères esthétiques dynamiques : préservation de la carte mentale, réduction de la charge cognitive, minimisation des confusions temporelles.
- Critères esthétiques d'adaptation (scalability) : préservation de la lisibilité quelque soit le nombre de nœuds, d'arêtes, de graphes.

Quelle technique choisir ? Robertson *et al.* (2008) a comparé les techniques de l'animation, du small multiples et des timelines sur des données multidimensionnelles évolutives et a montré que les small multiples donnaient de meilleures performances. Lorsque la carte mentale est préservée, pour Archambault *et al.* (2011a) l'analyse des small multiples est plus rapide que l'animation, sauf quand à la fois les nœuds et les arêtes subissent des changements simultanés, c'est alors l'animation qui présente de meilleures performances. Farrugia & Quigley (2011) a comparé deux séries de graphes dynamiques, et ce sont les small multiples qui semblent les plus adaptés dans la majorité des tâches testées.

Pour Beck *et al.* (2014), il reste difficile de dire quelle technique de visualisation de graphes dynamiques – entre l'animation et les timelines – permet de meilleures performances de la part des utilisateurs. En effet, les résultats publiés jusqu'ici montrent que les performances dépendent avant tout des tâches effectuées, et que chacune des techniques a ses avantages et ses défauts. Des approches hybrides peuvent sous certaines conditions produire de meilleurs résultats que l'une des deux techniques séparément (Rufiange & McGuffin, 2013) ; une illustration du principe des techniques hybride est présenté dans la figure 3.29.

3.2.3.4 Préservation de la carte mentale

La *carte mentale* est l'image mentale que se construit un utilisateur lorsqu'il visualise un graphe. La préservation de la carte mentale est souvent mentionnée dans les techniques de visualisation de graphe, en particulier pour la comparaison de graphes et l'analyse de graphes dynamiques. La définition de la carte mentale assume une idée généralement admise que la

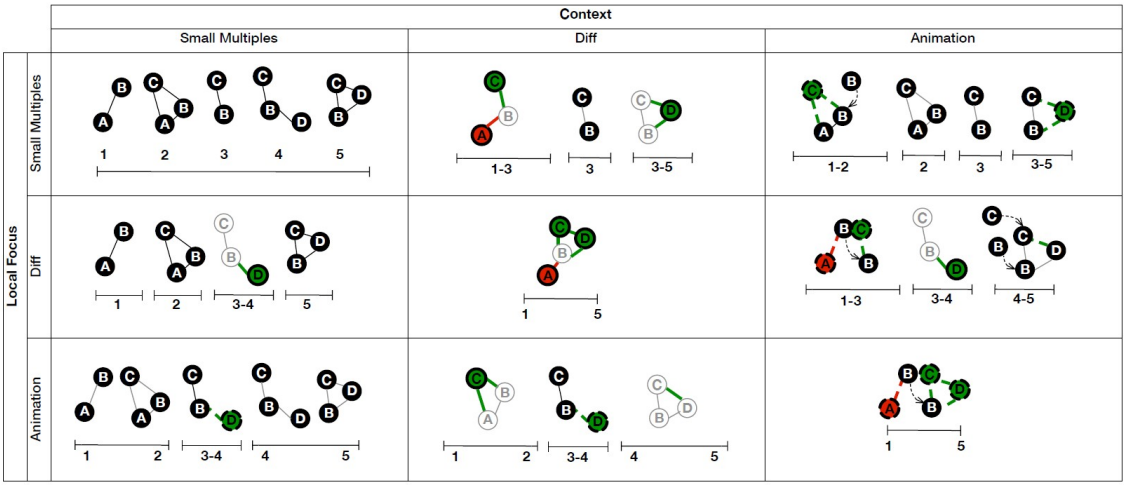


FIGURE 3.29 – Taxonomie des visualisations dynamiques hybrides pour un même graphe, tranches temporelles (Rufiange & McGuffin, 2013)

stabilité globale du dessin permet à une personne d'interpréter un diagramme évolutif sans nécessiter un effort d'efforts cognitifs importants lorsqu'un événement intervient sur le graphe. Le concept de carte mental en visualisation de graphes peut être rapprochée des concepts de *carte cognitive* en psychologie et de *vue générale* en visualisation d'informations.

L'intérêt de la communauté de la visualisation d'information pour la préservation de la carte mentale n'est pas récente (Moen, 1990; Eades *et al.*, 1991). Pour Coleman & Parker (1996), le placement des nœuds et des arêtes devrait changer aussi peu que possible lorsqu'un changement intervient sur le graphe, et ils se sont intéressés aux problèmes liés à la génération automatique de layouts tout en optimisant l'esthétique des graphes. Depuis plusieurs algorithmes ont été proposés pour aider à préserver la carte mentale de l'utilisateur (Diehl & Görg, 2002; Erten *et al.*, 2004; Frishman & Tal, 2008).

Archambault & Purchase (2013b) ont montré que la préservation de la carte mentale pouvait aider l'utilisateur à s'orienter lors de l'exploration de graphes dynamiques. L'ajout de contraintes à un layout favorise la préservation de la carte mentale et pourrait faciliter l'interaction de l'utilisateur quand les nœuds bougent (He & Marriott, 1997).

Cependant, selon Beck *et al.* (2014) le rôle de la carte mentale dans les performances de l'utilisateur pourrait avoir été surestimé. Les limites de la préservation de la carte mentale ont été relevées par Archambault & Purchase (2013a) si les utilisateurs ont l'impression subjectivement que préserver la carte mentale est une aide, il a été montré qu'elle n'aide pas nécessairement à l'interprétation d'un graphe, la détection de changements ou la mémorisation (Archambault & Purchase, 2012; Purchase *et al.*, 2007; Zaman *et al.*, 2011). Sur certaines tâches, la carte mentale peut même se révéler contreproductive pour les utilisateurs Purchase & Samra (2008), Saffrey & Purchase (2008).

Il existe **deux principales représentations de graphe : les diagrammes node-link et les matrices d'adjacence**. Les premiers sont plus adaptés aux graphes clairsemés et permettent de réaliser des tâches complexes, les seconds sont adaptés aux graphes denses et aux tâches simples de visualisation.
Deux grandes familles de layouts sont présentes dans la littérature : les **layouts de force et les layouts basé sur les attributs des nœuds**.
Lors de la conception d'une technique d'agencement des nœuds dans l'espace, **l'esthétique finale est à prendre en compte pour éviter les écueils classiques** qui conduisent à occulter une partie des données à visualiser et à de mauvaises interprétation sur les données.
Les deux catégories principales de techniques pour la visualisation de graphes dynamiques sont l'animation et les small multiples. Le choix de la technique dépend des tâches à effectuer par l'utilisateur.
Le rôle de la carte mentale dans la visualisation de graphes dynamiques est incertain.

3.3 Visualisation interactive

Les techniques de visualisation de graphes, statiques ou dynamiques, présentées dans la section 3.2 ne sont utiles pour l'exploration que dans un contexte où l'utilisateur peut interagir avec les données et les degrés de visualisation.

Dans cette section sont introduits les processus de visualisation identifiés dans la littérature et qui servent à guider le schéma général des actions utilisateurs. Dans une seconde partie, les taxonomies existantes des tâches en visualisation de graphes statiques et dynamiques sont présentées. Pour finir, les caractéristiques des principaux logiciels de visualisation de graphes sont discutées.

3.3.1 Processus de visualisation

Quand le graphe entier est trop complexe ou trop large pour être affiché dans une seule vue statique, l'utilisation de techniques d'interaction devient nécessaire. Par ailleurs, l'utilisateur peut demander à appliquer de nouveaux algorithmes ou des mises à jour des vues pour tester des hypothèses : l'exploration est profondément reliée à l'interaction. Ahlberg *et al.* (1992) décrit les interactions et plus spécifiquement les requêtes dynamiques comme nécessaires pour réellement accomplir une exploration. La raison principale est cognitive : l'exploration requière que plusieurs hypothèses soient maintenues dans la mémoire à court-terme qui est très limitée en capacité. Planifier des opérations complexes sans feedback ou sans utiliser une syntaxe textuelle consomme toute la mémoire à court-terme et l'exploration devient impossible avec la mémoire à court-terme seulement. Par conséquent, fournir des interactions avec un feedback immédiat pour les opérations les plus courantes d'exploration.

La catégorisation des techniques d'interaction peuvent être basée sur des critères variés tels

que les tâches, les intentions de l'utilisateur (Yi *et al.*, 2007) ou les actions de l'utilisateur (Elmqvist & Fekete, 2010). Ces trois critères sont liés. Par exemple, une tâche inclut l'exécution de plusieurs actions ou une tâche peut correspondre à plusieurs intentions de la part de l'utilisateur. De plus, une intention de l'utilisateur peut être réalisée par plusieurs actions, et une action peut correspondre à plusieurs intentions. Von Landesberger *et al.* (2011) répartit les techniques d'interaction en trois catégories :

- L'action utilisateur affecte-elle les données ? (la sélection des données affichées ou les valeurs des données)
- L'affichage visuel des données ? (les paramètres des données ou la représentation visuelle)
- La vue des données.

Les données, la représentation visuelle et la manipulation des vues peuvent être utilisés pour une exploration interactive.

Il existe deux familles de techniques d'interaction avec la ou les vues d'un graphe : les *déplacements* (panoramique, zoom), et les *lentilles* (magic lenses) qui permettent de focaliser l'attention une zone du graphe (fisheye) ou sur plusieurs zones (multiple foci). Un exemple d'utilisation du fisheye est donnée dans la figure 3.30 : seules les arêtes d'intérêt sont affichées dans l'espace visuel de la lentille.

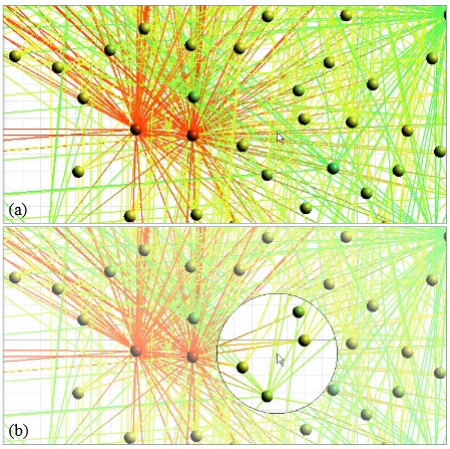


FIGURE 3.30 – (a) Quand le layout du graphe est encombré, il est difficile d'identifier les connexions entre des nœuds particuliers. (b) L'utilisation d'une lentille locale interactive permet de ne garder au niveau local que les arêtes d'intérêt (Tominski *et al.*, 2006)

L'utilisateur peut également interagir avec le graphe à travers les niveaux d'abstraction de la visualisation (Von Landesberger *et al.*, 2011) :

- Paramètres visuels : highlighting, brushing and linking, semantic zooming.
- Schéma visuel : changement du layout, changement de la représentation visuelle.
- Filtrage des données :
  - Approche top down : la visualisation commence avec le graphe entier.
  - Approche bottom up : la visualisation commence à partir d'un seul nœud sélectionné par l'utilisateur, la navigation se fait sur la structure du graphe ou la fonction de degré d'intérêt.

- Approche middle-out : approche hybride, qui combine les approches top down et bottom up.
- Modification des données : édition et agrégation du graphe.

Des techniques récentes qui ont été développées pour aider à l'exploration de graphes variés s'appuie sur l'utilisation des propriétés structurelles du graphe pour naviguer de façon interactive dans le graphe. L'outil de visualisation de graphes dynamiques CGV (Coordinated Graph Visualization) propose plusieurs vues interactives à l'utilisateur dont certaines sont calculées à partir des propriétés topologiques (Tominski *et al.*, 2009). Van Ham & Perer (2009) utilise la mesure du degré d'intérêt (degree-of-interest) pour identifier les sous-graphes les plus pertinents à afficher à partir d'un nœud choisi (approche de navigation bottom up).

3.3.2 Tâches en visualisation

Selon (Bertin, 1973), un designer devrait avoir en tête une tâche utilisateur spécifique lorsqu'il conçoit une nouvelle technique de visualisation. Depuis une dizaine d'année, le domaine de la visualisation d'information s'est doté de taxonomies des tâches auxquelles se référer.

Tâches de bas et de haut niveau Dans la bibliographie, les tâches en visualisation peuvent être séparées en deux niveaux (Amar *et al.*, 2005) :

- Les tâches de *bas-niveau* (*low-level*), qui sont de simples questions appelant un résultat fini. Leur portée est limitée aux éléments du réseau et leurs attributs.
- Les tâches de *haut-niveau* (*high-level*), qui requièrent un jugement humain pour comprendre le résultat : les utilisateurs utilisent leurs connaissances du domaine (Bach, 2014).

Une tâche composée est une tâche constituée de plusieurs tâches bas-niveau qui doivent être réalisées de façon séquentielle, la première tâche servant d'entrée dans la tâche suivante (Bach, 2014). Par exemple *Trouver le plus grand groupe de nœuds classifiés selon la propriété x* combine les tâches *Regrouper les nœuds selon la propriété x* et *Trouver le plus grand groupe parmi eux.*

Tâches en visualisation statique Plusieurs taxonomies des tâches pour la visualisation d'information ont été conçues les vingt dernières années, centrées principalement sur les graphes statiques. Les tâches primitives appliquées aux objets du graphe proposées par (Amar *et al.*, 2005) ont été complétées pour couvrir toutes les tâches bas-niveau (Lee *et al.*, 2006) , qui sont : récupérer une valeur, filtrer, calculer une valeur dérivée, trouver un extrémum, ordonner, varier, calculer la distribution, trouver des anomalies, trouver les clusters, corréler, parcourir et modifier et une valeur. Par ailleurs, quatre groupes de tâches haut-niveau ont été proposés par Lee *et al.* (2006) dans leur taxonomie : basé sur la topologie (adjacence, accessibilité, connexion commune, connectivité), basé sur les attributs (des nœuds et des arêtes), la navigation (suivre un chemin, réexamen) et les tâches de vue d'ensemble.

Brehmer & Munzner (2013) ont récemment présenté une étude des travaux précédents en taxonomies pour la visualisation, en mettant en exergue les points positifs et négatifs. Leur conclusion est que la plupart des taxonomies ne décrivent les tâches que partiellement, mais ne répondent pas nécessairement aux questions qu'un concepteur devrait se poser :

– POURQUOI une tâche est effectuée
– COMMENT une tâche est effectuée
– QUELLES sont les entrées et sorties de la tâche

Tâches en visualisation dynamique Shneiderman (1996) proposa en 1996 une taxonomie des types de données à croiser avec les tâches de recherche d'information. Les sept types de données sont : données 1D, 2D ou 3D, données temporelles et multi-dimensionnelles, arbres et réseaux. Les sept tâches de recherche d'information présentent un haut degré d'abstraction, sans vraiment s'intéresser aux spécificités des graphes dynamiques : vue générale, zoom, filtre, détails à la demande, relation, historique et extraction. Cependant, à travers les tâches d'historisation, les auteurs mettent en exergue que la visualisation est un processus exploratoire avec de nombreuses étapes, qui doivent être prises en compte lors de la conception d'une technique de visualisation pour des graphes complexes.

Six transformations possibles dans un graphe évolutif ont été identifiées (Palla *et al.*, 2007) : expansion, contraction, fusion, rupture, naissance et mort. Les quatre premières transformations s'appliquent au niveau du graphe, tandis que les deux derniers s'appliquent aux éléments du graphe uniquement. La plupart du temps, il y a un manque de niveaux de détails (granularité) dans le processus d'analyse visuelle (Ahn *et al.*, 2014) : des niveaux de sélection de l'analyse ne sont pas disponibles à l'utilisateur. La taxonomie des tâches des auteurs pour la visualisation de graphes dynamiques présente trois dimensions : les entités du graphe, les propriétés du graphe à visualiser et la hiérarchie des fonctions temporelles. Ils distinguent, de la même façon que Lee *et al.* (2006), les propriétés structurelles des attributs de domaine, et les appelle "niveaux du graphe".

Yi *et al.* (2010) a identifié trois niveaux d'analyse : (1) les changements temporels à un niveau global, (2) les changements temporels au niveau d'un groupe et (3) les associations temporelles selon les attributs des noeuds et des dyades. Pour Hadlak *et al.* (2011), le processus de visualisation peut être divisé en trois catégories, selon que le graphe est montré dans son intégralité (non-réduit) ou réduit par une sélection ou une abstraction.

Bach (2014) a proposé un travail intéressant, en fusionnant les taxonomies précédentes et se préoccupant en particulier des graphes dynamiques, qui avaient été négligés jusqu'à présent. La taxonomie qu'il présente est composée de trois dimensions : où (éléments statiques du graphe), quand (dimension du temps) et quoi (événements à court et long terme).

Tâches d'analyse spécifiques à un domaine Des tâches portant sur l'analyse et la comparaison de graphes pondérés pour l'analyse de la connectivité cérébrale (représentation non-spatiale) ont été identifiées (Alper *et al.*, 2013). Cependant, ces tâches ne portent que sur la comparaison entre deux graphes et non entre un ensemble de graphes considérés comme une continuité.

Peu de travaux traitent des techniques de visualisation pour la navigation et l'analyse de données complexes dans le domaine du PLM, et aucune ne propose des tâches de visualisation qu'il faudrait prendre en compte.

3.3.3 Logiciels de visualisation de graphes

Il existe de nombreuses librairies de visualisation de graphe, dans la plupart des langages de programmation – NetworkX ou Snap en python, iGraph en R, JUNG en Java, Sigma.js en Javascript, etc. En revanche le nombre de logiciels proposant une interface d'exploration interactive est beaucoup plus restreint. Nous proposons une liste non exhaustive, basée sur la popularité :

Pajek (1998) : Pajek [3] est le plus ancien des logiciels d'exploration de graphe et reste aujourd'hui encore pertinent, notamment grâce à sa version "XXL", spécialement conçue pour les gros graphes. Pajek est très orienté algorithmes et n'offre que peu d'options pour la visualisation. Pajek est un logiciel gratuit mais non open-source. (Batagelj & Mrvar, 1998)

UCINet (2002) : UCINet [4] peut être vu comme une version payante de Pajek partageant les même qualités et les mêmes défauts : une bibliothèque d'algorithme d'analyse de réseaux sociaux assez fournie mais des capacités de visualisation assez limitées. (Borgatti *et al.*, 2002)

Cytospace (2003) : Créé originellement pour la recherche en biologie (notamment pour pour représenter les interactions moléculaires), Cytoscape [5] est maintenant devenu un logiciel générique de visualisation de graphe. Cytoscape est open-source et propose un système de plugin Java. (Smoot *et al.*, 2011)

Tulip (2004) : Tulip [6] est un logiciel open-source proposant à la fois de nombreux algorithmes d'analyse de réseaux sociaux et des fonctionnalités de visualisation variées – représentation matricielle et heatmap. Tulip propose un mécanisme d'intégration d'algorithmes par plugin, en C++. (Auber, 2004)

NodeXL (2008) : NodeXL [7] est une extension du logiciel Excel, ce qui rend son usage trivial et permet un pré-traitement des données simple et flexible dans l'interface d'Excel. (Smith *et al.*, 2009)

Gephi (2009) : Gephi [8] est rapidement devenu un logiciel très populaire, notamment grâce à la qualité esthétique des visualisations qu'il génère et à sa facilité d'utilisation. Il permet la visualisation dynamique, pour les éléments du graphe comme pour leurs attributs. A l'inverse de Pajek ou UCINet, les algorithmes et métriques intégrées au logiciel restent restreints. Gephi fait cependant l'objet de contributions régulière via son système de plugin en Java. (Bastian *et al.*, 2009)

Commetrix [9] est un environnement logiciel moins populaire mais développé pour la visualisation dynamique de réseaux sociaux, et qui propose également des fonctionnalités de data

3. http://mrvar.fdv.uni-lj.si/pajek/
4. https://sites.google.com/site/ucinetsoftware/home
5. http://www.cytoscape.org/
6. http://tulip.labri.fr/TulipDrupal/
7. http://nodexl.codeplex.com/
8. http://gephi.github.io/
9. http://www.commetrix.de/

mining. Son interface interactive a été conçue à partir des principes de la théorie Gestalt (Trier, 2006).

L'interaction est nécessaire à l'exploration de données complexes. L'utilisateur peut interagir avec la visualisation selon trois axes : **les données elles-mêmes, la représentation des données et la vue des données**. Il existe deux familles de techniques d'interaction : les déplacements et les lentilles.
Les taxonomies de tâches en visualisation servent à faciliter le travail des concepteurs de techniques de visualisation. Il existe dans la bibliographie des taxonomies pour les tâches de visualisation statique et dynamique.
Plusieurs logiciels de visualisation de graphes sont développés par la communauté de la visualisation d'information. Peu d'entre eux proposent des fonctionnalités avancées pour l'exploration de graphes dynamiques.

3.4 Exploration de la connectivité fonctionnelle cérébrale

Bullmore & Sporns (2009) définissent la connectivité fonctionnelle comme la dépendance ou l'association statistique entre les éléments d'un *réseau*. Dans le cas de l'étude du cerveau, celui-ci est segmenté en régions dont les associations sont mesurées : les régions sont les *nœuds* et les associations sont les *arêtes* du *graphe* qui représente le réseau cérébral.

Depuis la naissance de la science des réseaux au milieu des années 90, des études interdisciplinaires ont été menées pour caractériser la structure et la fonction des réseaux. Les systèmes structurels et fonctionnels du cerveau présentent les caractéristiques des réseaux complexes, tels que la topologie du petit monde (small-world), une hiérarchie de sous-réseaux, des hubs extrêmement connectés et une forte modularité [10] à l'échelle du cerveau entier chez les humains.

3.4.1 Obtention des réseaux de connectivité fonctionnelle

Les étapes d'exploration des réseaux de connectivité fonctionnelle cérébrale, depuis leur calcul jusqu'à leur analyse, ont été présentées par Bullmore & Sporns (2009). Nous décrivons les quatre étapes dans le cas d'une étude à partir d'acquisitions IRM, qui sont illustrées dans la figure 3.31 :

1. Segmentation du cerveau en zones, ce qui définit les nœuds du graphe. Dans le cas de l'étude de la connectivité fonctionnelle obtenue avec des technique d'imagerie IRM, les zones sont définies anatomiquement ou fonctionnellement. Une segmentation particulière du cerveau est appelé *atlas*.
2. Mesure de la connectivité (aussi appelée mesure de la corrélation entre les decours temporels des régions) : Estimer une mesure continue de l'association entre les nœuds. Il peut s'agir du spectre de la cohérence ou de la causalité de Granger entre deux capteurs magnétoencéphalographiques, ou encore la connexion probable entre deux régions d'un jeu

10. Les concepts de la théorie des graphes sont définis dans la section 3.1 du chapitre 3.

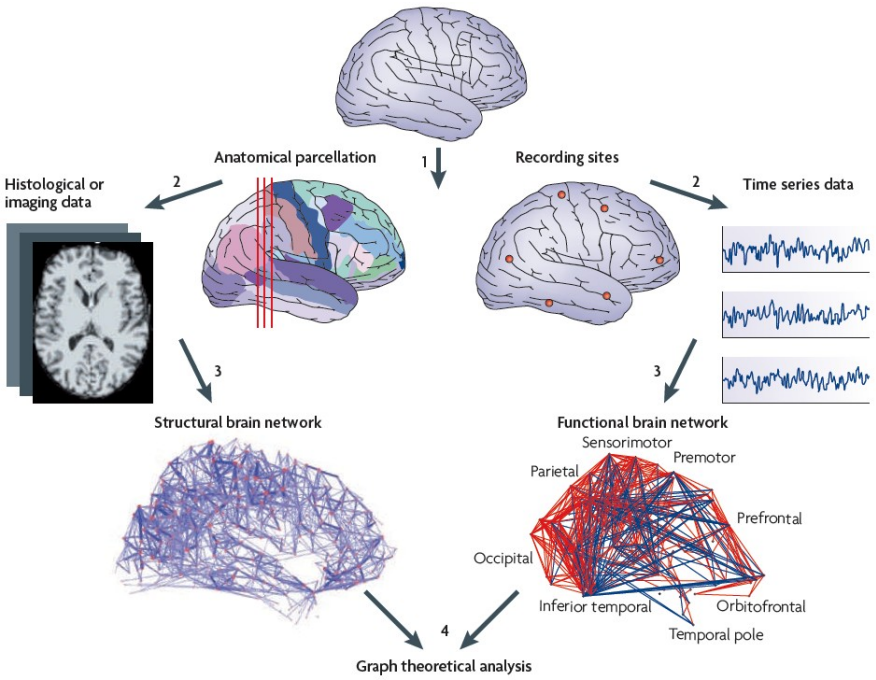


FIGURE 3.31 – Exploration de la connectivité cérébrale : 1) Segmentation du cerveau en zones, 2) Mesure de la connectivité, 3) Calcul de la matrice d'adjacence de connectivité pour un certain seuil, 4) Calcul et analyse des propriétés du réseau. (Bullmore & Sporns, 2009)

de données DTI, ou la corrélation inter-régions de l'épaisseur corticale ou l'estimation du volume IRM estimé sur un groupe de sujets.

3. Calcul de la matrice d'adjacence de connectivité pour un certain seuil : générer la matrice associée en compilant toutes les associations de paires de régions entre les nœuds et (de façon générale) appliquer un seuil sur chaque élément de cette matrice pour produire une matrice d'adjacence binaire qui n'est autre qu'un graphe non-dirigé.
4. Calcul et analyse des propriétés du réseau : calculer les paramètres d'intérêt du graphe (propriétés topologiques), comparer le réseau avec des réseaux calculés sur des populations différentes ou un graphe aléatoire...

La provenance associée à chaque étape doit être gérée avec les données pour s'assurer par exemple que deux réseaux calculés à des moments différents peuvent être comparés.

Le cerveau doit dynamiquement intégrer, coordonner et répondre aux stimuli internes et externes à travers plusieurs échelles de temps. La connectivité fonctionnelle est explorée dynamiquement, même si cette démarche est plus récente (Hutchison *et al.*, 2013). Il existe deux familles de stratégies d'analyse des variations temporelles : (1) capturer les variations entre paires de nœuds de la synchronisation inter-régionale et (2) identifier les changements de motifs de synchronisation à un niveau multivarié. La méthode la plus couramment utilisée appartient à la famille (1) et est appelée stratégie de la fenêtre glissante : une fenêtre temporelle de longueur fixe et la connectivité est calculée à chaque point dans le temps à l'intérieur de la fenêtre. Le différence entre l'intervalle entre deux points temporels et la taille de la fenêtre définit la quantité de chevauchement de deux fenêtres successives. Selon (Hutchison *et al.*, 2013), les résultats

obtenus par la méthode de la fenêtre glissante indiquent qu'elle serait adaptée pour mettre en évidence des phénomènes fonctionnels, même s'il reste encore à déterminer les paramètres les plus appropriés pour que l'approche soit entièrement validée.

3.4.2 Techniques d'exploration visuelle

Deux publications intéressantes introduisent aux techniques de modélisation par graphe du connectome humain en général (Bullmore & Bassett, 2011) et de la connectivité cérébrale en particulier (He & Evans, 2010).

Les réseaux de connectivité fonctionnelle sont souvent représentés sous la forme de graphes node-link [11] ou de représentations matricielles (voir figure 3.32). Les techniques node-link avec layout physique montrent des limites pour les réseaux denses avec beaucoup de nœuds, et les représentation sous forme matricielle sont plus efficaces pour l'analyse de la pondération des relations de connectivité entre les régions du cerveau (Alper *et al.*, 2013). Cependant, les graphes node-link permettent de présenter les données avec un layout en trois dimensions qui préserve l'anatomie du cerveau (le réseau fonctionnel cérébral de la figure 3.31 en est un exemple). Les projections anatomiques en deux dimensions sont également utilisées, car si l'usage de la visualisation en trois dimensions permet de côté réduire le croisement des arêtes, elle favorise l'occlusion visuelle de certaines éléments du graphe (nœuds comme arêtes), ce qui peut mener à de fausses interprétations.

L'analyse des réseaux de connectivité fonctionnelle est souvent abordée par un clustering hiérarchique qui met en avant les groupes de régions présentant une forte densité de connexion. Les groupes peuvent être visualisés facilement sur une représentation matricielle ou via un dendrogramme, comme illustré sur la figure 3.32.

Les techniques de visualisation de graphe les plus appropriées dans le cas d'une analyse dynamique de la connectivité fonctionnelle ne sont pas encore connues (Hutchison *et al.*, 2013). L'affichage de small-multiples (une vignette par état temporel), la réduction du graphe à ses clusters, le regroupement des arêtes en faisceau ou la création des glyphes basés sur la projection 2D des régions cérébrales sont les techniques actuellement les plus utilisées.

3.4.2.1 Les outils de visualisation pour l'analyse des réseaux biologiques

Un état de l'art des outils de visualisation pour l'analyse des réseau biologiques a été proposé par Pavlopoulos *et al.* (2008), où sont discutées les limites des outils actuels et la direction que devraient prendre les futurs développements. La quantité et l'hétérogénéité des données à visualiser posent des problèmes techniques – incapacité des outils à prendre en charge des réseaux trop volumineux – et fonctionnels – ergonomie visuelle non adaptée, les réseaux denses présentent des chevauchements de nœuds et d'arêtes. Pavlopoulos *et al.* (2008) suggère que les futurs outils devraient chercher à améliorer les performances, à intégrer les algorithmes d'analyse, à prendre en compte une troisième dimension spatiale dans la conception de nouveaux

11. Les nœuds sont représentés par des points et les arêtes par des traits (voir la section 3.2 du chapitre 3).

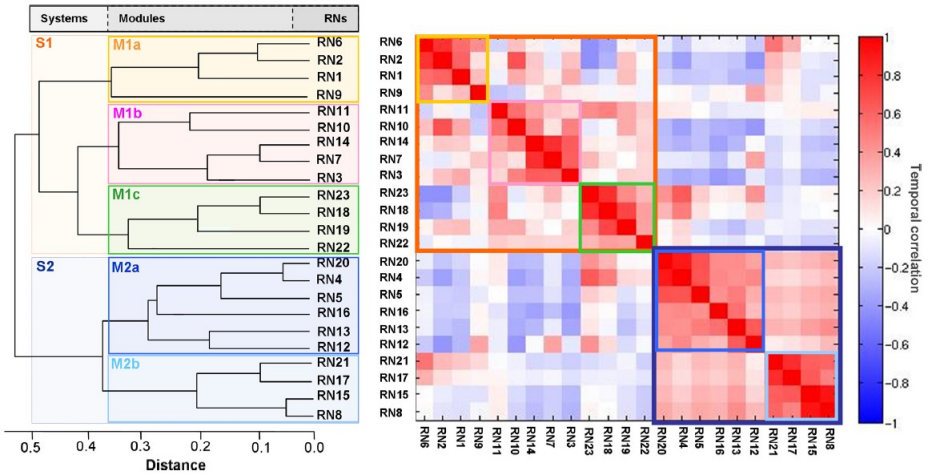


FIGURE 3.32 – Ananlyse par clustering hiérarchique des corrélations temporelles de 23 réseaux à l'état de repos (RN). Gauche : dendrogramme de l'analyse partitionnées en 2 systèmes (S1, S2) et 5 modules (M1a, M1b, M1c, M2a, M2b). Droite : matrice d'adjacence des corrélations temporelles des 23 RN, avec pondération. (Doucet *et al.*, 2011)

layouts. Cette dimension pourrait servir à afficher un paramètre ou des données temporelles, ce qui permettrait de mieux analyser la complexité des réseaux biologiques.

Parmi les outils de visualisation présentés dans l'article, Cytoscape est développé de façon active et soutenu par une large communauté d'utilisateurs, essentiellement autour de la visualisation de réseaux de molécules (Smoot *et al.*, 2011). Il existe également deux outils de visualisation générique de graphes, qui sont parfois utilisés pour la visualisation de réseaux biologiques : Tulip (Auber, 2004) et Gephi (Bastian *et al.*, 2009).

Le domaine de la neuroimagerie a développé des outils spécifiques, comme le Brain Net Viewer [12] qui permet notamment la visualisation de la connectivité fonctionnelle sous forme de graphes (Xia *et al.*, 2013).

Certaines bases de données intègrent des outils de visualisation de graphes, comme l'UMCD qui permet de calculer et de comparer des réseaux de connectivité fonctionnelle (Brown *et al.*, 2012). Le Connectome Viewer Toolkit (CVT) gère les chaînes de traitement d'images et la visualisation des données à chaque étape, ce qui inclut l'exploration de la connectivité fonctionnelle (Gerhard *et al.*, 2011).

12. http://www.yonghelab.org/downloads/brainnet-viewer

L'exploration de la connectivité fonctionnelle cérébrale se fait à l'aide de la visualisation de graphes (appelés réseaux en biologie). La plupart des techniques de visualisation utilisées sont soit les matrices d'adjacence, soit les graphes node-link en 3 dimensions avec conservation de la position réelle des régions dans l'espace.
L'exploration de la connectivité fonctionnelle cérébrale semble être le cas d'application idéal de cette thèse, puisque les chercheurs en neuroimagerie l'étudie à l'aide de la théorie des graphes et de techniques du domaine de la visualisation d'informations.

Conclusion du chapitre 3

Dans ce chapitre nous avons introduit les concepts de la théorie des graphes ainsi que les techniques de visualisation de graphes statiques et dynamiques. Il existe deux principales représentations de graphe : les diagrammes node-link et les matrices d'adjacence. Les premiers sont plus adaptés aux graphes clairsemés et permettent de réaliser des tâches complexes, les seconds sont adaptés aux graphes denses et aux tâches simples de visualisation. Dans leur état de l'art sur les techniques de visualisation de graphe, Von Landesberger *et al.* (2011) propose une classification, selon leur structure et leur dépendance au temps, qui est présentée dans la figure 3.33.

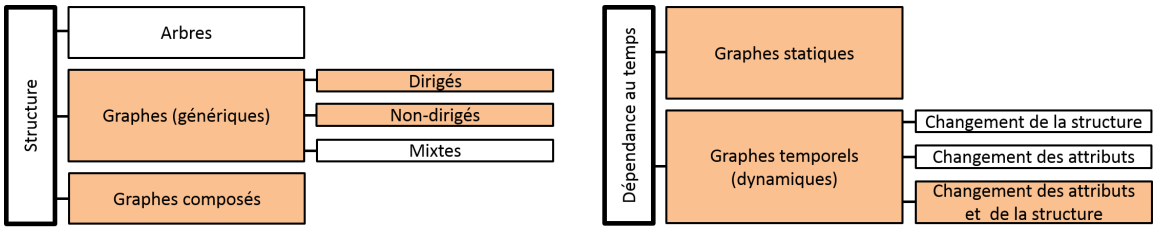


FIGURE 3.33 – Classification des graphes, adapté de (Von Landesberger *et al.*, 2011). En orange : les types de graphes intéressants pour représenter des jeux de données hétérogènes et multidimensionnels.

Deux grandes familles de layouts sont présentes dans la littérature : les layouts de force et les layouts basé sur les attributs des nœuds. Les deux catégories principales de techniques pour la visualisation de graphes dynamiques sont l'animation et les small multiples. Le choix de la technique dépend des tâches à effectuer par l'utilisateur. Le rôle de la carte mentale dans la visualisation de graphes dynamiques est incertain, même si plusieurs études montrent que sa préservation d'un moment temporel sur l'autre pourrait aider à l'exploration de graphes dynamiques.

Lors de la conception d'une technique d'agencement des nœuds dans l'espace, l'esthétique finale est à prendre en compte pour éviter les écueils classiques qui conduisent à occulter une partie des données à visualiser et à de mauvaises interprétation sur les données. L'interaction est nécessaire à l'exploration de données complexes. L'utilisateur peut interagir avec la visua-

lisation selon trois axes : les données elles-mêmes, la représentation des données et la vue des données. Les deux familles de techniques d'interaction ont été présentées : les déplacements et les lentilles. Les taxonomies pour les tâches de visualisation statique et dynamique existant dans la bibliographie ont été introduites. Elles servent à faciliter le travail des concepteurs de techniques de visualisation. Il existe plusieurs logiciels de visualisation de graphes qui sont développés par la communauté de la visualisation d'information, mais peu d'entre eux proposent des fonctionnalités avancées pour l'exploration de graphes dynamiques.

Pour finir, nous avons présenté l'exploration de la connectivité fonctionnelle cérébrale en neuroimagerie. Elle se fait à l'aide de la visualisation de graphes (appelés réseaux en biologie). La plupart des techniques de visualisation utilisées sont soit les matrices d'adjacence, soit les graphes node-link en 3 dimensions avec conservation de la position réelle des régions dans l'espace. Les graphes de connectivité fonctionnelle sont complets et ont un nombre de nœuds fixes, lié à l'atlas utilisé. Les arêtes sont souvent filtrées lorsqu'une représentation node-link est utilisée, afin de focaliser l'attention sur les connexions les plus fortes. **L'exploration de la connectivité fonctionnelle cérébrale semble être le cas d'application idéal de cette thèse**, puisque les chercheurs en neuroimagerie l'étudie à l'aide de la théorie des graphes et de techniques du domaine de la visualisation d'informations.

Afin d'orienter notre réponse au problème 2 "comment visualiser les structures de données multidimensionnelles et dynamiques ?" présenté dans le chapitre 1, nous avons identifié que les données à visualiser sont hétérogènes, multidimensionnelles et évolutives. Nous nous intéressons donc aux graphes multivariés, composés et dynamiques (attributs et structure), comme illustré dans la figure 3.33. La dimension temporelle des graphes dynamiques n'est pas suffisante pour représenter

Ce qui nous amène à formuler **deux objectifs** pour résoudre le problème 2 :

1. **Quelle représentation** pour les données hétérogènes et complexes (multidimensionnelles, hiérarchiques, évolutives) ?
2. **Comment explorer** et donc interagir avec ces données ?

Il n'existe pas actuellement de représentation sous forme de graphe des données que nous cherchons à explorer. Quelle forme conceptuelle pourrait prendre ce graphe ? Quelle structure de graphe créer qui s'appuierait sur les structures existantes (arbres, graphes génériques statiques et dynamiques, graphes composés...) ?

Pour explorer visuellement la nouvelle structure de graphe, nous pouvons nous appuyer sur les représentations visuelles dynamiques et les techniques d'interaction existantes. A l'étude de l'état de l'art, les diagrammes node-link et les techniques small multiples semblent de bonnes pistes. Cependant, nous pouvons identifier **trois verrous à lever** pour permettre l'exploration de données multidimensionnelles et dynamiques :

– **Stabilité du layout** – Comment préserver la carte mentale tout en mettant en exergue les changements ?

– **Stabilité des données** – Comment mettre en évidence des structures constantes et des structures variables ?
– **Interaction** – Quel processus d'interaction pour aborder les données ?

Nous avons vu dans l'introduction de ce chapitre qu'un problème de visualisation est caractérisé par trois composantes Aigner *et al.* (2011) : ce qui est présenté (*Quoi ?*), ce que l'utilisateur cherche à voir (*Pourquoi ?*) et la technique utilisée (*Comment ?*). Dans la suite de la thèse, le chapitre 5 répondra à l'objectif 1 (questions *Quoi ?* et *Pourquoi ?*), et le chapitre 6 à l'objectif 2 (question *Comment ?*).

Chapitre 4

Modélisation et structuration des données en neuroimagerie

Ce chapitre traite de l'objectif de recherche "faciliter la conservation de la provenance et structurer les données données hétérogènes" de notre thèse (voir figure 4.1). Nous avons vu dans la section 1.3.1.2 que les domaines de l'industrie manufacturière et de l'imagerie médicale présentent des caractéristiques similaires : la pluridisciplinarité, le partage entre différents acteurs, la réutilisation des données et le besoin de flexibilité du modèle de données. Notre hypothèse est que les approches PLM peuvent être utilisées pour mettre en œuvre une gestion efficace et pérenne des données d'imagerie médicale. Les modèles de données PLM traditionnels ont été conçus pour l'industrie manufacturière et ses spécificités, et ces modèles ont évolué pour permettre l'application du PLM à de nouveaux domaines. Dans ce chapitre nous présentons notre démarche pour définir un modèle et une structuration des données d'imagerie médicale pour la neuroimagerie, puis une description complète du modèle ainsi obtenu.

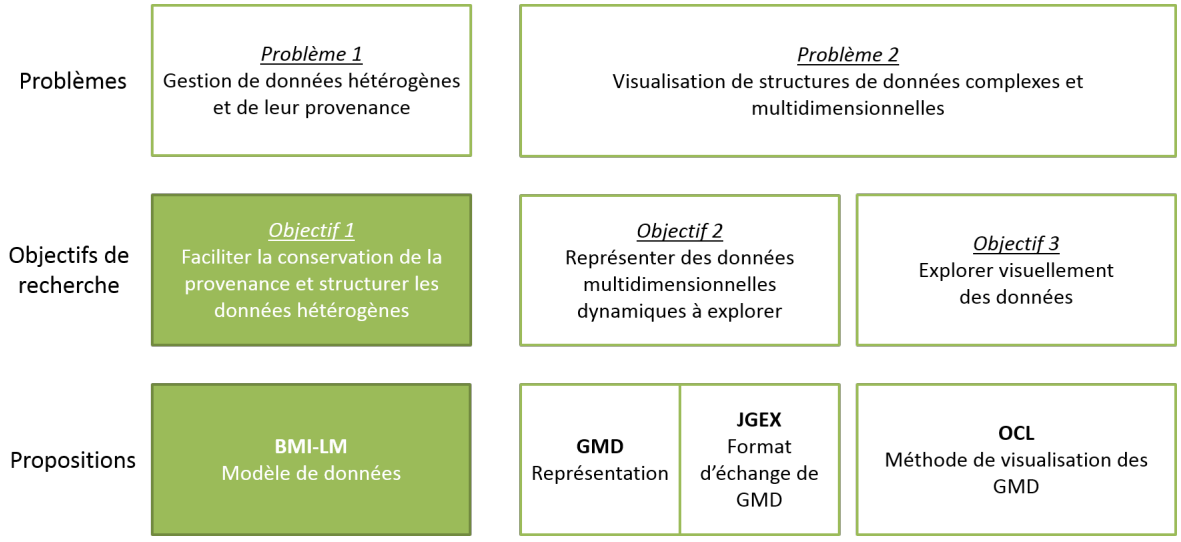


FIGURE 4.1 – Axe 1 : faciliter la conservation de la provenance et structurer les données hétérogènes.

Sommaire

4.1 Méthode

Afin de définir un modèle générique qui prenne en compte la complexité des études en neuroimagerie, nous appliquons la démarche suivante : 1. Clarification des besoins, 2. Définition du modèle de données, 3. Implémentation et retours utilisateurs.

4.1.1 Clarification des besoins

Dans notre démarche, nous étudions les besoins selon deux axes :

1. Étude de l'existant du domaine à partir de moteurs de recherche de données bibliographiques comme PubMed [1]. Notre analyse couvre les trente dernières années, qui correspondent à l'émergence de bases de données à grande échelle dans le domaine, et s'articule en deux points :
   - Compréhension des processus de recherche en neuroimagerie et analyse de la structuration des données et des modèles de référence. Les modèles répertoriés sont ensuite comparés et analysés pour en déterminer les limites éventuelles.
   - Analyse des solutions existantes de gestion des données en neuroimagerie. Des critères d'analyse sont établis pour pouvoir comparer et mettre en évidence les points forts et ceux à améliorer.
2. Interviews de chercheurs d'un groupe de recherche en neuroimagerie pour identifier leurs attentes vis à vis d'un système de gestion des données. Ces interviews se sont déroulées sur la journée du 23 octobre 2012 dans le laboratoire du Groupe d'Imagerie Neurofonctionnelle (GIN), à l'Université de Bordeaux.
   - Une présentation de notre démarche a été faite en commun à tous les participants, puis nous avons interviewés ceux-ci par petits groupes de spécialités (deux à trois personnes).
   - L'interview débute par la critique du système actuel de gestion des données du laboratoire pour analyser autant les points qui fonctionnent que les limites. La critique est formulée selon deux aspects : le système lui-même et son interface (ses usages) d'une part, la structuration des données dans le système d'autre part.
   - Des notes sur papier et sur le logiciel FreeMind [2] ont été constituées pendant les interviews. Un document de synthèse a ensuite été rédigé et soumis à l'approbation des interviewés.

4.1.2 Définition du modèle de données

L'objectif du modèle de données est de proposer une structuration des données de neuroimagerie qui permette aux utilisateurs d'accéder aux données et aux informations dont ils ont besoin pour mener leurs tâches routinières dans les meilleures conditions. Le modèle de données

1. PubMed est développé par le United States National Library of Medicine (NLM) et le National Institutes of Health (NIH). Il permet notamment un accès à la base de données MEDLINE qui contient des publications en biologie et sciences de la vie.
2. FreeMind est un logiciel libre permettant de créer des cartes heuristiques, aussi appelées *Mind Map*.

est conçu à partir des besoins identifiés, en collaboration avec deux utilisateurs finaux du système de gestion des données au GIN : Marc Joliot (directeur de recherche) et Pierre-Yves Hervé (post-doctorant).

L'interaction facilitée avec les utilisateurs finaux assure une compréhension fine des processus d'une étude pour pouvoir prendre assez de recul pour en abstraire des concepts. Les processus en neuroimagerie sont complexes, et dans un contexte de recherche les dénominations doivent être précises. A chaque proposition, les concepts sont validés en vérifiant qu'ils correspondent bien aux cas d'usage définis dans le cahier des charges, et que nul aspect n'est oublié.

Le modèle de données résultant, appelé BMI-LM pour Bio-Medical Imaging - Lifecycle Management, a été modifié de façon incrémentale après des tests de validation effectués sur son implémentation dans un système de gestion des données PLM (voir la sous-section suivante).

4.1.3 Implémentation dans un système de gestion des informations

Dans le cadre de notre thèse, le système de gestion des données choisi est le PLM Teamcenter édité par Siemens [3]. Ce choix a été motivé d'une part via la perspective d'acquisition d'une machine IRM de marque Siemens par le groupe de recherche GIN ce qui permettrait une meilleure intégration de la solution, et d'autre part en regard des compétences disponibles au sein de la société CADESIS sur la solution Teamcenter. Comme le modèle de données BMI-LM n'a pas été conçu pour ce système PLM en particulier, rien ne permet ne penser qu'il ne pourrait être implémenté dans les systèmes PLM concurrents.

Le modèle de données BMI-LM est implémenté dans le système PLM pour être testé en deux étapes principales :

1. Validation de la structuration des données proposée par le modèle.
   Des données d'études de recherche en neuroimagerie sont migrées dans le système PLM. Ces données proviennent de la base de données précédente du GIN.
2. Validation du modèle de données en contexte d'utilisation.
   Un système PLM est déployé pour permettre la gestion des données du laboratoire en contexte d'acquisition et de fonctionnement routinier (traitement, validation, analyse et publication de données). Le système PLM est interfacé avec :
   - La grille de calcul du laboratoire pour lancer les traitements sur les données,
   - Un client pour requêter la base de données,
   - Un client pour visualiser le contenu de la base de données sous forme de graphes,
   - Le logiciel JMP, un outil de calcul statistique couramment utilisé en neuroimagerie.

   Tester le modèle de données dans ce cadre permet de vérifier sa robustesse en contexte d'utilisation.

A chaque étape de validation, les fonctions issues du cahier des charges (initial ou mis à jour à partir des retours utilisateurs) sont testées. Les retours utilisateurs sont également pris en compte, à la fois en collaboration avec les deux utilisateurs finaux qui ont contribué à la

3. www.siemens.com/teamcenter

définition du modèle de données BMI-LM, et aussi en présentant le système et ses fonctionnalités sous la forme d'une démonstration à l'ensemble du laboratoire.

Les détails de l'implémentation et les résultats qui en découlent sont présentés dans la section 7.2.

4.2 Clarification des besoins

4.2.1 Étude de la littérature

Dans le chapitre 2, nous avons présenté d'une part les caractéristiques propres à la neuroimagerie et d'autre part une étude des solutions de gestion des données utilisées actuellement par le domaine. En conclusion nous avons établi que les données et informations à gérer dans le domaine de la neuroimagerie sont principalement caractérisées par :

- Une forte hétérogénéité,
- Une provenance complexe,
- La confidentialité d'une partie des données,
- Des dénominations précises et néanmoins évolutives.

Prendre en compte ces caractéristiques premières pour définir une structuration des données et un modèle de données pertinents sont un minima.

Les caractéristiques des solutions de gestion de données actuelles en neuroimagerie ont été données dans le tableau de synthèse de la littérature 2.1. Deux catégories de solutions ont été répertoriées : les systèmes de gestion de données conçus pour un partage local ou multi-partenaires, et les bases de données à destination d'un partage massif et ouvert à tous. Un dénominateur commun à ces solutions est une volonté de rendre immédiatement familière la base de données en privilégiant le vocabulaire et les concepts du domaine, mais aussi de fournir une interface qu'il est facile de prendre en main. Toutefois, ces actions induisent souvent une limitation à la fois des données qui sont gérées dans la base et des fonctionnalités proposées. D'autres limites des solutions existantes ont été mises en évidence, quelle que soit leur catégorie :

- L'absence de gestion globale tout au long du cycle de vie d'une étude de recherche : chaque catégorie de solution gère des étapes différentes de ce cycle de vie.
- La gestion intégrée des documents dans la base de données : présence partielle ou inexistante des fichiers dans la base, des chemins indiquent leur localisation dans des sources de stockage externes.
- Une gestion fine des accès aux données : bases de données ouverte à tous ou sur contrôle d'identification mais sans considération du rôle de l'utilisateur dans l'étude.
- Des stratégies de réutilisation des données : la provenance est souvent insuffisante, ce qui limite les possibilités de réutilisation.
- Hétérogénéité limitée des données qui sont gérées : les solutions ne permettent de gérer que les données d'une seule discipline.
- La flexibilité du modèle de données : difficulté à intégrer de nouveaux concepts sans

effectuer de modifications majeures sur la base de données.

4.2.2 Interview d'un groupe de chercheurs

Le document de synthèse rédigé à l'issue des interviews est proposé en annexe C. Les points principaux mis au jour pendant les interviews peuvent être regroupés en deux catégories, selon qu'ils concernent plutôt les relations entre les données ou l'interface du système de gestion des données et les possibilités qu'il offre.

1. Structuration des données
   - Complétude de la structure : toute la chaîne de données doit figurer dans la base.
   - Transparence de la structure : les utilisateurs peuvent utiliser le système de façon ponctuelle sans connaitre tous les détails de l'organisation des données.
   - Flexibilité de la structure : possibilité de rajouter de nouveaux types de données et de nouveaux attributs. Par exemple la base de données du laboratoire au moment de commencer nos travaux de recherche ne pouvait accueillir que des données liées à l'imagerie, aux tests psychologiques et aux informations cliniques. Mais dans le futur la base devra également accueillir des données génétiques, voire d'autres types de données si des partenariats supplémentaires devaient avoir lieu.
   - Segmentation des données : les données doivent être identifiables indépendamment les unes des autres.

2. Utilisation du système de gestion des données
   - Construction de requêtes : possibilité de former des requêtes complexes rapidement et facilement.
   - Visualisation de la provenance : les utilisateurs doivent pouvoir accéder facilement à la traçabilité d'une donnée et son contexte de création.
   - Lancement d'analyses : des traitements sont lancés depuis la base de donnée vers des grilles de calcul et les résultats sont remontés automatiquement.
   - Utilisation occasionnelle : un utilisateur novice doit pouvoir utiliser les fonctionnalités principales offertes par l'interface sans avoir besoin de suivre une formation longue.

4.2.3 Cahier des charges

A partir des conclusions sur l'étude des solutions existantes et des points soulevés lors des interviews, nous définissons le cahier des charges d'un système de gestion des données pour la neuroimagerie fonctionnelle.

Les étapes d'une étude d'imagerie médicale peuvent être modélisées comme un cycle qui constitue le cycle de vie d'une étude de recherche, depuis l'étape 1 jusqu'à l'étape 4. A chaque étape sont générées des données dont la provenance doit être renseignée dans sa totalité pour permettre de les retrouver et de les réutiliser dans d'autres contextes. Un schéma SADT des quatre phases d'une étude de recherche est proposé en figure 4.2 : il montre à la fois les entrées et sorties de chaque étape et les informations qui permettent de définir chaque étape.

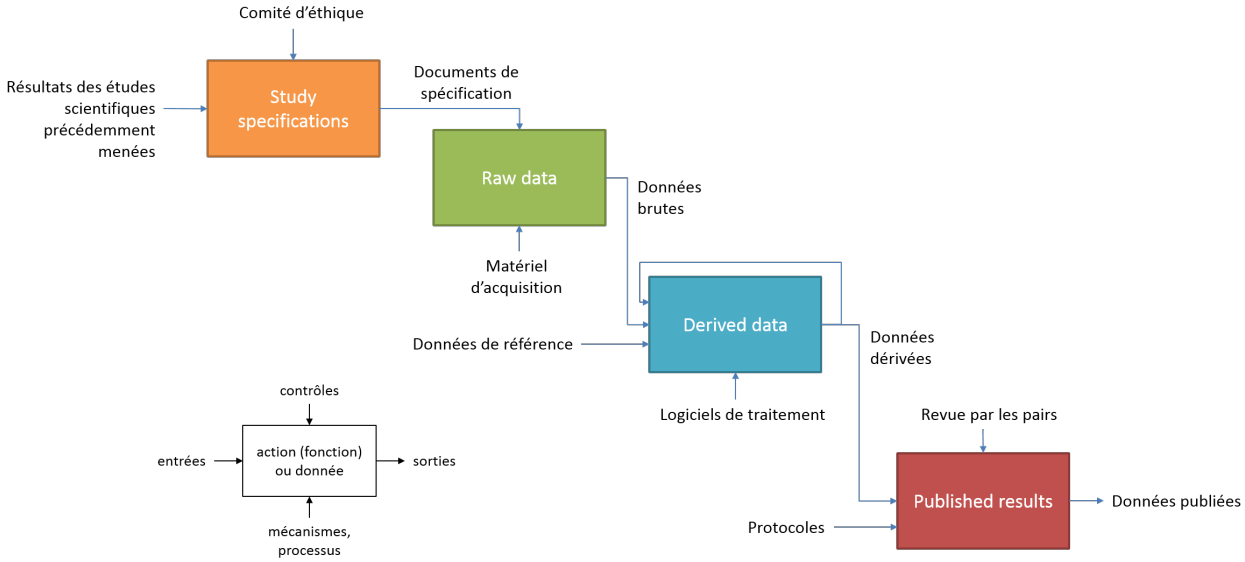


FIGURE 4.2 – Schéma SADT des quatre phases d'une étude de recherche

A la suite de la clarification des besoins, un système de gestion des données dédié aux données des études de recherche en neuroimagerie doit remplir les quatre conditions suivantes :

- Gestion les données tout au long des quatre étapes depuis les spécifications d'une étude jusqu'à la publication des résultats.
- Partage des données selon un management des accès efficace.
- Extraction et réutilisation des données à travers un modèle de données transparent pour les utilisateurs.
- Flexibilité suffisante pour permettre des évolutions de la définition du modèle de données.

Une gestion des données en neuroimagerie doit pouvoir s'adresser à deux catégories d'utilisateur : les utilisateurs occasionnels qui consultent les données, et les utilisateurs réguliers qui créent de nouvelles données. Les utilisateurs proviennent de disciplines différentes et doivent cependant se comprendre au sein d'un même modèle. Un soin tout particulier doit être apporté pour éviter les écueils classiques liés à l'interdisciplinarité, comme une incompréhension de vocabulaire ou des conventions différentes.

Les systèmes PLM assurent les fonctionnalités suivantes : gestion des accès, gestion intégrée des fichiers, suivi de la traçabilité, lancement de workflows, gestion du cycle de vie. Les systèmes PLM permettent de gérer le cycle de vie d'un produit uniquement. Il faut donc adapter le modèle de données du PLM pour que les spécificités d'une étude de recherche soient prises en compte. Si les systèmes PLM permettent une flexibilité minimale de la structuration des données, celle-ci doit être repensée pour les besoins d'un travail aussi évolutif qu'est la recherche. La provenance d'une données en neuroimagerie est complexe. Connaître le contexte d'une donnée, c'est-à-dire le cadre dans lequel elle a été générée, et sa traçabilité, c'est-à-dire d'où la donnée provient, ne suffit pas. Il est également nécessaire de connaître la définition de la donnée, c'est-à-dire son identité conceptuelle qui est donnée par les processus et outils utilisés pour la générer. Les

systèmes PLM dans leur forme actuelle ne permettent de gérer que le contexte et la traçabilité, ce qui fait de l'identification des données un verrou à lever.

Pour pouvoir utiliser un système PLM dans la gestion des données en neuroimagerie, il apparait donc nécessaire de concevoir une structuration et un modèle de données associé qui soit non seulement adapté au *cycle de vie d'une étude* en neuroimagerie, mais qui permette également de gérer efficacement la *provenance* des données, leur *hétérogénéité* et qui reste *flexible* face aux évolutions de la recherche. Ce modèle de données est présenté dans la suite du chapitre.

4.3 Le modèle de données BMI-LM

Le modèle de données présenté dans cette section, appelé BMI-LM pour Bio-Medical Imaging – Lifecycle Management, s'articule autour de trois axes principaux : *provenance*, *hétérogénéité* et *flexibilité*. Les trois sous-sections suivantes exposent chacune un de ces axes.

Les systèmes PLM permettent de gérer des objets – au sens informatique du terme –, qui sont des concepts génériques auxquels sont associées les données. Le modèle de données BMI-LM est donc un modèle orienté-objet, et dans la suite du document on désigne en raccourci par *objet* un objet du modèle de données BMI-LM. Dans les systèmes PLM, l'objet principal est appelé *article*. Une caractéristique de cet objet est qu'il est *révisable*, c'est-à-dire que son contenu peut être mis à jour, et que ces mises à jours sont conservées sous forme de versions appelées *révisions*. Tous les objets qui composent le modèle BMI-LM présentent les mêmes caractéristiques et fonctionnalités qu'un article, en sus de leurs propriétés propres. Les objets du modèle sont présentés dans la table 4.1 ; à chacun est associé un sigle et une icône qui permettent les identifier plus rapidement lors de la navigation dans les données de la base. Les icônes utilisées sont libres de droit. Une description complète du modèle de données BMI-LM implémenté dans le système PLM Teamcenter est donné dans l'annexe D.

4.3.1 Deux catégories d'objets pour garantir la *provenance*

Les objets sont divisés en deux catégories, selon leur rôle dans l'organisation des données : les objets *résultat* et les objets *d'identification*.

La distribution des objets en fonction de la catégorie et de l'étape de l'étude auxquelles ils sont associés est présentée dans la figure 4.3.

4.3.1.1 Objets résultat

Sept objets permettent de stocker des données résultat, brutes ou traitées, sous la forme de fichiers et/ou de métadonnées. Ces objets sont *Study*, *Study Subject*, *Exam Result*, *Acquisition Result*, *Data Unit Result*, *Processing Result* et *Processing Unit Result*. Ils ne peuvent être définis que dans le contexte d'une étude et appartiennent à celle-ci. Chaque objet est conceptuellement associé à une étape de l'étude.

Objet	**Sigle**	**Traduction française**	**Description**	**Icône**
Acquisition	ACQ	Acquisition	Période indivisible d'acquisition de données	
Acquisition Definition	ACD	Définition d'une acquisition	Description d'un protocole d'acquisition	
Acquisition Device	AQD	Dispositif d'acquisition	Description d'un dispositif utilisé pendant un examen	
Bibligraphical Reference	BBR	Référence Bibliographique	Article scientifique	
Data Unit Result	DUR	Unité de données	Donnée acquise isolée	
Data Unit Definition	DUD	Définition d'une unité de données	Description d'une unité de données	
Exam	EXA	Examen	Ligne continue d'acquisitions	
Exam Definition	EXD	Définition d'un examen	Description de la chaîne d'acquisitions	
Subjects Group	SGP	Groupe de sujets (dans l'étude)	Ensemble de sujets dans l'étude regroupés selon un critère (brut ou dérivé)	
Processing Definition	PCD	Définition d'une chaîne de traitement	Description d'une chaîne de traitement	
Processing	PCR	Chaîne de traitement	Chaîne de traitement	
Processing Parameter	PCP	Paramètres de traitement	Jeu de paramètres utilisés pour un traitement	
Processing Unit Definition	PUD	Définition d'une unité de traitement	Description d'une unité de traitement	
Processing Unit Result	PUR	Unité de traitement	Traitement effectué sur des données	
Reference data	RFD	Donnée de référence	Donnée d'entrée d'un traitement hors du contexte d'une étude	
Software Tool	STL	Logiciel	Description d'un logiciel de traitement	
Study	STU	Etude	Etude de recherche	
Study Subject	SSU	Sujet dans l'étude	Sujet dans le contexte d'une étude	
Subject	SUB	Sujet	Sujet unique dans la base	

TABLE 4.1 – Objets du modèle de données BMI-LM

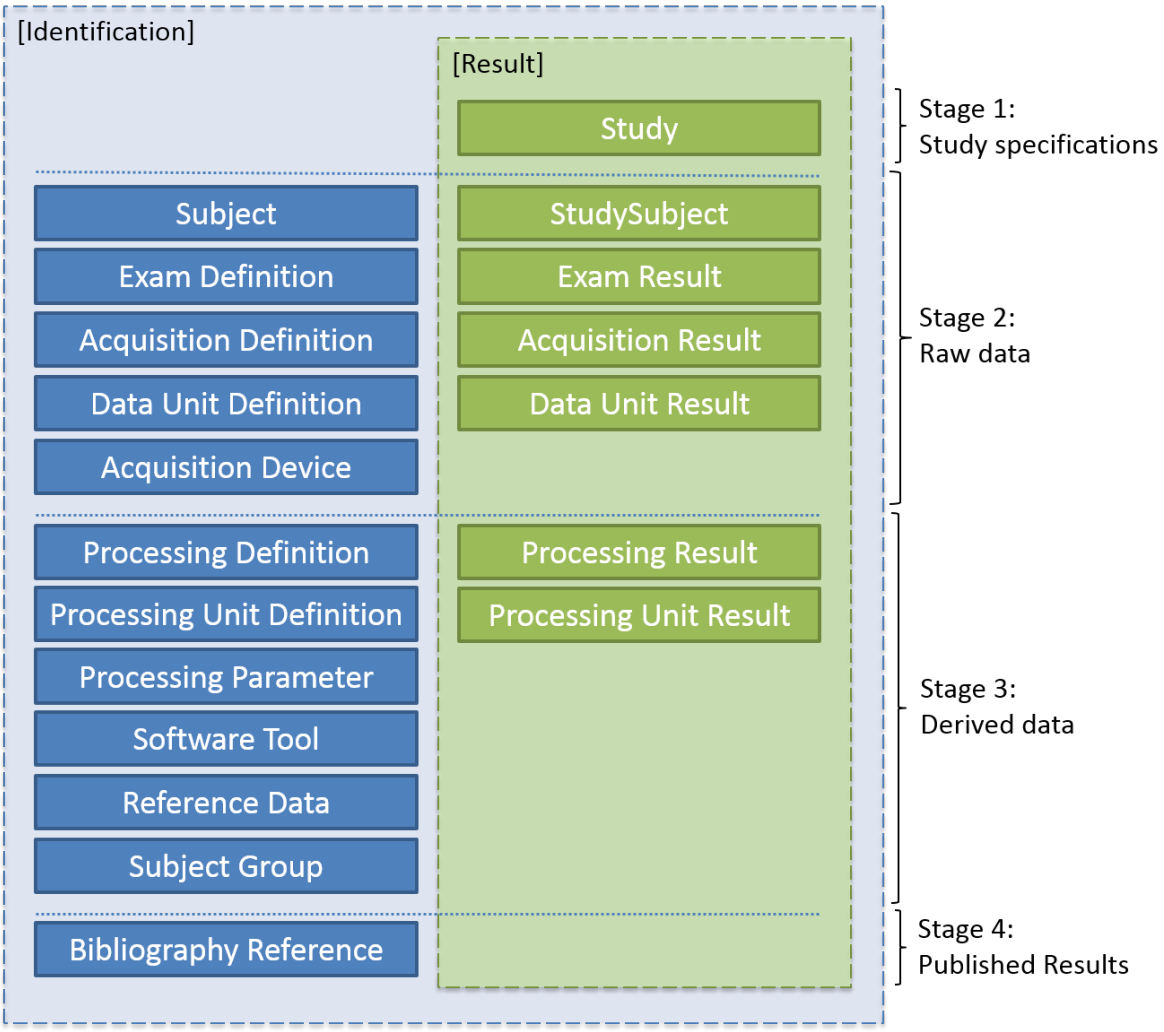


FIGURE 4.3 – Répartition des objets du modèle de données BMI-LM en fonction de leur catégorie et de l'étape d'une étude de recherche pour lesquels ils sont utilisés

Individuellement, ces objets ne présentent pas de provenance : ils contiennent les données d'intérêt principal et tout l'enjeu consiste à leur fournir une traçabilité et une identification les plus complètes pour que les données qu'ils contiennent puissent être réutilisées dans un contexte différent. La traçabilité est obtenue grâce aux relations entre les objets résultat, que ce soit à l'intérieur d'une étude ou entre plusieurs études.

4.3.1.2 Objets d'identification

Douze objets vont permettre d'identifier les objets résultat : ils décrivent comment les données ont été obtenues. Ces objets sont *Subject*, *Exam Definition*, *Acquisition Definition*, *Data Unit Definition*, *Acquisition Device*, *Processing Definition*, *Processing Unit Definition*, *Processing Parameter*, *Software Tool*, *Reference Data*, *Subject Group*, *Bibliography Reference*. Ils peuvent être créés à n'importe quel moment et sont réutilisables d'une étude sur l'autre. Chaque objet d'identification est néanmoins conceptuellement associé à une étape du cycle de

vie des études.

Naturellement, le système PLM fournit le contexte d'un objet, et les relations entre objets fournissent la traçabilité. Les objets d'identification complètent la stratégie de définition de la provenance d'un objet en fournissant la dernière composante de la provenance. Les objets d'identification peuvent eux-aussi présenter une provenance complète - contexte, traçabilité et identification - comme n'importe quel objet. Cependant leur traçabilité n'est pas obtenue grâce à des relations avec des objets résultat, mais avec d'autres objets d'identification.

4.3.1.3 Objets ambivalents

Les objets *Reference Data*, *Bibliographical Reference* et *Subject Group* sont des objets d'identification que nous qualifions d'ambivalent, car leur emploi est multiple dans l'établissement de la provenance des objets. Ils constituent la traçabilité et ils permettent d'identifier le contenu de tous les objets, à la fois d'identification et résultat.

Contrairement aux autres objets d'identification, la traçabilité des objets ambivalents peut être assurée à la fois par des objets d'identification et des objets résultat.

4.3.1.4 Relations entre objets

Les relations entre objets sont typées, dirigées et ont une cardinalité. Leur type est défini selon l'objet *enfant* (ou *secondaire*) qui est référencé par l'objet *parent* (ou *primaire*), pour rendre les requêtes plus efficaces et interdire les relations entre objets qui n'ont pas de signification dans le modèle de données. Par exemple, le sujet d'une étude (objet *Study Subject*) a passé un examen d'imagerie (objet *Exam Result*). Le sujet dans l'étude référence l'examen, et l'examen est référencé par le sujet dans l'étude. Le nom de la relation entre le sujet dans l'étude et l'examen est typé "*relation_Exam*".

Toutes les relations n'ont pas la même signification :

- Relation de traçabilité : liaison entre deux objets résultat ou deux objets d'identification. La direction de la relation se lit ainsi : un objet (primaire) est composé d'un objet (secondaire).
- Relation d'identification : liaison entre un objet résultat et un objet d'identification. La direction de la relation se lit ainsi : un objet (primaire) est identifié par un objet (secondaire).

Un schéma UML du modèle de données BMI-LM présentant les relations entre objets et leur cardinalité est proposé dans la figure 4.4.

4.3.2 Des concepts génériques pour gérer l'*hétérogénéité*

Pour éviter la surabondance d'objets dans le modèle de données et permettre de gérer des données hétérogènes, chaque objet représente un concept générique. Chaque objet est donc indépendant de toute discipline et de tout type ou format de données, mais est dépendant d'une étape des études de recherche (voir la figure 4.3).

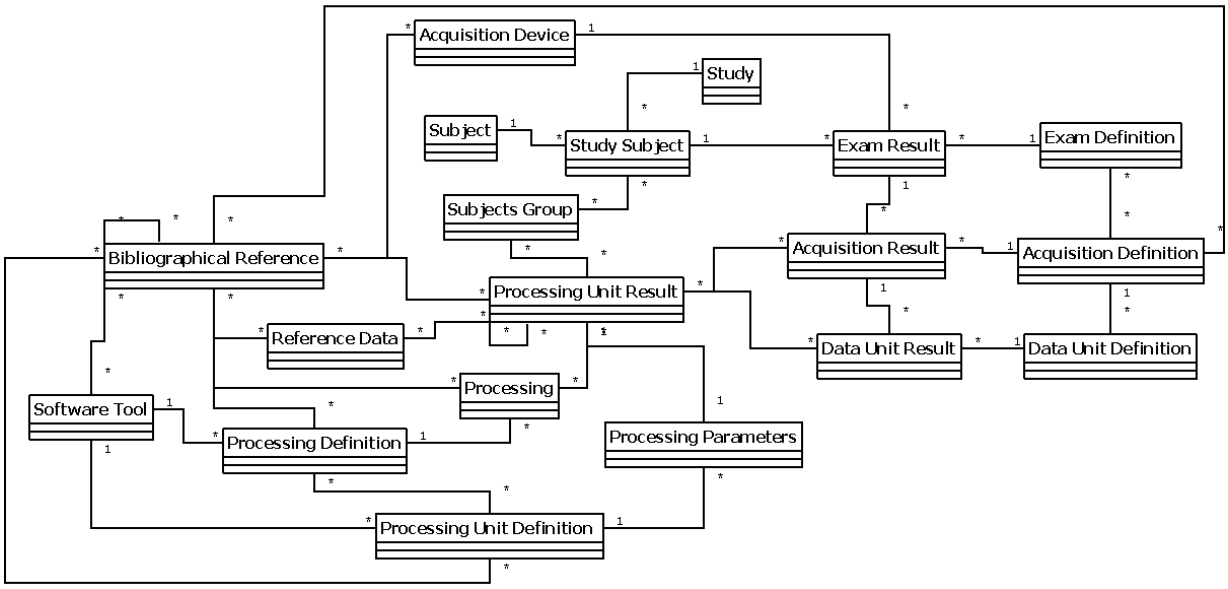


FIGURE 4.4 – Schéma UML du modèle de données BMI-LM

Nous distinguons trois grands types de concepts : les données *brutes*, les données *dérivées* et les données *de référence*.

4.3.2.1 Données brutes

Il s'agit des données d'une étude qui sont entrées dans la base de données telles qu'elles ont été recueillies, et sans avoir subi de traitement dans le système de gestion des données. Les objets concernés sont *Study*, *Study Subject*, *Exam Result*, *Acquisition Result* et *Data Unit Result* (objets résultat), *Subject*, *Acquisition Device*, *Exam Definition*, *Acquisition Definition* et *Data Unit Definition* (objets d'identification). Les étapes 1 et 2 d'une étude de recherche sont couvertes par ces objets.

Nous avons choisi d'utiliser quatre objets résultat pour stocker les données acquises lors de l'étape 2 d'une étude. Un objet pour les données invariables des sujets (par exemple la date de naissance ou le sexe), et une structure de trois objets pour les données acquises lors d'examens. C'est grâce à cette structure que le modèle devient capable de stocker des données d'examen de n'importe quelle discipline. Un examen est constitué d'au minimum d'une acquisition, elle-même constituée au minimum d'une unité de donnée. Une acquisition représente une suite continue de données acquises, tandis qu'une unité de données représente des données indivisibles.

Par exemple un examen d'IRM fonctionnelle consiste en plusieurs acquisitions (une anatomique et deux fonctionnelles) qui sont effectuées successivement sur un sujet, et il est très important de savoir qu'elles ont été acquises le même jour. Pour une acquisition fonctionnelle, les données résultantes sont naturellement composées d'images, mais il existe également d'autres informations sur le ressenti et le comportement du sujet. Les images n'ont pas de sens sans les informations de débriefing recueillies auprès du sujet après les acquisitions, et inversement. Par conséquent, les images et le formulaire de débriefing sont stockés sous une même acquisition

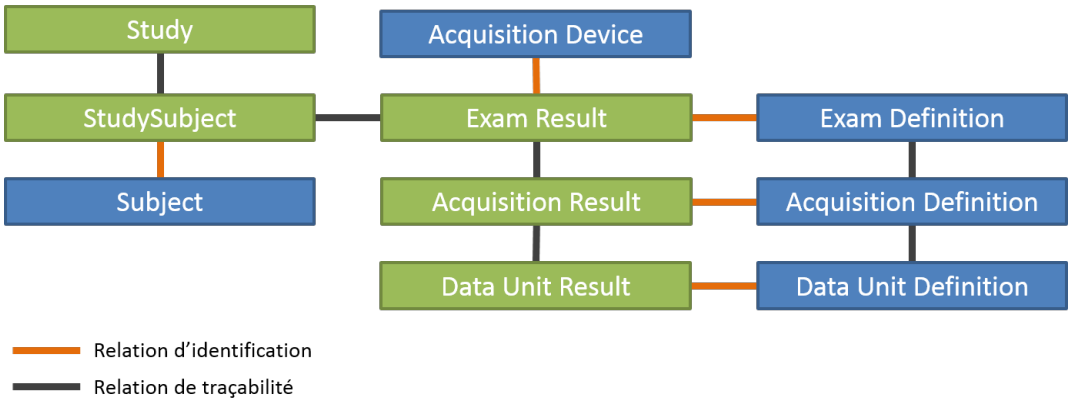


FIGURE 4.5 – Structure d'objets pour gérer l'hétérogénéité des données brutes.

(fonctionnelle), mais sous deux unités de données différentes, car ce sont sémantiquement deux données différentes.

Remarque : Les données des sujets doivent être rendues anonymes avant d'être importées dans une base de données. L'existence d'une référence publique unique pour chaque sujet dans la base grâce à l'objet *Subject* permet de savoir si un sujet anonyme a participé à plusieurs études, tandis que l'objet *Study Subject* stocke les information d'un sujet dans le contexte spécifique d'une étude de recherche et peut demeurer privé.

4.3.2.2 Données dérivées

Il s'agit de données issues d'un traitement. Les objets concernés sont *Processing Result* et *Processing Unit Result* (objets résultat), *Processing Definition*, *Processing Unit Definition*, *Processing Parameter* et *Software Tool* (objets d'identification). L'étape 3 d'une étude de recherche est couverte par ces objets.

La structure de stockage des données dérivées est simple : une unité de traitement sert à stocker les résultats d'un traitement, et le second objet résultat représente une chaine de traitements.

Par exemple avant d'être analysées, les images issues des acquisitions IRM doivent être traitées à la fois pour repositionner correctement les images dans le temps et dans l'espace, et pour faire apparaitre des contrastes. Une chaine de traitement est utilisée pour calculer les images dérivées qui constituent une base incontournable à la suite de l'étude.

4.3.2.3 Données de référence

Les objets ambivalents *Reference Data*, *Bibliographical Reference* et *Subject Group* (voir section associée 4.3.1.3) contiennent des données de référence, qui sont réutilisées d'une étude sur l'autre et qui sont employées pour identifier les objets à la fois pour les données brutes et dérivées. Aucune structure particulière ne leur est associée.

FIGURE 4.6 – Structure d'objets pour gérer l'hétérogénéité des données dérivées.

4.3.2.4 Exemple général

Le schéma de la figure 4.7 présente un exemple de l'organisation des données pour deux sujets (sujet n°12 et sujet n°34) dans une étude (étude n°1). Les instances d'objets résultat sont représentés en rose et celles des objets d'identification en gris. Les icônes et les sigles proposés pour chaque objet dans le tableau 4.1 sont repris sur le schéma pour faciliter la lecture.

Le sujet dans l'étude n°12 est identifié de façon unique dans la base comme l'humain n°28. Ce sujet a passé deux examens pendant l'étude : un examen d'imagerie (EXA#02) et un examen de psychologie (EXA#01). L'examen d'imagerie est composé de deux acquisitions, une anatomique (ACQ#03) et une fonctionnelle (ACQ#04). S'il n'y a qu'une unité de données associée à l'acquisition anatomique, trois unités de données sont reliées à l'acquisition fonctionnelle : l'image IRM fonctionnelle du sujet (DUR#06), le diagramme de stimulation temporelle correspondant à cette image (DUR#07) et un questionnaire auquel le sujet à répondu à l'issue de l'acquisition (DUR#08).

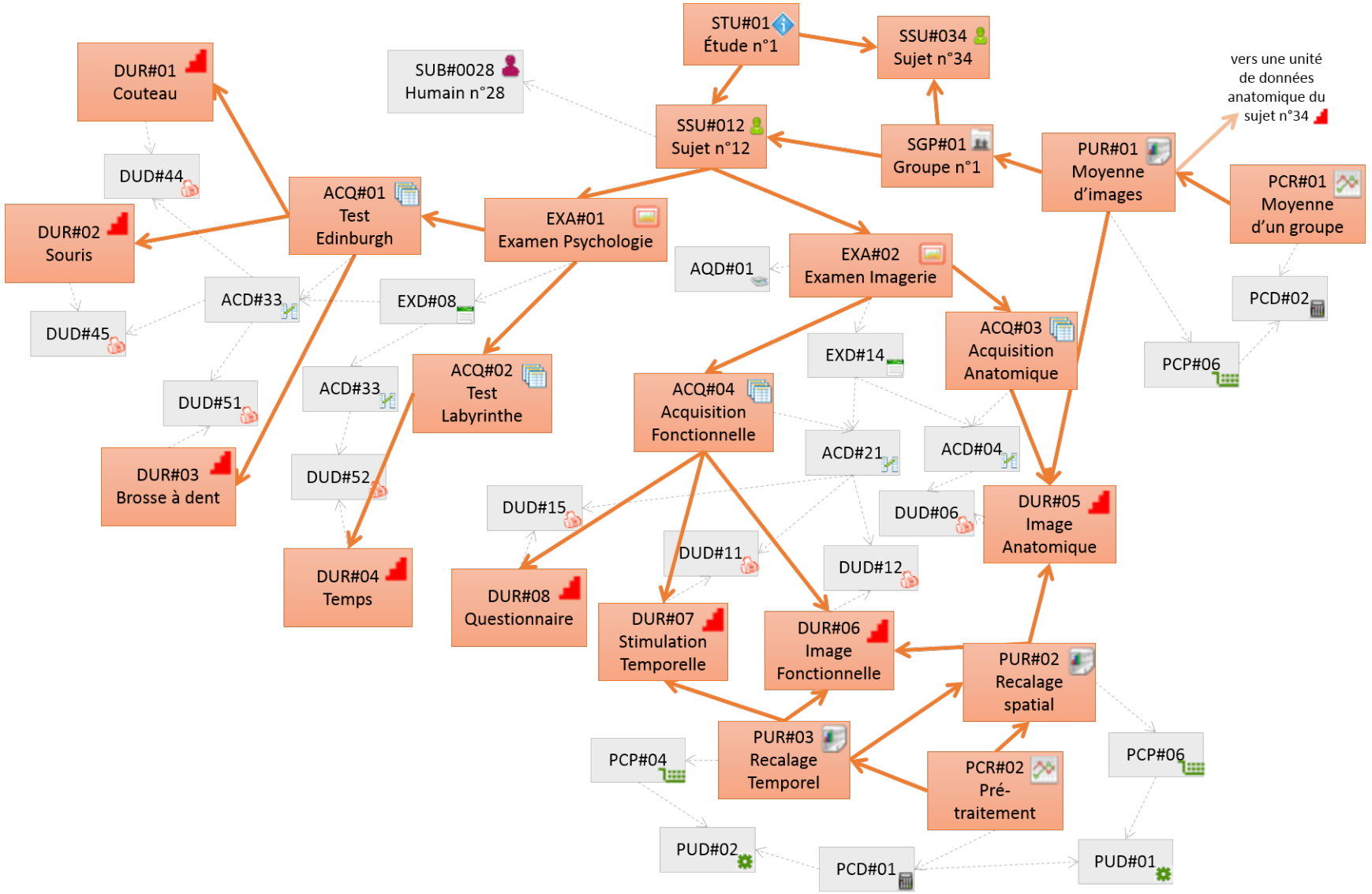


FIGURE 4.7 – Schéma exemple présentant une partie des données de deux sujets associés à l'étude n°1

L'unité de donnée contenant l'image issue de l'acquisition IRM fonctionnelle (DUR#06) a été traitée en recalage temporel (PUR#03), puis en recalage spatial (PUR#02). Ces deux traitements ont été réalisés à la suite au cours d'une même chaine de traitement (PCR#02). Pour le traitement de recalage spatial de l'image fonctionnelle, l'image de l'acquisition anatomique (DUR#05) est utilisée en entrée.

Les données d'examen du sujet n°34 ne sont pas montrées sur le schéma. Par contre, il est indiqué que les sujets n°34 et n°12 constituent un groupe de sujet et que les données de leur acquisition anatomique sont utilisées pour produire une donnée dérivée qui consiste en une moyenne d'images (PUR#01).

4.3.3 Des classes spécifiques comme support de la *flexibilité*

Dans la section précédente, nous avons présenté les concepts génériques associés aux objets qui permettent de gérer n'importe quel type de données. Ces objets à eux seuls ne suffisent pas, car il est nécessaire de conserver des métadonnées liées au type, et aussi d'associer des attributs aux objets pour stocker des informations.

4.3.3.1 La classification, une structure évolutive

Si les grandes étapes d'une étude de recherche sont stables, les méthodes, les protocoles et les métadonnées associées sont elles amenées à évoluer régulièrement. L'hétérogénéité grandissante des données implique également de définir facilement de nouveaux attributs associés aux objets, lors de l'ajout d'une nouvelle discipline dans la base par exemple.

Afin de rendre flexible la spécification d'un objet et donc la définition de ses attributs, des *classes* spécifiques sont associées aux objets génériques. Une *classe* définit les attributs contenus par un objet, et le spécifie. Toutes les classes sont organisées au sein d'un arbre hiérarchique appelé *classification* qui permet l'héritage des attributs. Chaque objet du modèle de données BMI-LM peut être classifié.

S'il est difficile de faire évoluer un modèle de données sans remettre en cause toute la structure et nécessiter l'arrêt temporaire du système de gestion des données le temps de la maintenance, la classification est modifiable à l'envie par l'utilisateur à tout moment.

La capacité des utilisateurs à requêter les instances d'objets contenues dans le système de gestion des données est améliorée par la classification, puisque un utilisateur peut chercher des instances d'objets spécifiées au sein d'un même type d'objet sans utiliser les attributs associés à celui-ci.

Toutes les disciplines en imagerie biomédicale n'utilisent pas le même vocabulaire, ou les mêmes paramètres d'acquisition et de traitement. Par conséquent une classification est dépendante d'un domaine.

4.3.3.2 Méthode de construction de la classification

La conception d'un classification n'est pas triviale, et demande un investissement substantiel en temps et en expertise.

Dans un premier temps, il convient de stabiliser les branches principales de la classification pour autoriser sans risque les évolutions des feuilles et branchages : des modifications à l'extrémité d'une branche n'impactent pas sur le reste de l'arbre. Pour cette raison, et aussi pour permettre à l'utilisateur de naviguer facilement dans la classification, il apparait nécessaire de définir une branche de la classification par type d'objet du modèle de données.

Dans un second temps, réutiliser des structures de connaissance existantes semblent pertinent : créer un nouveau lexique est coûteux en temps et ne garantie pas l'adhésion des utilisateurs aux terminologies utilisées. Nous proposons donc d'utiliser des ontologies de domaine existantes et admises dans la communauté pour construire les branches de la classification.

Un autre avantage de l'utilisation d'ontologies existantes est de faciliter un futur partage de données entre le système PLM et d'autres bases de données en neuroimagerie. En effet, une ontologie peut être utilisée comme un modèle de médiation entre les modèles de données utilisés par les bases de données.

Cependant, la liberté qu'offre l'usage de la classification pour l'utilisateur ne doit pas dispenser d'une maintenance de cette classification, bien au contraire. Il faut notamment veiller à mettre à jour la classification lorsque les ontologies utilisées sont elles-mêmes mises à jour, et il faut également veiller à ce que l'utilisateur ne duplique pas des classes déjà existantes.

Dans le cas de domaines multidisciplinaires comme la neuroimagerie, il est nécessaire d'utiliser plusieurs ontologies pour couvrir l'hétérogénéité des données. Un écueil à prendre en compte est la redondance de certains concepts d'une ontologie à l'autre (bien que ne portant pas le même nom) ou au contraire l'apparition de fausses concordances si un même mot reflète des concepts différents dans plusieurs ontologies.

4.3.3.3 Classification pour la neuroimagerie

Une classification pour la neuroimagerie a été conçue en collaboration avec Pierre-Yves Hervé, post-doctorant au GIN. Nous avons utilisées plusieurs ontologies orientées neuroimagerie qui sont activement utilisées par la communauté (Temal *et al.* (2008)) : OBO, OCRe, QIBO, OntoNeuroLog, CogPo, Cognitive Atlas, NEMO, RadLex. La multiplicité des ontologies a été nécessaire, car une partie d'entre elles définissent des concepts génériques en neuroimagerie, et l'autre partie proposent des concepts très précis sur un domaine en particulier.

Une "photographie" de la classification à la date de juillet 2015 est présentée en annexe E. Un schéma de principe présentant les branches principales de la classification est présenté en figure 4.8. La branche d'identification (*Identification Branch*), la branche de résultats (*Result Branch*) et la branche de références (*Reference Branch*) sont les trois branches principales. Ces branches sont elles-mêmes divisées en sous-catégories.

Conclusion du chapitre 4

Dans ce chapitre nous avons présenté le modèle BMI-LM pour la gestion des données en neuroimagerie. Ce modèle orienté-objet a été créé pour permettre de gérer les spécificités du

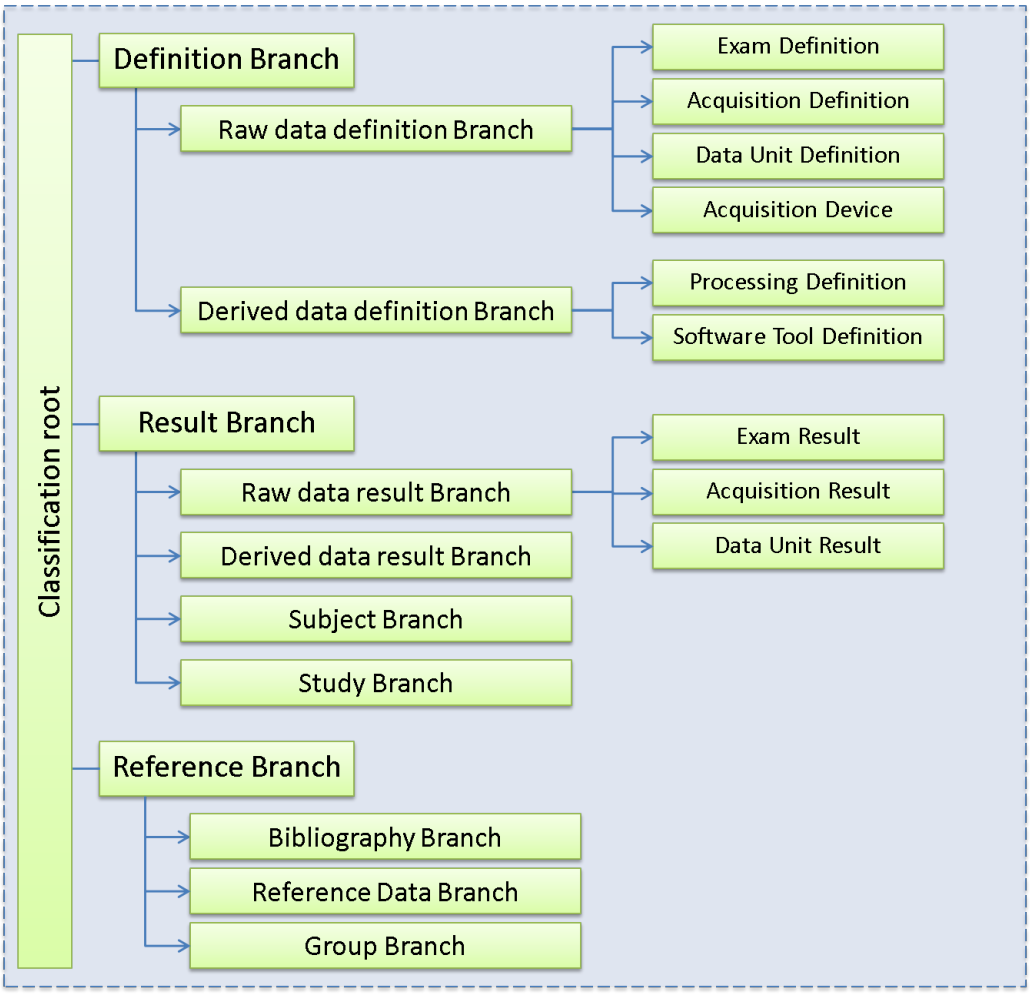


FIGURE 4.8 – Schéma des branches principales d'une classification métier associée au modèle de données BMI-LM

domaine. Il s'articule autour de trois axes principaux :

- *provenance* garantie pour l'ensemble des étapes d'une étude de recherche et entre les études,
- gestion de l'*hétérogénéité* des données au sein de concepts génériques,
- *flexibilité* du modèle de données à l'aide de classes pour permettre l'évolution de la structuration des données au grès des avancées de la recherche.

L'implémentation du modèle de données au sein du système PLM Teamcenter, ainsi que son utilisation au sein d'un laboratoire de neuroimagerie, sont détaillés dans le chapire 7.

Conscients qu'une structuration des données adaptée est un pré-requis indispensable à l'accès aux données, mais pas une condition suffisante, nous nous intéressons dans la suite du manuscrit à l'analyse visuelle de données hétérogènes. Nous définissons dans le chapitre 5 un nouveau type de graphe, le Graphe Dynamique Multidimensionnel. Puis nous proposons dans le chapitre 6 des méthodes d'analyse visuelle adaptées à ce type de graphe.

Chapitre 5

Des Graphes Multidimensionnels Dynamiques pour modéliser la complexité

Nous avons présenté dans le chapitre 1 une structuration des données et une gestion de leur provenance adaptées aux caractéristiques de la neuroimagerie au sein d'un système PLM. Les fonctionnalités classiques des systèmes PLM ne sont pas suffisantes pour pouvoir analyser les relations entre les données. Avant de s'intéresser aux techniques d'explorations visuelles, il est nécessaire de définir comment représenter les données et leurs relations.

Dans ce chapitre nous traitons de l'objectif de recherche "représenter des données multidimensionnelles et dynamiques à explorer" (voir figure 5.1). Nous présentons le *Graphe Dynamique Multidimensionnel* (GDM) qui sert à représenter des données hétérogènes multidimensionnelles aux relations complexes. Une taxonomie des tâches pour la visualisation est ensuite présentée pour compléter la spécification des GMD. Pour finir, le format d'échange JGEX (pour JSON Graph EXchange) est développé pour permettre de sauvegarder et de visualiser les GMD.

Sommaire

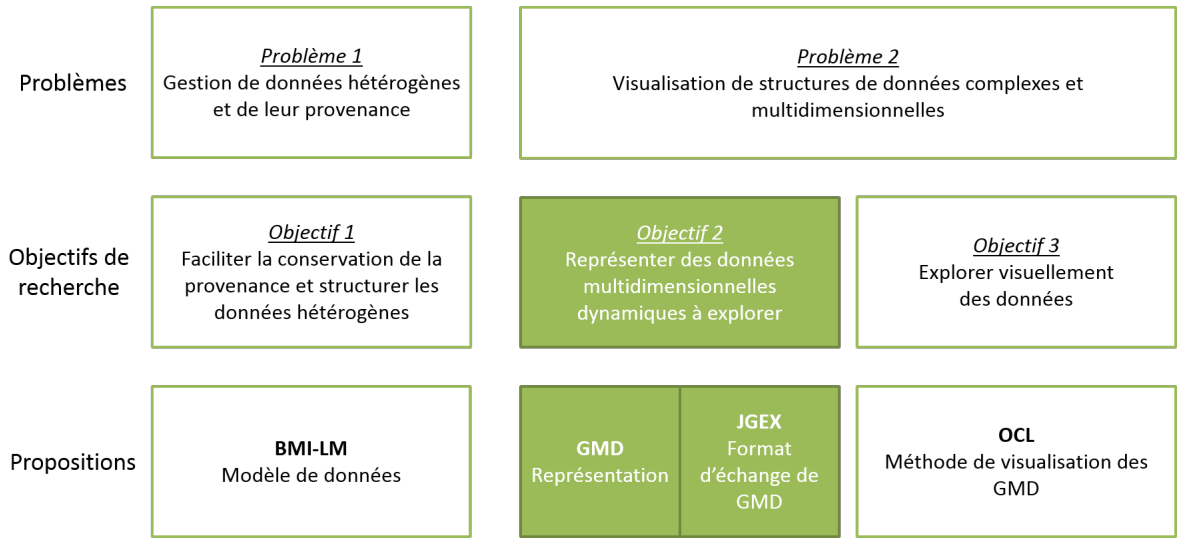


FIGURE 5.1 – Axe 2 : Structurer des données multidimensionnelles dynamiques à analyser.

5.1 Graphes Multidimensionnels Dynamiques

5.1.1 Méthode

Nous avons mis en évidence dans le chapitre 1 le besoin de visualiser des données qui évoluent selon plusieurs dimensions. D'après étude de la bibliographie du chapitre 3, aucune définition de graphe ne permet de représenter ces données complexes évolutives et multidimensionnelles. Les définitions existantes de graphes ne permettent d'analyser qu'une partie des données à la fois : soit un graphe statique (par exemple, le graphe de cerveau du sujet n°3), soit un graphe dynamique temporel (par exemple, le graphe de cerveau du sujet n°3 aux temps d'acquisition t1-t20).

Il apparait donc indispensable de définir un type de graphes adapté, que nous nommons *Graphe Multidimensionnel Dynamique* (*GMD*).

Nous choisissons de partir des définitions de graphe existantes pour spécifier les GMD.

La proposition de représentation s'articule en trois étapes :

- **La définition du GMD** : l'analyse de l'étude de cas (présentée en annexe A) et de la bibliographie permettent de clarifier les besoins et de définir les caractéristiques que doivent posséder les GMD. Les études de cas portent principalement sur deux domaines : la neuroimagerie fonctionnelle et l'étude du cycle de vie du produit.
  En parallèle, l'étude de la littérature (chapitre 3) permet de vérifier qu'une modélisation adéquate n'existe pas déjà, et de définir formellement les GMD par rapport aux définitions de graphes couramment utilisées.
- **Une taxonomie des tâches en visualisation adaptée aux GMD** : c'est-à-dire les tâches que l'utilisateur exécute lors de l'exploration d'un GMD. A partir des taxonomies des tâches pour la visualisation des graphes dynamiques présentées dans l'étude de la littérature du chapitre 3, la granularité et les spécificités des tâches de visualisation des GMD sont développées. Un espace de conception (design space) est présenté en synthèse : il permettra aux concepteurs de méthodes de visualisation d'identifier les tâches auxquelles elles répondent.
- **Le développement d'un format de stockage et d'échange** : il doit avoir la capacité de prendre en charge les particularités des GMD pour éviter les réductions ou pertes d'information. Pour déterminer quel format d'échange est le plus adapté aux GMD, une étude est menée pour analyser leurs caractéristiques. La liste des caractéristiques est tout d'abord définie en se basant sur une liste déjà existante, et en complétant avec les besoins liés aux GMD.
  Si au terme de l'étude aucun format de graphe ne satisfait les critères ou ne peut être étendu pour y satisfaire, il convient de définir un nouveau format. La méthode appliquée est alors, en partant de la liste des critères établis, de chercher à satisfaire aux besoins.

Chaque étape est validée indépendamment : représentation de données complexes à l'aide d'un ou de plusieurs GMD, stockage des données GMD. L'implémentation de ces propositions et les résultats associés sont présentés dans le chapitre 7.

5.1.2 Exemple

Pour présenter les concepts associés aux GMD, nous introduisons un exemple de jeu de données MD à représenter sous la forme de graphes, appelé GMD-4-6, qui reprend l'ensemble des spécificités MD : les données présentent plus d'une dimension d'évolution, ces dimensions ne sont pas continues et ont des attributs.

Les figures 5.1, 5.2, 5.3, 5.4 et 5.5 présentent des visualisations partielles des données de cet exemple. La totalité du jeu de données GMD-4-6 est proposée en annexe F.

5.1.2.1 Description des données

Une étude est menée pour déterminer si l'âge ou la latéralité ont un impact sur la connectivité entre les régions du cerveau humain. Un atlas de quatre régions est utilisé pour segmenter le cerveau des sujets. Trois sujets passent deux acquisitions d'imagerie IRM successives, l'une au repos (t1) et l'autre avec écoute d'une musique (t2). Après traitement des données d'imagerie, une matrice d'adjacence représentant la connectivité entre les régions du cerveau (absence de connectivité = 0, présence de connectivité = 1) est obtenue pour chaque acquisition, soit six matrices d'adjacence pour les trois sujets.

Le but de l'étude est de déterminer si l'âge ou la latéralité ont un impact sur la connectivité cérébrale en fonction de la tâche effectuée par le sujet.

L'âge et la latéralité des trois sujets est donné dans la table 5.1. Le schéma de stockage des données du sujet n°001 dans les objets du modèle de données BMI-LM est présenté en figure 5.2 : seuls les objets résultats sont indiqués, et les données du problème sont indiquées en bleu. Le sujet a passé deux examens : un d'imagerie et un autre de tests psychologiques. La donnée de latéralité a été acquise pendant l'examen de tests psychologiques, et les deux acquisitions fonctionnelles t1 et t2 ont été effectuées pendant l'examen d'imagerie. Les matrices d'adjacence sont obtenues à la suite des pré-traitements effectués sur les images des acquisitions fonctionnelles. Les matrices d'adjacence du sujet n°001 sont données dans la figure 5.3.

sujet	âge	latéralité
001	25	gauche
002	23	droite
003	34	droite

TABLE 5.1 – Âge et latéralité des trois sujets de l'exemple.

5.1.2.2 Représentation des données sous forme de graphe

Pour aider à l'exploration des données, nous choisissons de les représenter par un graphe qui sera plus facile à étudier. Chaque matrice d'adjacence est transformée en graphe statique. Ces graphes sont détaillés dans la figure 5.4. Les six graphes statiques peuvent être regroupés en un graphe de quatre nœuds et six arêtes, dont l'existence des arêtes va être conditionné aux

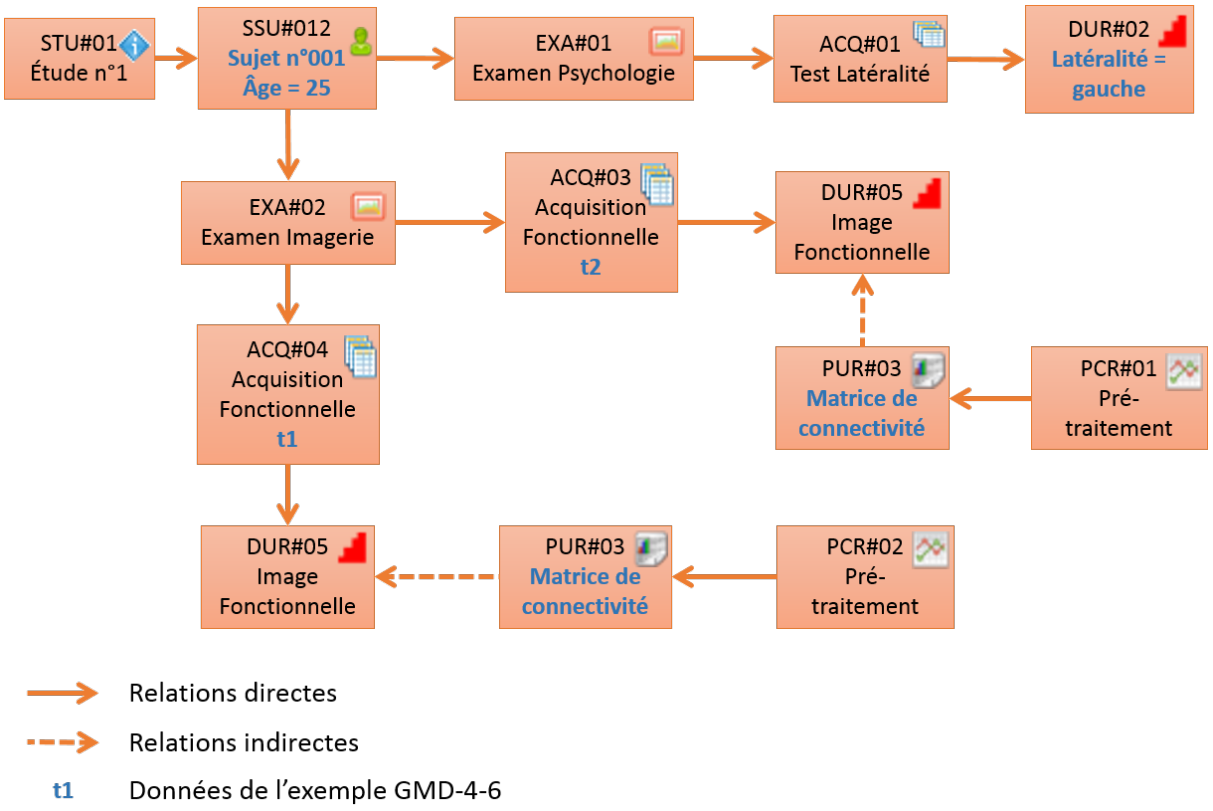


FIGURE 5.2 – Schéma des données du sujet n°001 dans le modèle de données BMI-LM (voir chapitre 4 pour les caractéristiques du modèle).

valeurs du couple {sujet, acquisition}. Les données relatives aux sujets peuvent être stockées dans un graphe statique à trois nœuds (un nœud pour chaque sujet) ; leur représentation sous forme de graphique est donnée dans la figure 5.5.

Un ensemble de graphes est donc nécessaire pour représenter les données. D'une part le graphe des données "primaires" qui décrit les données qui évoluent : les matrices d'adjacence en fonction des sujets et des acquisitions. D'autre part le graphe des données "secondaires" introduit les caractéristiques associées à l'évolution : les caractéristiques des sujets.

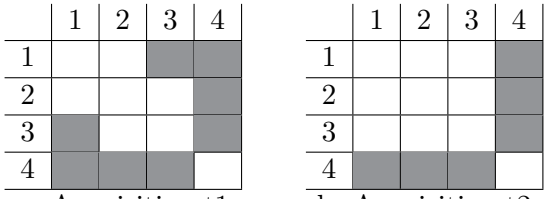


FIGURE 5.3 – Matrices d'adjacence du sujet n°001, obtenues avec une segmentation du cerveau en quatre régions.

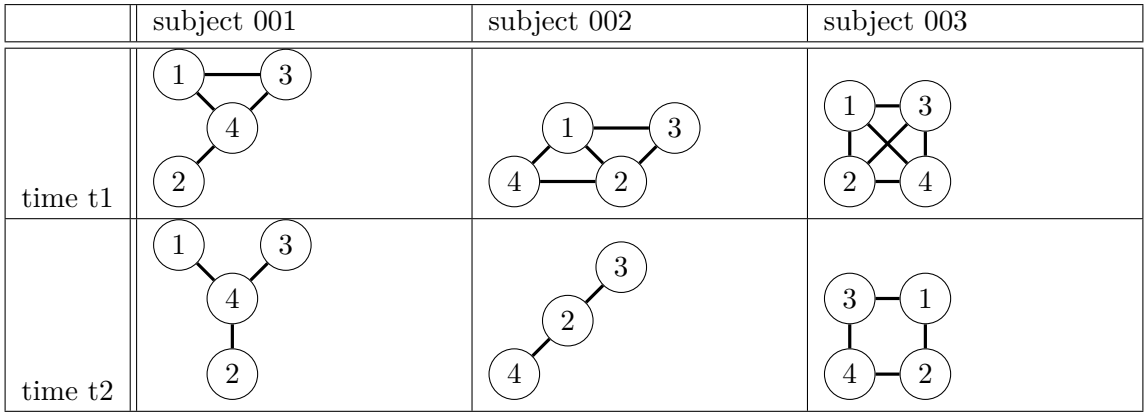


FIGURE 5.4 – Graphes statiques (représentation node-link) des six matrices d'adjacence de connectivité cérébrale obtenus dans l'étude.

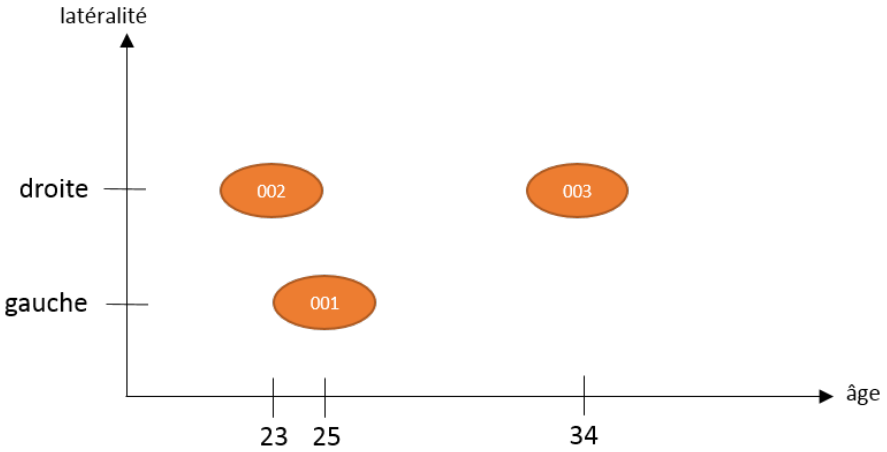


FIGURE 5.5 – Distribution des sujets en fonction de l'âge et de la latéralité.

5.1.3 Définitions

5.1.3.1 Le Graphe Multidimensionnel Dynamique (GMD)

Un *Graphe Multidimensionnel Dynamique* (GMD) est défini par une séquence de graphes :

$$\Gamma = (G_1, G_2, ..., G_n) \tag{5.1}$$

où les $G_i = (V_i, E_i)$ sont des graphes statiques dont l'indice réfère à un *moment dimensionnel* $M = (m_1, m_2, ..., m_n)$. Chaque moment dimensionnel représente un instantané du graphe sur l'ensemble des dimensions pour lesquelles il varie. Des attributs peuvent être associés à tous les éléments du graphe et au graphe lui-même. Ces attributs peuvent évoluer en fonction des dimensions.

Nous présentons le vocabulaire associé aux GMD ci-dessous. Un exemple issu du GMD-4-6 présenté dans la sous-section 5.1.2 est donné pour chaque définition. La figure 5.6 schématise

les concepts du GMD.

Dimension – Attribut variable. Une dimension peut être continue ou discrète, et représente un ensemble d'éléments. Ceux-ci peuvent être ordonnés.
Exemple : les sujets d'une étude de recherche neurofonctionnelle.

Condition d'une dimension – Élément d'une dimension. Une condition peut être continue ou discrète. Des attributs peuvent être associés à une condition.
Exemple : la dimension sujets est composée des conditions {001 ; 002 ; 003}. Le sujet numéro "001" a pour attributs {age = "25" ; handedness = "left"}.

État d'un graphe – Intersection des dimensions du GDM, définie par un ensemble de conditions.
Exemple : {subject number "001" ; "t1"}.

Configuration d'un graphe – Extraction des valeurs associées à chaque élément du graphe pour un état donné du GDM.
Exemple : chaque configuration – 6 au total pour ce graphe – peut être étudiée de façon isolée comme un graphe statique (voir figure 5.4.b).

Existence – Propriété d'un élément qui indique si un élément du graphe est défini pour un état.
Exemple : le nœud "n1" n'existe pas pour l'état (0002,t2).

5.1.3.2 Gestion de l'ordonnancement des dimensions

Les dimensions selon lesquelles varient un GMD peuvent être autant continues que discrètes. L'ordre des conditions d'une dimension peut être naturel – par exemple une dimension dérivée du temps – ou à déterminer – par exemple une liste de sujets. La répartition des conditions le long d'une dimension est donné en entrée du problème ou déterminée à partir des attributs des conditions.

Une dimension peut présenter plusieurs ordres d'étude différents. Sur l'exemple simple de GMD, la dimension *sujet* n'est pas naturellement ordonnable. Plusieurs ordres pourraient être choisis, selon la nature du problème à résoudre : l'ordre croissant sur l'identifiant du sujet – {001 ;002 ;003} – ou l'ordre croissant sur l'âge des sujets – {002 ;001 ;003}.

Le *graphe des dimensions* permet de représenter les informations associées à une dimension : chaque nœud représente une condition, et les attributs du nœud les caractéristiques de la condition. Si un ordre est associé à la dimension, il est représenté par des arêtes orientées entre les conditions. C'est la chaîne des arêtes qui indique l'ordre, et le nœud de début de chaîne est

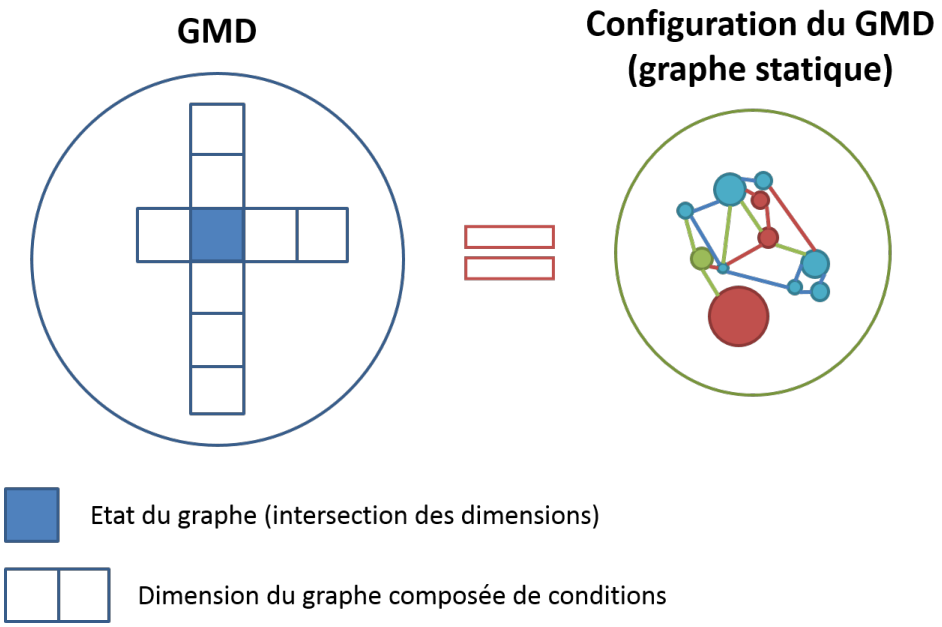


FIGURE 5.6 – Illustration des concepts associés au GMD : une configuration de GMD est obtenue en filtrant les attributs des éléments du GMD pour un état donné.

indiqué en attribut du graphe. Dans le cas où la dimension accepte plusieurs ordres, le graphe devient un GMD : les arêtes du graphe et le nœud de début de chaîne vont évoluer le long de la dimension *ordre*.

La modélisation GMD est donc constituée d'un ensemble de graphes qui va permettre de modéliser totalement la complexité de certains problèmes, comme illustré sur le figure 5.7.

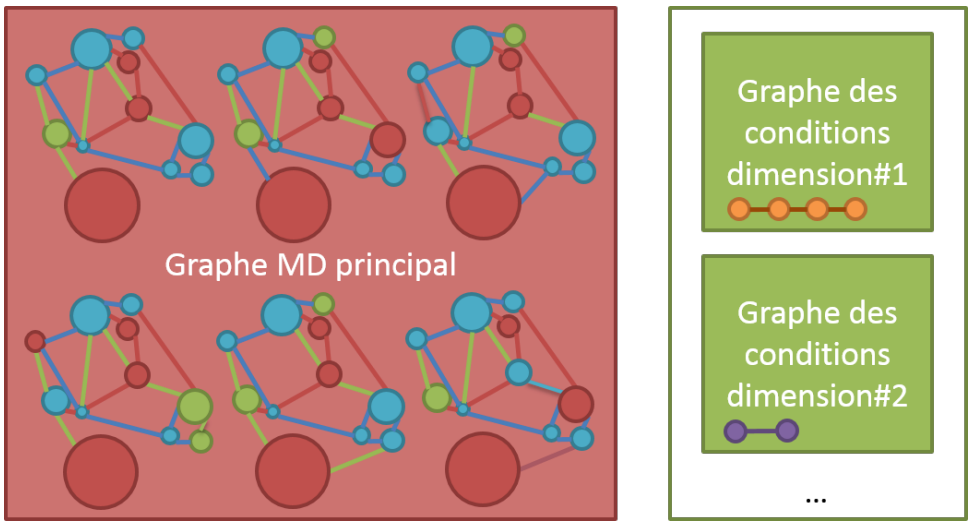


FIGURE 5.7 – La représentation des données est constituée d'un GMD ayant plusieurs états statiques, mais peut être complétée par autant de graphes des conditions (qui présentent les données associées aux conditions des dimensions) que de dimensions.

5.2 Taxonomie des tâches pour la visualisation multidimensionnelle

La taxonomie des tâches pour la visualisation des GMD présentée dans cette section s'appuie sur les taxonomies de Bach (2014) et Ahn *et al.* (2014) identifiées dans l'état de l'art présenté dans le chapitre 3. Ces deux taxonomies ont été conçues pour les tâches de visualisation des graphes dynamiques temporels, une extension dédiée aux tâches de visualisation des GMD est présentée ici.

5.2.1 Granularité de l'espace visuel

Nous identifions quatre niveaux de granularité, qui correspondent à des niveaux d'analyse, pour la visualisation des GDM :

- **niveau dimension** – l'information associée aux conditions d'une dimension ou à la dimension
- **niveau graphe** – l'information associée aux configurations ou au graphe dans son ensemble
- **niveau groupe** – l'information associée à un groupe de nœuds (défini selon un critère)
- **niveau élément** – l'information associée à un nœud ou une arête (attributs, existence)

Par rapport aux taxonomies existantes, nous intégrons la visualisation des données associées aux dimensions dans les tâches d'exploration du graphe. Les dimensions peuvent être représentées dans leur graphe propre qui peut être un graphe statique ou un GMD – si les conditions sont ordonnées plusieurs fois ou que les conditions possèdent un attribut dynamique.

5.2.2 Tâches bas et haut-niveau

5.2.2.1 Tâches bas-niveau

Nous reprenons la liste de *tâches bas-niveau* (ou low-level tasks en anglais) proposées par Lee *et al.* (2006) pour les graphes statiques, car ils correspondent aux tâches identifiées dans les domaines de l'imagerie neurofonctionnelle et du cycle de vie d'un produit (voir l'annexe sur l'analyse des cas d'étude de l'annexe A). La table 5.2 présente des exemples de tâches bas-niveau des deux domaines.

Concernant les quatre niveaux d'analyse visuelle, ils sont tous concernés par les tâches bas-niveau identifiées.

5.2.2.2 Tâches haut-niveau

Les tâches de visualisation *haut-niveau* (ou high-level task en anglais) pour les GDM que nous proposons sont proches des tâches haut-niveau pour les graphes dynamiques définies par Ahn *et al.* (2014) and Palla *et al.* (2007). Cependant, nous pensons qu'explorer un graphe selon plusieurs dimensions apporte une complexité supplémentaire à la visualisation. Des tâches haut-niveau générales sont données :

Low level task	Brain connectivity examples	Product lifecycle examples
retrieve (value)	volume of a brain region	number of holes in the product
filter	edge connectivity by 0.8	revisions of a product by date
find (extremum)	maximal connectivity through all dimensions	product with the lowest number of revisions
sort	brain regions by highest connectivity	products by latest revisions
range	edge connectivity between 0.2 and 0.5	products with pieces between 10 and 20
cluster	brain regions by connectivity rate	sub-assemblies in a product
correlate	brain regions hemisphere and connectivity	product size and number of pieces Y
scan	number of subjects	number of design revisions applied to a product
set operation	find brain regions with similarities	find products made of the same material

TABLE 5.2 – Exemples de tâches bas-niveau dans les domaines de la neuroimagerie fonctionnelle et du cycle de vie du produit présentées dans l'annexe A : étude de jeux de données multidimensionnels (selon la dénomination des tâches bas-niveau générales décrites par Lee *et al.* (2006))

- Comparaison de configurations pour révéler des motifs
- Tendance sur une ou plusieurs dimensions
- Comportement d'un élément du graphe ou du graphe selon les dimensions
- Évolution de la propriété d'un élément ou du graphe selon les dimensions

5.2.3 Comment : les trois types d'événement

Il existe trois types d'événements auxquels associer des tâches de visualisation à destination des GMD :

- Individuels – les tâches consistent à rechercher des changements au niveau individuel, selon trois axes : l'existence, les propriétés topologiques ou les attributs. Par exemple : la valeur d'un attribut augmente sur un nœud, un groupe de nœud, un graphe ou une condition.
- Successifs – les tâches portent sur la comparaison d'états successifs (dans le cas de dimensions ordonnées), dans le but d'identifier des motifs répétitifs, qui se séparent ou qui se rejoignent. Par exemple : la propriété d'un nœud qui se répète sur la dimension, un groupe de nœuds qui se sépare progressivement en deux sous-groupes au fur et à mesure des conditions, un graphe qui renforce sa caractéristique du petit monde sur plusieurs conditions, une suite de conditions qui se répète quel que soit l'ordre associé à la dimension.
- Agrégés – les tâches portent sur l'analyse visuelle de tendances et de distributions sur plusieurs conditions ensemble, principalement pour l'identification d'extrémums, d'un classement, d'une convergence ou de croissance de valeurs. Par exemple : la stabilité du poids

d'une arête sur une dimension, le groupe ayant le plus fort taux de clustering dans le graphe, le coefficient d'expansion d'un graphe.

5.2.4 Espace de conception

Afin de synthétiser les tâches de visualisation de la taxonomie, un *espace de conception* (ou *design space* en anglais) est proposé en figure 5.8. Il est constitué de deux dimensions : la granularité de l'analyse visuelle, et le type d'événements sur lequel porte la recherche. L'espace de conception permet aux concepteurs de répondre à des tâches spécifiques, car les besoins des utilisateurs peuvent être mieux identifiés.

				Granularity of visualisation analysis			
				Dimension level	Network level	Group level	Element level
Types of events and associated tasks	Aggregated events	Trends	Growth				
			Contraction				
			Convergence				
		Distribution	Rating				
			Extremum				
	Successive events	Comparison	Merging				
			Splitting				
			Repetition				
	Individual events	Attribute change					
		Topology change					
		Existence					

FIGURE 5.8 – Vue de l'espace de conception pour répondre aux tâches de visualisation pour les GDM.

5.3 JGEX : the Json Graph Exchange format

Pour pouvoir manipuler et stocker les GDM, il est nécessaire d'utiliser un format de graphes qui prenne en compte l'intégralité des données. Dans cette section nous étudions dans un premier temps les formats de graphe existants et déterminons qu'il est nécessaire de concevoir un nouveau format. Dans un second temps, nous présentons le format JGEX.

5.3.1 Etude des formats de graphes existants

De nombreux formats de graphes ont été développés, parfois en parallèle d'un outil de visualisation de graphes. Cette sous-section propose une comparaison des formats libres existants qui peuvent être trouvés sur internet. La liste n'est pas exhaustive, mais reprend les formats les plus couramment utilisés dans les logiciels de visualisation.

5.3.1.1 Critères de comparaison

Le consortium Gephi propose sur son site web une comparaison des caractéristiques des principaux formats de graphe existants [1], afin de mettre en exergue les capacités du format de graphe GEXF [2] développé par le consortium en parallèle de leur logiciel de visualisation de graphe. Huit critères de comparaison sont proposés :

- **Edge List / Matrix Structure** : le format présente les arêtes sous la forme d'une liste ou sous un format matriciel.
- **XML Structure** : le format présente les données sous la forme d'une structure en balises, ce qui permet l'ajout d'attributs sur tous les éléments du graphe.
- **Edge Weight** : le format permet le stockage des informations de pondération sur les arêtes.
- **Attributes** : la structure du format autorise l'ajout d'attributs sur les éléments du graphe.
- **Visualisation Attributes** : la structure du format autorise la définition d'attributs dédiés à la visualisation des éléments sur lesquels ils sont définis.
- **Attribute Default Value** : il est possible d'indiquer une valeur que prendra un attribut par défaut si rien n'est renseigné. Cela permet notamment d'alléger le fichier et de permettre une plus grande flexibilité.
- **Hierarchical Graphs** : un graphe hiérarchique peut être modélisé comme un arbre.
- **Dynamics** : les graphes dynamiques peuvent être stockés dans le format.

Nous sommes conscients qu'une structure basée sur XML ou JSON offre d'avantage de possibilités d'une liste d'arêtes, qui ne permet pas le stockage d'attributs par exemple. Par ailleurs, la structure de données dans un format a un impact sur les performances de ce format en lecture ou en écriture. Cependant, notre objectif est de comparer les formats pour identifier si l'un d'entre eux pourrait convenir pour le stockage et l'échange de GMD.

Nous ne gardons qu'une partie des critères proposés par le consortium Gephi, les critères que nous essayons d'établir doivent prendre en compte les caractéristiques propres aux GMD : pondéré, multi-varié, multidimensionnel dynamique, multi-graphes, graphes hiérarchiques et références entre graphes. Les critères retenus pour la comparaison des formats de graphe sont au nombre de dix :

- **Graphe pondéré** : (idem *Edge Weight*) le format permet le stockage des informations de pondération sur les arêtes.
- **Attributs** : (idem *Attributes*) la structure du format autorise l'ajout d'attributs sur les éléments du graphe.
- **Attributs de visualisation** : (idem *Visualisation Attributes*) la structure du format autorise la définition d'attributs dédiés à la visualisation des éléments sur lesquels ils sont définis.
- **Valeur par défaut d'un attribut** : (idem *Attribute Default Value*) il est possible d'in-

1. Le tableau comparatif proposée par le consortium Gephi est disponible à l'adresse : http://gephi.github.io/users/supported-graph-formats/
2. http://gexf.net/format/

diquer une valeur que prendra un attribut par défaut si rien n'est renseigné.
- **Dynamique** : (idem *Dynamics*) les graphes dynamiques peuvent être stockés dans le format.
- **Multidimensionnel Dynamique** : le format permet le stockage de graphes dynamiques à plusieurs dimensions.
- **Plusieurs graphes** : le format permet le stockage de plusieurs graphes dans un même fichier.
- **Attributs sur les graphes** : des attributs peuvent être ajoutés au niveau du graphe (ces attributs peuvent être dynamiques).
- **Groupes de nœuds** : le format permet la définition de groupes de nœuds via des super-nœuds (ou nœuds de nœuds).
- **Références entre graphes** : le format autorise les liens entre nœuds de graphes différents. Les nœuds et arêtes n'appartiennent qu'à un et un seul graphe, mais une arête pour référencer comme nœuds *source* et *cible* des nœuds qui n'appartiennent pas au même graphe qu'elle.

5.3.1.2 Tableau comparatif

Le résultat de la veille bibliographique et technique effectuée est présentée selon les critères définis précédemment dans le tableau 5.3. L'étude du tableau comparatif nous indique que peu de formats de graphe satisfont un grand nombre de critère. Le fait que peu de formats de graphe soit adapté au stockage de graphes dynamiques n'est pas étonnant, puisque ces formats ont été conçus à une époque où la visualisation de graphes statiques était en pleine expansion, cependant nous aurions pu raisonnablement nous attendre à ce qu'ils évoluent avec la multiplication des techniques de visualisation dynamique.

Le format GEXF est celui qui satisfait le plus grand nombre de critères : graphe pondéré, attributs, attributs de visualisation, valeur par défaut d'un attribut, graphes hiérarchiques, dynamique, plusieurs graphes, attributs sur les graphes et groupes de nœuds. Après avoir étudié sa structure, elle nous parait limitant. Par ailleurs, le format XML est très verbeux, et cela pourrait poser problème en terme de taille de fichiers pour stocker les graphes GMD de l'étude de cas (annexe) qui sont de grande taille.

5.3.2 Le format JGEX

Suite à l'analyse effectuée sur les formats de graphe existants, nous avons développé un nouveau format capable de stocker les GDM, qui est appelé JGEX pour Json Graph EXchange format. Le document primer[3] qui a été rédigé pour spécifier le format JGEX est proposé en annexe G. Dans cette sous-section sont présentées les principales notions du format JGEX.

Une base JSON[4] a été choisie pour développer le format JGEX, car JSON est un format

3. Un *primer* est un document non-normatif qui fournit une description d'un format de données pour qu'il puisse être facilement pris en main par de nouveaux utilisateurs.
4. JSON : JavaScript Object Notation.

	Graphe pondéré	*Attributs*	*Attributs de visualisation*	*Valeur par défaut d'un attribut*	*Graphes hiérarchiques*	*Dynamique*	*Multidimensionnel Dynamique*	*Plusieurs graphes*	*Attributs sur les graphes*	*Groupes de nœuds*	*Références entre graphes*
CSV	■										
DL Ucinet	■	■									
DOT Graphviz	■		■								
GDF	■	■	■	■							
GEXF	■	■	■	■	■	■		■	■	■	
GML	■	■	■								
GraphML	■	■	■	■	■	■				■	
NET Pajek	■	■									
TLP Tulip	■	■	■	■	■						
VNA Netdraw	■	■									
DGS	■	■									

TABLE 5.3 – Caractéristiques des formats de graphes existants

moins verbeux et plus flexible que XML. Par ailleurs, le Javascript est un langage utilisé pour le développement de clients web, comme celui qui est présenté en annexe I. Le JSON est un format qui peut être étendu via la description d'un schéma JSON [5]. Le schéma JSON du format JGEX est proposé en annexe H.

5.3.2.1 Un format multigraphes

Un schéma de la structure générale d'un fichier JGEX est présenté en figure 5.9. Le format JGEX permet le stockage de plusieurs graphes dans une collection. Les attributs sont définis en commun pour tous les graphes dans une collection située elle-aussi à la racine du fichier. Un graphe possède sept propriétés :

- *Id* : (requis) chaîne de caractères unique parmi les identifiants (Id) de tous les graphes du fichier.
- *Label* : (requis) chaîne de caractères indiquant le nom du graphe.
- *Mode* : (requis) chaîne de caractères indiquant le mode des arêtes du graphe {orientées ;non-orientées}.
- *Type* : (requis) chaîne de caractères indiquant le type du graphe.
- *Nodes* : (requis) collection des nœuds du graphe.
- *Edges* : collection des arêtes du graphe.

5. Les spécifications d'un schéma JSON sont disponibles à l'adresse : `http://json-schema.org/`

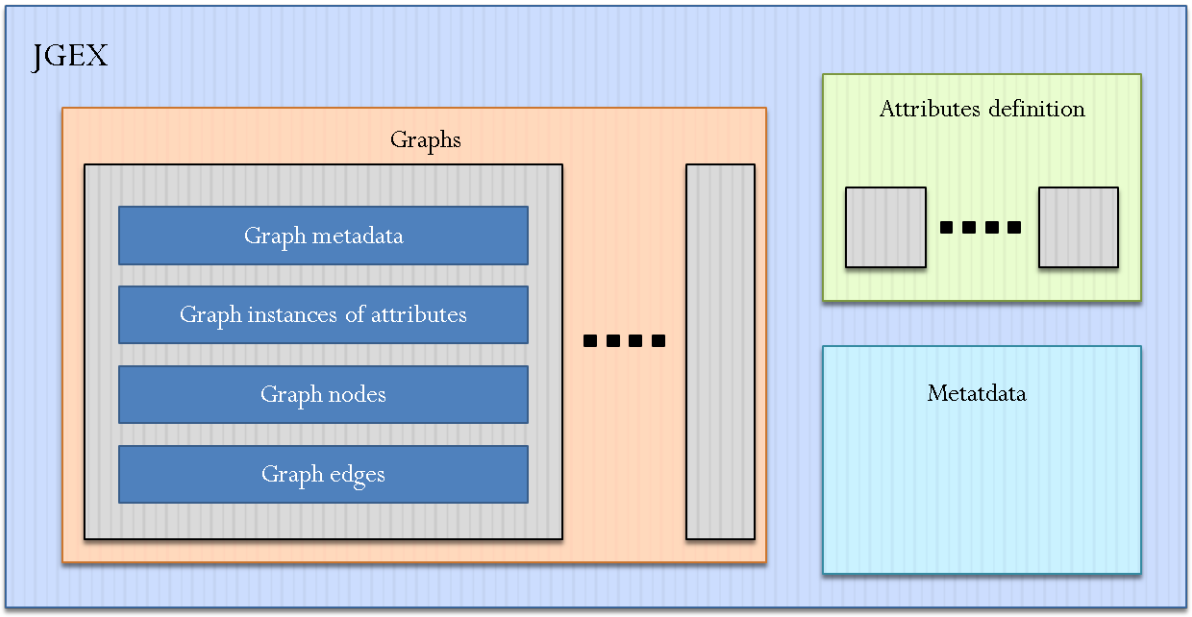


FIGURE 5.9 – Schéma de la structure générale d'un fichier JGEX

– *Attvalues* : collection des attributs du graphe.

Le nœud d'un graphe possède quatre propriétés :
– *Id* : (requis) chaîne de caractères unique parmi les identifiants (Id) de tous les nœuds du graphe.
– *Label* : (requis) chaîne de caractères indiquant le nom du nœud.
– *Type* : (requis) chaîne de caractères indiquant le type du nœud.
– *Attvalues* : collection des attributs du nœud.

L'arête d'un graphe possède sept propriétés :
– *Id* : (requis) chaîne de caractères unique parmi les identifiants (Id) de toutes les arêtes du graphe.
– *Label* : (requis) chaîne de caractères indiquant le nom de l'arête.
– *Type* : (requis) chaîne de caractères indiquant le type de l'arête.
– *Source* : (requis) référence à l'identifiant (Id) du nœud source de l'arête.
– *Target* : (requis) référence à l'identifiant (Id) du nœud cible de l'arête.
Lorsque les arêtes ne sont pas orientées, les nœuds identifiés par *Source* et *Target* sont utilisés de façon indifférenciée.
– *Weight* : (requis) valeur réelle du poids d'une arête, qui par défaut prend la valeur 1.0. Cette propriété n'est pas dynamique.
– *Attvalues* : collection des attributs de l'arête.

Le code JGEX définissant un graphe est présenté en figure 5.10.

Un nœud ou une arête ne peuvent appartenir qu'à un et un seul graphe. Cependant, une arête peut référencer comme source ou cible des nœuds qui appartiennent à d'autres graphes. Les nœuds de nœuds (ou groupe de nœuds) sont définis selon ce principe : le nœud groupe est relié à chacun des nœuds du groupe par une arête de constitution. Les nœuds du groupe peuvent appartenir au même graphe ou à des graphes différents.

```
"graphs":
        [{"id":"graphId",
          "label":"graphLabel",
          "mode":"undirected",
          "type":"graphType",
          "nodes": [{"id":"nodeId",
                     "label":"nodeLabel",
                     "type":"nodeType",
                     "attvalues":[]
                   }],
          "edges": [{"id":"edgeId",
                     "label":"edgeLabel",
                     "source":"sourceNodeId",
                     "target":"targetNodeId",
                     "weight":1.0,
                     "type":"edgeType",
                     "attvalues":[]
                   }],
          "attvalues":[]
        }]
```

FIGURE 5.10 – Code JGEX de définition d'un graphe.

5.3.2.2 Fonctionnement des attributs

Les graphes, les nœuds et les arêtes peuvent être associés à des instances d'attribut grâce à la propriété de collection *attvalues*. La définition d'un attribut possède dix propriétés :

- *Id* : (requis) chaîne de caractères unique parmi les identifiants (Id) de toutes les définitions d'attribut.
- *Label* : (requis) chaîne de caractères indiquant le nom de la définition d'attribut.
- *Scope* : (requis) chaîne de caractères indiquant la porté d'un attribut sur les éléments du graphe. Valeurs autorisées : {global, graph, node, edge}.
- *Impact* : (requis) chaîne de caractères indiquant le ou les graphes sur lesquels s'applique l'attribut : référence un graphe en particulier ou tous les graphes (prend la valeur "all").
- *Type* : (requis) chaîne de caractères indiquant le type d'attribut.
- *Structure* : (requis) chaîne de caractères indiquant le type de structure que prend la valeur d'une instance de l'attribut. Valeurs autorisées : {simple, array, map}.
- *Unit* : chaîne de caractères indiquant l'unité des valeurs de l'attribut.

```
{
  "id": "aheight",
  "impact": "all",
  "scope": "node",
  "label": "height",
  "type": "double",
  "unit": "m",
  "structure": "simple",
  "script": "",
  "condGraph": "",
  "default": 0.0
}
```

FIGURE 5.11 – Code JGEX de définition d'un attribut.

- *CondGraph* : référence le *graphe des conditions* associé à l'attribut si celui-ci est utilisé comme une dimension et que des données sont associées ses conditions. La sous-section 5.3.2.3 explique comment sont utilisées les dimensions.
- *Default* : valeur par défaut associée aux instances d'attributs.
- *Script* : chaîne de caractères pouvant contenir du code javascript.

Une instance d'attribut possède trois propriétés :

- *Attr* : (requis) référence à la définition d'attribut.
- *Value* : (requis) valeur de l'instance (type spécifié par la définition d'attribut).
- *CurrentValue* : valeur temporaire de l'instance (type spécifié par la définition d'attribut). Cette propriété est utilisée quand *value* est dynamique.

Les types de données autorisées dans le format JGEX sont les types standard : boolean, integer, float, double, object, string, date.

5.3.2.3 Gestion des dimensions

Lorsqu'une instance d'attribut est dynamique, une valeur configurée (*configured value*) remplace la valeur de *Value*. Cette valeur configurée a deux propriétés :

- *Dim* : (requis) chaîne de caractères unique parmi les identifiants (Id) de tous les attributs.
- *Ranges* : (requis) collection d'objets *range*, un range représente l'impact d'une condition pour une dimension donnée.
  - *Condition* : (requis) valeur d'une condition (type spécifié par la définition de l'attribut dimension). Deux *range* d'une instance ne peuvent pas avoir une condition identique. Si la condition est un intervalle, elle est définie par deux bornes typées ({ opened, closed, infinite }).
  - *Value* : (requis) valeur de l'instance (type spécifié par la définition d'attribut).

```
"attvalues":
        [{"attr":"aweight",
          "value":
            {"dim":"conditionId",
             "ranges":
                   [{"condition":
                       {"start":
                         {"value":"intervalStartValue",
                          "type":"infinite"}},
                        "end":
                         {"value":"intervalEndValue"
                          "type":"closed"}},
                       },
                     "value":0.64}
                   ]
                },
          "currentValue":0.27
        }]
```

FIGURE 5.12 – Code JGEX de définition d'une instance d'attribut avec valeur configurée pour une condition sous forme d'intervalle.

Dans le format JGEX, les dimensions peuvent être continues ou discrètes, et les conditions peuvent être de plusieurs types : integer, double, string ou un intervalle. Le type de la dimension est défini au niveau de la définition de l'attribut-dimension.

Les dimensions sont imbriquées les unes dans les autres, puisqu'un état est défini par un ensemble de conditions.

Il est possible dans le format JGEX d'associer des données aux conditions d'une dimension. Au moment de la définition d'un attribut, le champ *condGraph* fait référence au graphe dans lequel ces données sont stockées. Le graphe correspondant est appelé *graphe des conditions* et chaque nœud représente une condition de la dimension.

Conclusion du chapitre 5

Dans ce chapitre les Graphes Multidimensionnels Dynamiques (GMD) ont été définis. Ils servent à représenter des relations complexes entre des données, notamment des données qui évoluent selon plusieurs dimensions. Le vocabulaire associé aux GMD a été précisé, et une taxonomie des tâches visuelles a été développée pour aider à concevoir dans le futur des méthodes d'exploration visuelle adaptées aux spécificités des GMD. Afin de permettre l'échange et le stockage de GMD, par exemple entre une base de données PLM et une interface de visualisation de graphes, le format JSON Graph EXchange (JGEX) a été présenté.

Dans la suite du manuscrit, une méthode d'exploration des GMD est proposée (chapitre 6), puis appliquée à l'étude des réseaux cérébraux (chapitre 7).

Chapitre 6

Exploration de graphes multidimensionnels dynamiques

Nous avons présenté dans le chapitre 5 les GMD (Graphes Multidimensionnels Dynamiques) qui permettent de représenter les problèmes multidimensionnels et dynamiques. La représentation GMD constitue une première étape pour permettre l'exploration de données hétérogènes et de leur provenance. Dans ce chapitre nous nous intéressons à notre dernier objectif de recherche "explorer visuellement des données hétérogènes et multidimensionnelles" (voir figure 6.1).

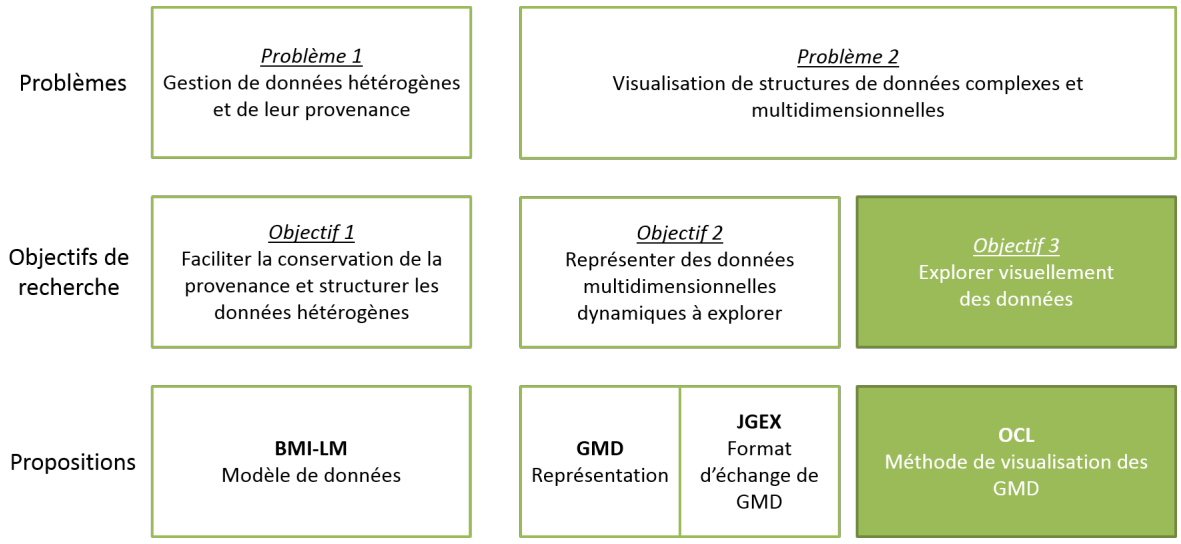


FIGURE 6.1 – Axe 3 : Méthode de visualisation de données multidimensionnelles dynamiques.

La méthode de visualisation *OCL* (*Overview Constraint Layout*) a été développée afin de pouvoir explorer les GMD. Elle reprend le principe du mantra de la visualisation de Shneiderman (1996) et se divise en deux grandes étapes : une vue d'ensemble du GMD (données réduites), puis des vues locales (données complètes) explorées dans le contexte dimensionnel.

L'exemple GMD-4-6 introduit dans le chapitre 5 est repris dans ce chapitre pour illustrer les concepts présentés (l'intégralité de l'exemple est présenté en annexe F). Dans la section 6.1, les stratégies et le scénario de la méthode d'exploration OCL sont introduits. Le vocabulaire utilisé

dans la suite du chapitre est défini, et les analyses quantitatives nécessaires à la préparation des GMD pour l'exploration sont présentées. Ensuite, les méthodes de visualisation d'un état dans le contexte local du GMD sont présentées dans la section 6.2, puis des méthodes d'analyse des états dans leur totalité et dans le contexte global du GMD sont détaillées dans la section 6.3.

Sommaire

6.1 La méthode d'exploration OCL

Dans ce chapitre nous essayons de répondre à la question : comment explorer des données multidimensionnelles et dynamiques, ces données étant représentées sous la forme de graphes ? Cette section présente dans un premier temps la démarche qui nous permet de proposer une réponse, puis nous détaillons les principes de la méthode *OCL* (*Overview Constraint Local*) : les stratégies recherchées et le scénario global. Pour finir, nous présentons les opérations de préparation des données requises pour l'application de la méthode OCL.

6.1.1 Démarche

Afin de concevoir un scénario pour l'exploration des GMD, nous organisons le travail en plusieurs étapes : 1. Clarification des besoins, 2. Définition des stratégies d'exploration, 3. Implémentation du scénario.

6.1.1.1 Clarification des besoins

Pour définir les besoins en exploration des GMD, nous croisons trois sources d'information :
- les caractéristiques des GMD (chapitre 5).
- l'espace de conception issu de la taxonomie des tâches de visualisation pour les GMD (chapitre 5, figure 5.8).
- les études de cas présentées dans l'annexe A.

Nous savons qu'il n'y a pas une "bonne" visualisation (Chi, 2000), et que les techniques de visualisation les plus appropriées dépendent des caractéristiques des graphes. Il apparait donc impossible de concevoir un scénario d'exploration et des méthodes de visualisation qui seraient adaptées à tous les types de GMD. L'objectif est de réduire la portée du problème d'exploration à un type défini de GMD.

Nous choisissons d'étudier les GMD off-line (l'ensemble des données est connu), et non complets (toutes les arêtes n'existent pas sur l'ensemble des états).

6.1.1.2 Définition des stratégies d'exploration

Pour définir des stratégies d'exploration, nous nous basons sur l'espace de conception de visualisation des GMD développé au chapitre 5 (figure 5.8) pour identifier sur quelle(s) tâche(s) de visualisation porte la méthode d'exploration.

De plus nous utilisons l'état de l'art présenté dans le chapitre 3 pour choisir les techniques de visualisation adaptées : mode de représentation du graphe, technique de visualisation dynamique et heuristique esthétique.

6.1.1.3 Implémentation du scénario

Le scénario d'exploration est implémenté dans le chapitre 7 sur des données de connectivité fonctionnelle cérébrale d'un groupe de 231 sujets qui sont stockées dans un système PLM dont le modèle de données est proposé au chapitre 4. Le client web SwoViewer est utilisé pour visualiser

les graphes ; une description de ses fonctionnalités est donné dans l'annexe I. Un serveur pour le calcul des graphes est développé en parallèle : c'est lui qui prépare les données pour l'application du scénario dans le visualiseur SwoViewer.

Les propositions de ce chapitre sont implémentées dans le serveur graphe développé dans le cadre du projet BIOMIST. Les résultats qui en découlent sont présentés dans le chapitre 7.

6.1.2 Principes de l'exploration

Suite à l'analyse des études de cas (annexe A), nous choisissons de réduire notre proposition à l'étude de GMD *non-orientés*, *acycliques*, *pondérés* et *non complets*. L'exploration proposée peut également fonctionner sur des graphes non-pondérés : il suffit d'initialiser tous les poids à la valeur 1. Le respect de la non-complétude du graphe peut conduire à l'application d'un filtre sur les arêtes, selon les considérations du domaine auquel le GMD se rapporte.

6.1.2.1 Stratégies d'exploration

Notre proposition est de mettre en œuvre plusieurs stratégies pour l'exploration des GMD :

- Persistance partielle de la carte mentale sur tout le GMD : le cerveau humain a besoin de repères pour naviguer facilement d'une image à une autre. Des éléments dont le comportement est stable sur l'ensemble des états du GMD sont identifiés puis leur position est rendue fixe dans tout le graphe (dans la suite ils sont appelés *éléments constants*).
- Comparaison et persistance locale : le zoom sur un état des données *complètes* (c'est-à-dire non *réduites*) permet d'observer localement – c'est-à-dire dans le contexte dimensionnel – les changements.
- Alternance de données réduites et complètes : l'utilisateur peut naviguer entre des vues d'ensemble (réduction dimensionnelle) et des vues locales (zoom sur une partie des données, par exemple un ou plusieurs états).

La stratégie principale, à savoir la persistance globale de la carte mentale, donne son nom à la méthode d'exploration, *OCL* pour *Overview Constraint Layout*, puisque l'idée est de permettre à l'utilisateur de garder une vue d'ensemble sur les données à tout moment de l'exploration grâce à l'application d'une contrainte dans la visualisation d'une partie des données, celles qui sont constantes.

6.1.2.2 Scénario d'exploration

Le scénario utilisateur de la méthode d'exploration OCL, constitué de deux grandes étapes, s'appuie sur le mantra de la visualisation de Shneiderman (1996) :

1. Exploration *de synthèse* : données réduites qui donnent des indications sur la façon dont il faut aborder les données complètes, puisqu'elles permettent d'identifier les éléments constants du GMD. La réduction peut s'effectuer sur une dimension ou sur la totalité des dimensions. Cette phase de l'exploration nécessite peu d'interactions. Le layout calculé à partir des éléments constants du GMD est proposé par défaut, mais un layout physique peut également être appliqué.

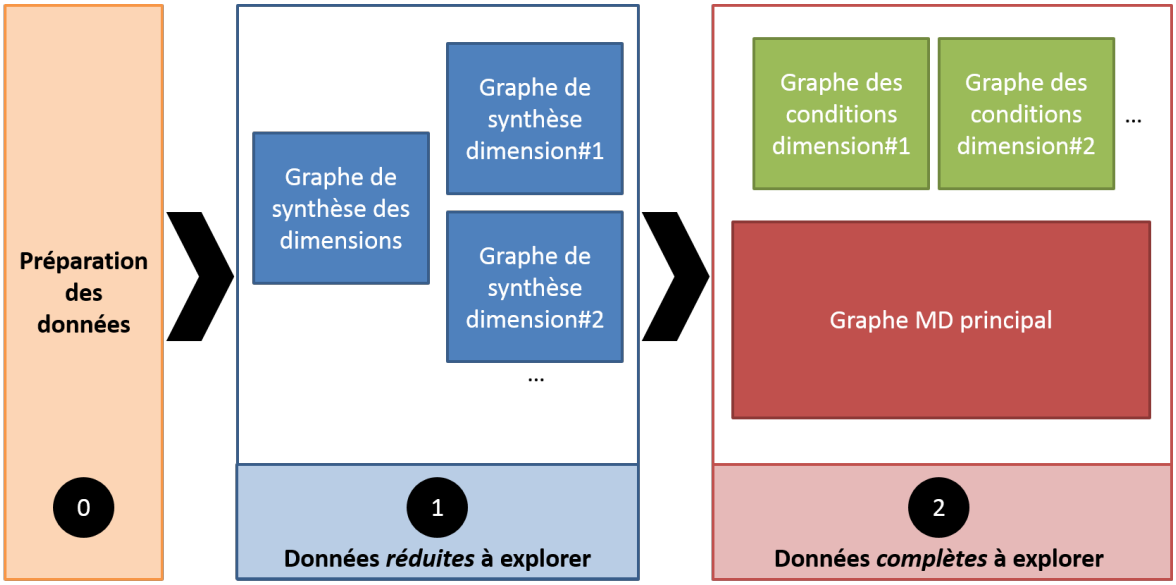


FIGURE 6.2 – Étapes du scénario d'exploration OCL, et graphes associés.

2. Exploration *en contexte* : données complètes qui ne peuvent être visualisées en une seule vue. L'interaction est indispensable à cette étape. L'utilisateur est aidé dans son exploration grâce à la préservation partielle de la carte mentale : les éléments stables sont fixés dans l'espace, ce qui permet de mettre en exergue les changements d'un état sur l'autre. Lors de cette étape, l'utilisateur peut utiliser les techniques classiques d'exploration locale comme filtrer, zoomer, comparer ou encore utiliser les vignettes multiples. Les changements ainsi que les éléments stables *en contexte* sont également mis en valeur grâce aux techniques de visualisation classiques (couleur, taille et visibilité des éléments). L'exploration s'étend naturellement à la visualisation des graphes de conditions, à l'aide des techniques classiques.

L'étape (1) est un pré-requis pour aborder (2), mais ensuite il est possible à tout moment de faire des aller-retours entre les deux étapes. Avant de pouvoir commencer l'exploration, les données doivent être préparées. L'enjeu principal est de déterminer les éléments constants qui vont participer à la construction du layout de préservation partielle de la carte mentale de l'utilisateur et au calcul du ou des graphes de synthèse.

L'espace total d'exploration est un ensemble de graphes : le graphe des données initiales à explorer (étape 2) et les graphes qui aident à l'interprétation des données (étape 1). La figure 6.2 illustre cet ensemble.

Aux deux étapes d'exploration de la méthode s'ajoute une étape de préparation des données pour mettre en place la persistance visuelle des éléments du graphe entre les vues avec lesquelles l'utilisateur va interagir.

6.1.3 Préparation des données

Pour rendre fluide et permettre une meilleure interaction de l'utilisateur avec les graphes à explorer, notamment via lune persistance partielle de la carte mentale, les données doivent être préparées. La figure 6.3 présente la chaîne minimale des opérations à effectuer pour préparer les données avant l'exploration. Cette chaîne est constituée de huit opérations :

1. Filtre sur le poids des arêtes : si le GMD principal à explorer est complet, il est nécessaire avant de calculer les éléments constants et les layouts de filtrer les arêtes, car la méthode OCL n'est pas valide sur un graphe complet.
2. Calcul des éléments du *graphe de synthèse* : il est obtenu d'après les valeurs associées aux éléments du graphe sur une dimension ou un ensemble des dimensions du graphe. Les premières étapes de détermination des layouts sont calculées sur le graphe de synthèse.
3. Calcul des *éléments constants* : les éléments qui présentent une stabilité sur l'ensemble des états du GMD sont identifiés, en fonction de paramètres.
4. Layout sur les *éléments constants actifs* : les éléments constants actifs [1] sont extraits du graphe de synthèse et un layout de positionnement est calculé.
5. Fixation des positions des éléments constants actifs : les positions des éléments actifs sont extraites du layout obtenu à l'étape précédente et fixées dans leurs attributs.
6. Layout sur tous les éléments : le calcul du layout est lancé sur l'ensemble des éléments du graphe de synthèse, avec les positions des éléments constants actifs maintenues fixes.
7. Fixation des positions des *éléments constants inactifs* : les positions des éléments inactifs sont extraites du layout obtenu à l'étape précédente et fixées dans leurs attributs.
8. Layout sur chaque état du GMD : le layout final de chaque état est calculé. C'est dans ce layout que l'utilisateur va naviguer pour préserver sa carte mentale.

Les opérations 4 à 8 constituent l'exécution d'un *layout à contraintes* (appelé *LàC* dans la suite du manuscrit). Le GMD de sortie est enrichi des données de comparaison en contexte sur demande de l'utilisateur, au fur et à mesure de l'exploration, pendant l'étape 2 de la méthode OCL.

6.2 Visualisation d'états *en contexte*

Un état *en contexte* est un état *ordonné* au sein d'une dimension ou de plusieurs dimensions. Cela implique que ses états précédents et suivants sont connus sur la ou les dimensions (GMD offline). Une façon de représenter un état dans son contexte au sein de plusieurs dimensions est proposé en figure 6.4.

L'objectif est de pouvoir visualiser comment la situation des éléments d'un état évolue par rapports aux états voisins immédiats. La construction de cette visualisation revient à comparer

1. la définition des éléments actifs et inactifs est donnée dans la section 6.2.2

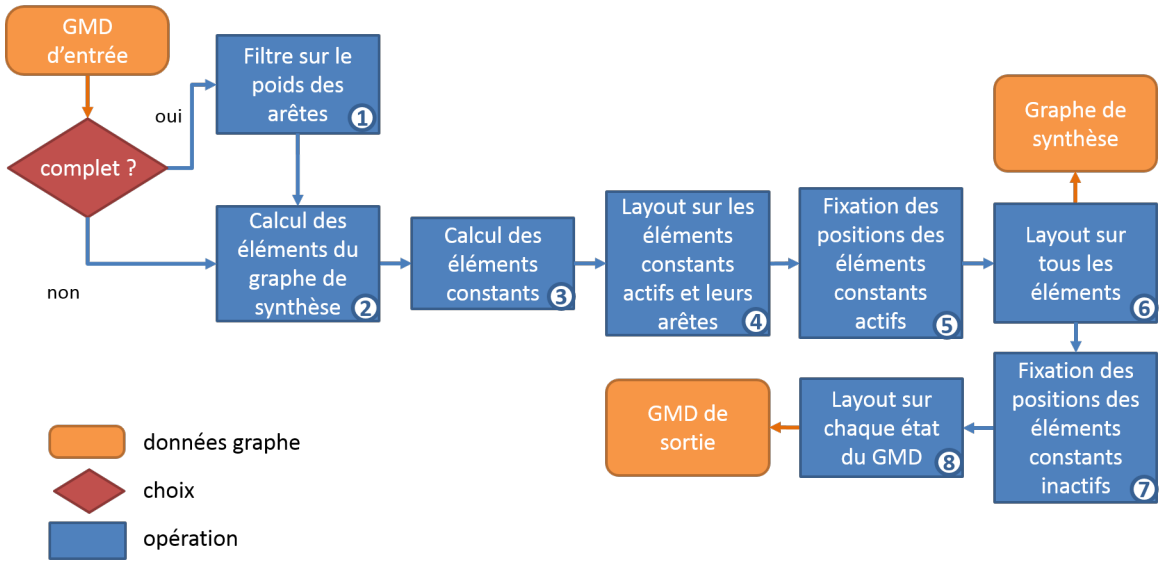


FIGURE 6.3 – Chaîne des opérations de préparation des données pour l'exploration OCL

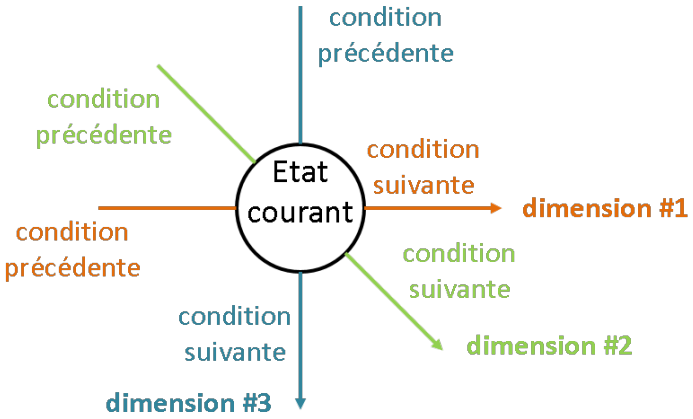


FIGURE 6.4 – Représentation d'un état dans son contexte, exemple avec trois dimensions.

l'état avec ses états voisins et de transcrire visuellement le résultat de cette comparaison. La comparaison de deux configurations de graphe s'effectue selon axes : ce qu'elles ont de commun et de différent.

Pour toutes les méthodes présentées dans cette section, les états du graphe sont considérés en contexte. Un seul état est visualisé à la fois : l'état est étudié dans son contexte immédiat. Dans cette section les attributs et les propriétés topologiques des éléments d'un graphe ne sont pas différenciés et sont désignés par le terme *attribut*.

6.2.1 Mise en exergue des changements

Les changements pour un contexte donné sont causés au niveau local (élément) par deux types d'événements :

– L'apparition d'un élément ou l'augmentation de la valeur d'un de ses attributs vis à vis de l'état précédant ou suivant.
– La disparition d'un élément ou la diminution de la valeur d'un de ses attributs vis à vis de l'état précédant ou suivant.

Par conséquent, nous pouvons identifier trois types de changement local pour un contexte :

– *Croissance* : décrit une apparition ou une augmentation vis à vis de l'état précédent ou suivant.
– *Décroissance* : décrit une disparition ou une diminution vis à vis de l'état précédent ou suivant.
– *Inflexion* : décrit une apparition et une disparition ou une augmentation et une diminution vis à vis des états précédants et / ou suivants.

Un moyen de visualiser ces changements locaux est présenté dans cette sous-section. L'approche est la même, que les événements aient lieu sur les éléments ou leurs attributs.

Dans un second temps, nous nous intéressons à l'impact de chaque événement local sur les éléments du graphe en mesurant la quantité de changements de relations pour chaque nœud, grâce à la mesure de centralité du changement de Federico *et al.* (2012) que nous adaptons à notre problème.

6.2.1.1 Naissance et mort des éléments du graphe

L'usage de la couleur pour mettre en exergue des changements est courant dans la bibliographie, notamment pour les animations (Bach *et al.*, 2014a). Pour indiquer visuellement les types de changements, nous choisissons d'utiliser un triplet de couleur car la propriété à visualiser est binaire. Pour un contexte à une dimension, on donne :

Rouge : disparition à l'état suivant
Vert : apparition par rapport à l'état précédent
Bleu : apparition par rapport à l'état précédent et disparition à l'état suivant

Par opposition, l'absence de couleur (noir) indique l'absence de changement.

Dans le cadre d'un contexte à plusieurs dimensions, la visualisation des changements peut être abordée de deux façons :

– Selon le contexte : c'est l'état en contexte qui est visualisé. Les changements identifiés
– En tant que synthèse du contexte : c'est un état synthèse qui contient tous les nœuds et toutes les arêtes possibles dans tous les états du contexte qui est visualisé. Les changements sont identifiés par rapport à l'ordre des états et le référentiel de comparaison reste l'état courant du contexte. Cette méthode de mise en exergue des changements permet en particulier d'identifier s'il existe des éléments dans le graphe aux comportements extrêmes : soit instables soit stables par rapport aux comportements des autres éléments du graphe.

Pour illustrer ces concepts, nous reprenons l'exemple de GMD du chapitre 5 et étudions l'état (002,t1) du graphe dans son contexte. La représentation de cet état selon l'ordre des

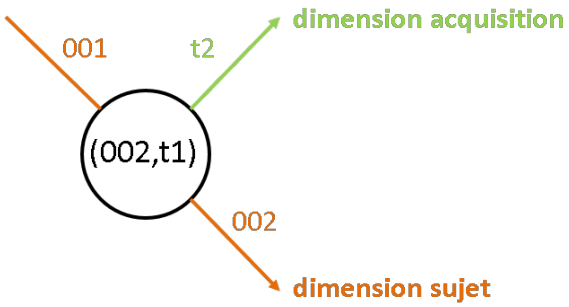


FIGURE 6.5 – Représentation de l'état (002,t1) dans son contexte à deux dimensions.

deux dimensions du GMD est donné dans la figure 6.5 : l'état est précédé de l'état (001,t1) et suivi de l'état (003,t1) selon la dimension *sujet*, il est également suivi de l'état (002,t2) selon la dimension *acquisition*.

La figure 6.6 présente les résultats de la visualisation en contexte de l'état (002,t1) :

a) Selon la dimension *sujet* : les connexions entre les nœuds 1 et 2, ainsi que 2 et 3 apparaissent à l'état (002,t1). Globalement il y a croissance du nombre d'arêtes sur la dimension.

b) Selon la dimension *sujet*, avec un layout fixe et la visualisation en contexte est également donnée pour les états précédent (001,t1) et suivant (003,t1). Nous constatons qu'il est plus aisé de visualiser les différences quand le layout est fixe d'un état à un autre.

c) Selon la dimension *acquisition* : le nœud 1 disparait à l'état suivant de la dimension, et par conséquent ses arêtes disparaissent aussi.

d) Selon les dimensions *sujet* et *sujet* (soit suivant le contexte total) : la somme binaire des changements sur le contexte montrent que le nœud 1 et une partie de ses arêtes a une tendance à disparaitre sur le contexte. La connexion entre les nœuds 1 et 2 apparait et disparait sur le contexte, tandis que la connexion entre les nœuds 2 et 3 apparait sur le contexte.

e) Synthèse sur l'ensemble des éléments du contexte : cette visualisation met en exergue que toutes les connexions subissent des changements sur le contexte, sauf l'arête entre les nœuds 2 et 4 qui est stable sur tout le contexte. Par ailleurs, le nœud 1 est le seul à présenter un changement (disparition).

Les GMD ne sont généralement pas des graphes de petite taille, aussi il convient de comprendre son application à des graphes plus grands. Un exemple de comparaison en contexte sur l'existence des nœuds et des arêtes d'un graphe de connectivité fonctionnelle d'un cerveau humain segmenté en 384 régions selon une dimension *sujet* est donné dans la figure 6.7. L'existence des nœuds est régie par l'absence de connexions à ce nœud dans l'état, en réalité les nœuds ne disparaissent pas puisqu'ils modélisent les régions du cerveau.

Nous observons qu'il est difficile d'interpréter individuellement les éléments du graphe quand leur nombre est trop important. Cependant cela permet d'identifier une tendance sur le contexte :

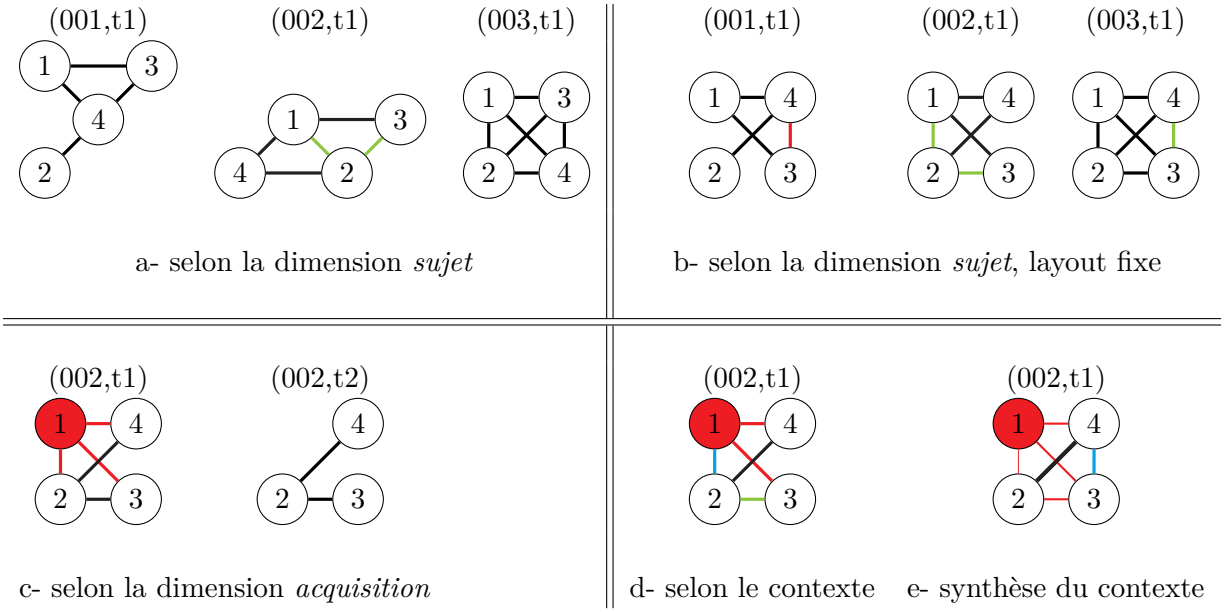


FIGURE 6.6 – Comparaison en contexte de l'état (002,t1) sur l'existence des nœuds et des arêtes. a et c : les états (001,t1) et (003,t3) (respectivement (002,t2)) sont donnés à titre de référence pour la compréhension du fonctionnement de la comparaison, mais normalement seul l'état sur lequel porte le contexte est visualisé. b : la comparaison en contexte des états (001,t1) et (003,t3) est donnée pour illustrer l'absence de symétrie de la comparaison. b, c, d et e : un layout fixe est appliqué sur les nœuds pour aider à la comparaison visuelle des différents cas.

la proportion de changements et le type de changements renseignent sur la stabilité de l'état vis à vis de ses états précédent et suivant. L'état de la figure 6.7.a montre peu de changements (la plupart de ses arêtes restent noires), tandis que l'état 6.7.b présente de nombreux changements sur ses arêtes et surtout des changements d'inflexion (arêtes bleues), ce qui signifie que ces arêtes apparaissent et disparaissent à cet état.

6.2.1.2 Évolution d'un attribut

Le principe de comparaison en contexte pour l'évolution d'un attribut est similaire à l'étude de l'évolution de l'existence d'un élément, hormis que le critère de détermination du triplet visuel est qualitatif :

Rouge : baisse de la valeur par rapport à l'état précédent ou à l'état suivant

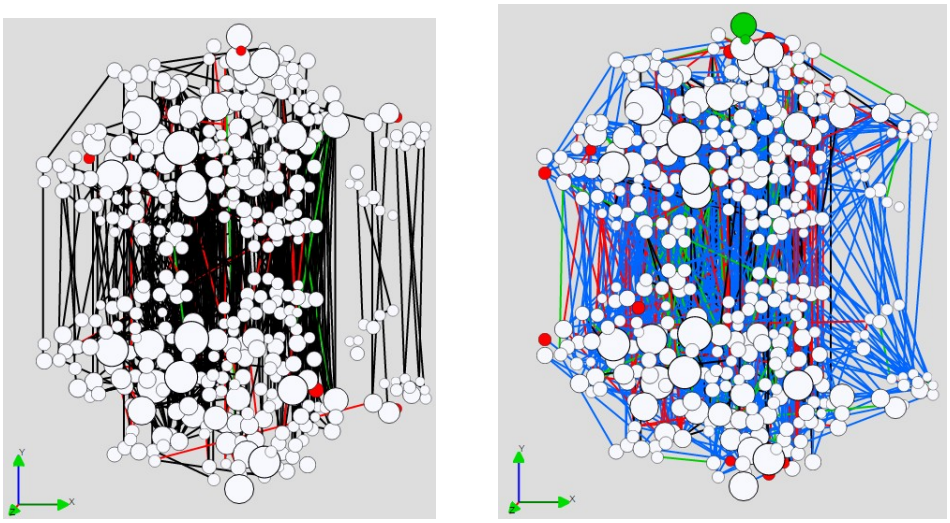
Vert : augmentation de la valeur par rapport à l'état précédent ou à l'état suivant

Bleu : baisse et augmentation par rapport aux états précédents et suivants (point d'inflexion)

Aux nœuds du GMD introduit dans le chapitre 5, nous relevons le degré de chaque nœud (propriété topologique) et lui associons comme un attribut. Le relevé des degrés pour chaque nœud et pour chaque état est présenté dans le tableau 6.1.

La figure 6.8 présente les résultats de la comparaison pour l'état (002,t1) :

a) Selon la dimension *sujet* : les nœuds 1, 2 et 3 gagnent des connexions dans le contexte de la dimension sujet, tandis que le nœud 4 perd puis gagne des connexions dans le même



a- Configuration "stable" b- Configuration d'"inflexion"

FIGURE 6.7 – Comparaison en contexte des variations des éléments d'un graphe de connectivité fonctionnelle selon une dimension *sujets*. Le graphe est composé de 384 nœuds qui sont mis en forme par une projection anatomique 2D ; les arêtes du graphe complet sont filtrées à 95%, soit 3677 arêtes. a et b sont obtenus pour deux états différents.

nœud / état	**(001,t1)**	**(001, t2)**	**(002,t1)**	**(002, t2)**	**(003,t1)**	**(003, t2)**
1	2	1	3	0	3	2
2	1	1	3	2	3	2
3	2	1	2	1	3	2
4	3	3	2	1	3	2

TABLE 6.1 – Degré des nœuds pour chaque configuration du GMD

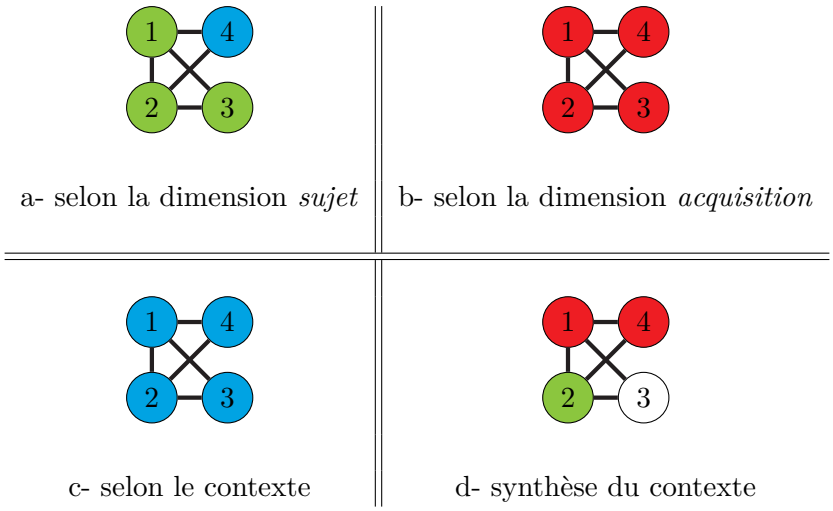


FIGURE 6.8 – Comparaison en contexte de l'état (002,t1) sur les nœuds (attribut degré des nœuds)

contexte.

b) Selon la dimension *acquisition* : les quatre nœuds perdent des connexions entre les états (002,t1) et (002,t2).

c) Selon le contexte global (dimensions *sujet* et *acquisition*) : les quatre nœuds présentent une inflexion dans le contexte, ils gagnent et perdent chacun des connexions.

d) Synthèse du contexte : la somme signée des changements est nulle sur le nœud 3 ; tandis que les nœuds 1 et 4 perdent globalement des connexions sur le contexte, le nœud 2 en gagne.

Si l'ordre des conditions de la dimension sujet était inversé, les résultats seraient différents : la comparaison n'est pas symétrique puisque nous ne visualisons pas l'ensemble des éléments du graphe présents dans le contexte, mais seulement les éléments appartenant à l'état courant. Il convient d'être conscient lors de l'usage de la comparaison en contexte des changements que l'ordre des conditions est central pour la visualisation.

6.2.1.3 Mesure du changement

Pour des graphes de taille moyenne, il devient difficile d'analyser les changements pour chaque éléments à l'aide d'une comparaison binaire. La mesure du *Change Centrality* (CC) de Federico *et al.* (2012) présentée dans le chapitre 3 permet de calculer une valeur normalisée pour chaque nœud qui indique la quantité de changements à son voisinage. Cette mesure a été conçue pour des graphes simples, sans cycles.

La figure 6.9 présente une extension de la figure 1 de l'article de Federico *et al.* (2012) : une arête est ajoutée entre les nœuds A et D, aux deux états t1 et t2, ce qui créé un cycle A-B-D à l'état 1.

Calculs du Change Centrality pour la figure 6.9

$CC_{t_1,t_2}(A) = 0.125(= \frac{1}{2} * \frac{|0|}{|A|} + \frac{1}{4} * \frac{|0|}{|B+D|} + \frac{1}{8} * \frac{|B+E|}{|C+D+B+E|} + \frac{1}{16} * \frac{|E+C|}{|E+C|})$

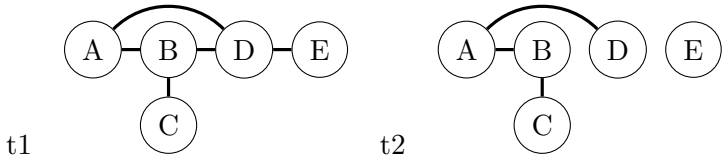


FIGURE 6.9 – Exemple 1 d'un réseau dynamique à deux états temporels successifs (t1, t2), à partir de la figure proposée par Federico *et al.* (2012).

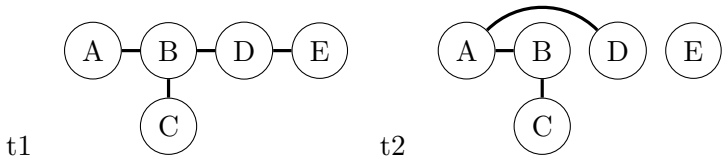


FIGURE 6.10 – Exemple 2 d'un réseau dynamique à deux états temporels successifs (t1, t2), à partir de la figure proposée par Federico *et al.* (2012).

$CC_{t_1,t_2}(B) = 0.166(= \frac{1}{2} * \frac{|0|}{|B|} + \frac{1}{4} * \frac{|D|}{|A+C+D|} + \frac{1}{8} * \frac{|E+A|}{|E+A+D|})$
$CC_{t_1,t_2}(C) = 0.135(= \frac{1}{2} * \frac{|0|}{|C|} + \frac{1}{4} * \frac{|0|}{|B|} + \frac{1}{8} * \frac{|D|}{|A+D|} + \frac{1}{16} * \frac{|A+E|}{|A+D+E|} + \frac{1}{32} * \frac{|E|}{|E|})$
$CC_{t_1,t_2}(D) = 0.25(= \frac{1}{2} * \frac{|0|}{|D|} + \frac{1}{4} * \frac{|B+E|}{|A+B+E|} + \frac{1}{8} * \frac{|A+C|}{|A+B+C|} + \frac{1}{16} * \frac{|0|}{|C|})$
$CC_{t_1,t_2}(E) = 0.469(= \frac{1}{2} * \frac{|0|}{|E|} + \frac{1}{4} * \frac{|D|}{|D|} + \frac{1}{8} * \frac{|A+B|}{|A+B|} + \frac{1}{16} * \frac{|A+C|}{|A+C|} + \frac{1}{32} * \frac{|C|}{|C|})$

Lorsqu'il y a un cycle dans le graphe, une même arête est prise en compte plusieurs fois dans le calcul. Cela a un impact sur la valeur de CC : la quantité de changement diminue globalement car l'union des nœuds graphe à la distance i sur les deux états a tendance à être plus grande.

L'exemple de la figure 6.9 présente une extension de la figure 1 de l'article de Federico *et al.* (2012) : une arête est ajoutée entre les nœuds A et D, à l'état t2 uniquement. Le graphe ne présente pas de cycle, et les nœuds A et D sont reliés à une distance de 2 à l'état t1 et de 1 à l'état t2. On obtient, pour le calcul du CC :
$CC_{t_1,t_2}(A) = 0.125(= \frac{1}{2} * \frac{|0|}{|A|} + \frac{1}{4} * \frac{|0|}{|B+D|} + \frac{1}{8} * \frac{|D|}{|C+D|} + \frac{1}{16} * \frac{|E|}{|E|})$
Le nœud D est pris en compte deux fois : pour son voisinage direct avec le nœud A et pour l'impact qu'il a au voisinage du nœud B.

Sur des graphes de moyenne à grande taille contenant des cycles, ce qui nous intéresse est de connaître les changements intervenant sur le plus court chemin entre deux nœuds, soit la distance géodésique. Par conséquent nous proposons de modifier l'algorithme proposé par Federico *et al.* (2012) pour que chaque nœud du graphe ne soit pris en compte que pour la distance géodésique qui le relie au nœud sur lequel porte la mesure. Cette modification de l'algorithme va permettre d'éviter les effets de réduction du CC mentionné plus haut et également d'augmenter les performances de l'algorithme, ce qui est intéressant pour les grands graphes.

Par ailleurs, le CC prend en compte la disparition des nœuds qui impacte pour moitié le

résultat : si aucun nœud ne disparait, le CC ne peut pas valoir plus de 0,5. Cela est dû à la suite géométrique de la formule, $\frac{1}{2^{n+1}}$ avec n la distance.

Pour le cas d'un graphe sans disparition de nœuds, nous proposons de modifier la suite géométrique pour obtenir une valeur normalisée entre 0 et 1, au lieu de 0 et 0,5. La distance $n = 0$ est exclue de la somme et les coefficients affectés à chaque distance multipliés par $\frac{1}{2}$. En reprenant l'équation donnée en 3.6, nous obtenons :

$$CC_{t_1,t_2}(i) = \sum_{n=1}^{e_i} \frac{1}{2^n} r^n_{t_1,t_2}(i) \tag{6.1}$$

Remarques sur la mesure de CC :

- La mesure est symétrique : $CC_{t_1,t_2}(i) = CC_{t_2,t_1}(i)$.
- L'impact d'un changement sur la valeur totale est fonction de la distance au nœud sur lequel porte la mesure et le nombre de nœuds à cette distance. La valeur de CC a donc davantage de légitimité à être calculée sur des états présentant un même nombre de nœuds et d'arêtes.

Le CC est une mesure sur les nœuds de comparaison entre deux graphes. Pour l'appliquer à la visualisation en contexte, il faut soit obtenir une mesure globale sur l'ensemble des états du contexte, soit visualiser en même temps toutes les mesures. Prenons pour exemple un graphe à une dimension : sur un même nœud, nous voulons observer la valeur de la mesure par rapport aux états $e - 1$ et $e + 1$:

- Sans modifier la forme du nœud, il faut combiner les valeurs CC_{next} et CC_{prev}, par exemple les additionner. La valeur résultante n'est plus normalisée, mais il est toujours possible. L'inconvénient est de ne pas indiquer de quel état provient la plus grande quantité de changements.
- En modifiant la forme du nœud, il devient possible de visualiser à la fois CC_{next} et CC_{prev} : une moitié de la forme supporte CC_{next}, et l'autre moitié CC_{prev}. Dans la représentation node-link classique, un nœud est matérialisé par un disque, nous partons donc sur une base simple de deux disques.

Un exemple de graphe sur lequel sont visualisées les valeurs du change centrality avec et sans modification de la forme du nœud est présenté dans la figure 6.11.

6.2.2 Mise en exergue des éléments communs

Les éléments qui ne présentent pas de changements selon un critère sur le contexte sont appelés *éléments communs*. Un nœud peut être commun à tous les états du contexte, tandis qu'un de ses attributs va présenter des variations sur ces mêmes états du GMD. Les éléments communs s'identifient en opposition à la présence de changements. Dans le cas de propriétés des éléments qui ne sont pas binaires, ce sont les éléments qui présentent des changements minimum qui sont recherchés.

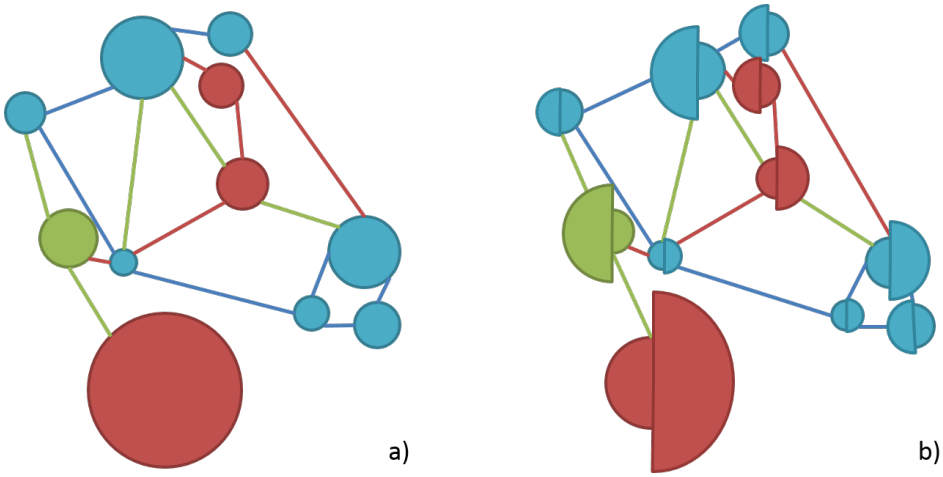


FIGURE 6.11 – État d'un graphe sur lequel ont été calculées des valeurs CC_{prev} et CC_{next} par rapport aux états précédent et suivant. a) CC_{next} et CC_{prev} sont additionnées pour définir le rayon des nœuds. b) Les nœuds sont représentés par deux demi-disques dont les rayons sont définis respectivement par CC_{prev} (gauche) et CC_{next} (droite). [graphe réalisé à la main : au moment de la rédaction de ce manuscrit, la fonctionnalité n'est pas encore disponible sur l'interface SwoViewer]

Nous décrivons dans cette sous-section la recherche d'éléments communs appliquée à deux types de critères : la stabilité topologique individuelle d'un élément et l'appartenance d'un élément à une structure commune.

6.2.2.1 Identification des éléments stables

Dans cette sous-section sont présentés le vocabulaire et les critères de définition associés à l'identification des éléments stables, d'un point de vue topologique, d'un GMD sur son contexte. Des critères de stabilité basés sur les attributs des éléments pourraient être définis de façon similaire.

Nœuds actifs et inactifs pour un état : un nœud *actif* possède un nombre de connexions élevé avec les autres nœuds du graphe.

Un nœud *inactif* possède un faible nombre de connexions avec les nœuds du graphe, voire aucune connexion.

Nœuds stables sur une dimension : un nœud *stable* présente peu de variations dans ses connexions. La quantité de connexions est donnée par la moyenne des degrés du nœuds sur l'ensemble des états de la dimension. Cette variation peut être mesurée par plusieurs indicateurs – qu'on appellera *indicateurs de stabilité* –, selon ce que l'on veut mettre en évidence :

- Écart-type sur la moyenne des degrés du nœud pour l'ensemble des états.
- Somme des mesures de change centrality (implique que la ou les dimensions du contexte soient ordonnées).

Un nœud actif est constant si sa somme des mesures de change centrality est faible.

		état 1	état 2	état 3	état 4	moyenne	écart-type
$e1$	non-filtré	0.92	0.31	0.9	0.99	0.78	0.273
	filtré ≥ 0.9	0.92	-	0.9	0.99	0.94	0.038
$e2$	non-filtré	0.92	0.93	0.88	0.8	0.8825	0.051
	filtré ≥ 0.9	0.92	0.93	-	-	0.925	0.005

TABLE 6.2 – Poids des arêtes $e1$ et $e2$ en fonction de quatre états, accompagnés des valeurs de la moyenne et de l'écart type avec et sans filtre à *poids* ≥ 0.9.

Un nœud inactif est constant si l'écart-type liée à la moyenne des degrés du nœud est faible.

Il est possible de déterminer des classes de nœuds stables pour mieux observer. On appelle l'identifiant de cette classe *degré de stabilité des nœuds*.

Arêtes stables : une arête est *stable* à la fois si son existence et son poids sont stables sur l'ensemble de son existence.

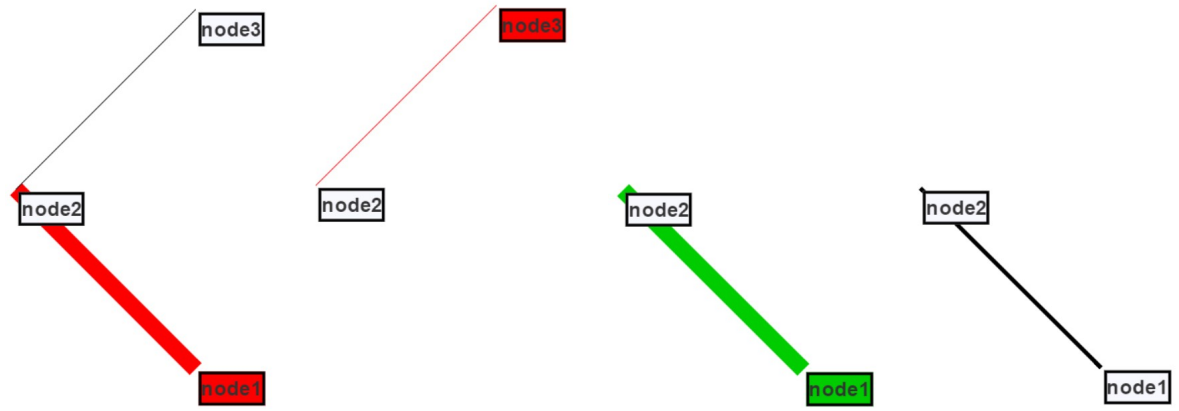
Tendances des arêtes pondérées Il peut être intéressant d'étudier cette tendance à la fois sur le graphe filtré que sur le graphe non filtré en pondération. Un filtre sur la pondération (des arêtes ou des nœuds) introduit potentiellement un biais : si la limite est fixée à 1,05 inclus, que peut-on dire de l'élément dont le poids vaut 1,049 ? Ainsi, si un graphe possède quatre états selon une dimension et que l'on considère les arêtes $e1$ et $e2$, $e1$ est existe dans trois états sur quatre, tandis que $e2$ existe dans deux états sur quatre. Peut-on en déduire que $e1$ est plus stable que $e2$? A première vue oui, mais si on s'intéresse aux valeurs pondérées non filtrées, il est possible de s'apercevoir que $e2$ est plus stable que $e1$.

La table 6.2 présente en exemple les poids de deux arêtes $e1$ et $e2$ sur quatre états. Après l'application du filtre, en regardant uniquement l'existence sur les arêtes, c'est $e1$ qui est la plus stable. Si on s'intéresse à la stabilité liée au poids sur le graphe non-filtré, c'est alors l'arête $e2$ qui apparait la plus stable.

Si on étudie dans son contexte l'état 2 sur le graphe filtré, et qu'on visualise la naissance et la mort des arêtes, $e1$ et $e2$ apparaissent toutes les deux en rouge. Pour certaines analyses, ce constat est suffisant. Pour des explorations plus fines, il devient indispensable d'indiquer que l'arête disparait pour une raison forte – la valeur du poids est très éloignée de la valeur du filtre – ou faible – la valeur du poids est très proche de la valeur du filtre. Un moyen simple de visualiser la quantité d'accroissement ou de décroissance en valeur absolue du poids de l'arête est de jouer sur l'épaisseur de l'arête. Un exemple d'une telle visualisation est donné en figure 6.12, on y voit que l'arête $e1$ disparait à l'état 2 suite à une forte décroissance de son poids, comparativement à l'arête $e2$ qui disparait à l'état 3 suite à une décroissance faible de son poids.

6.2.2.2 Identification des groupes communs

La difficulté d'identifier des groupes de nœuds communs dans un contexte survient quand le nombre et la taille des groupes varient entre les états du contexte. Comparer deux groupes



State 1 State 2 State 3 State 4

FIGURE 6.12 – Visualisation en contexte d'un graphe à trois nœuds $\{node1; node2; node3; node4\}$ et deux arêtes $\{e1 : node1 - node2; e2 : node2 - node3\}$ en fonction de quatre états (données issues de la table 6.2). La naissance et la mort des éléments du graphe sont mis en exergue par la couleur (technique présentée dans la sous-section 6.2.1.1). La variation du poids en contexte de l'arête est représenté par sa taille.

revient alors à calculer le nombre de nœuds du plus petit groupe effectivement inclus dans le plus grand groupe, par rapport à la taille du plus petit groupe. La mesure ainsi obtenue est exprimée en pourcentage. Le mécanisme est inchangé quand les deux groupes sont de taille identique : le petit groupe et le grand groupe sont déterminés arbitrairement.

La comparaison de deux groupes est booléenne, selon une seuil de similitude en pourcentage donné par le paramètre *perSimi*. Pour que les deux groupes soient comparables, la taille du petit groupe ne doit pas être trop faible par rapport à la taille du grand groupe. Cette proportion est définie par le paramètre *perSize* en entrée de l'algorithme. L'algorithme de comparaison de deux groupes est présenté en pseudo-code dans le bloc 1 de l'annexe J.

Pour étendre cette comparaison simple au contexte d'un état, il s'agit de chercher les nœuds communs d'un groupe pour chaque état du contexte. L'ensemble de ces nœuds pour un groupe donné est appelé *motif*. L'algorithme 2 de l'annexe J présente comment obtenir les motifs d'un contexte.

L'instabilité des nœuds est mesurée à partir des motifs identifiés. Elle est obtenue en faisant la somme des écarts du nombre de nœuds groupe-motif pour chaque état du contexte et de état courant, divisé par la somme des nœuds du groupe pour chaque état. Pour nb_m : nombre de nœuds dans le motif, nb_i : nombre de nœuds du groupe de l'état i du contexte, nb_c : nombre de nœuds dans le groupe de l'état courant :

$$instability = \frac{(nb_c - nb_m) + (nb_0 - nb_m) + ... + (nb_{imax} - nb_m)}{nb_c + nb_0 + ... + nb_{imax}} \tag{6.2}$$

L'instabilité vaut :

– 0 quand il y a stabilité totale : le groupe est constitué exactement des mêmes nœuds à

tous les états du contexte.
- 1 quand il y a instabilité totale : aucun nœud du groupe est présent dans tous les états du contexte. Par exemple pour quatre groupes suivants : état1={1,2,4}, état1={1,2,3}, état1={2,3,4} et état1={1,3,4} les groupes sont identifiés communs pour $perSimi = 60\%$, et pourtant aucun nœud du groupe n'est commun aux quatre états : le motif est vide et l'instabilité vaut 1.
- $\frac{1}{2}$ quand il y a exactement autant de nœuds dans le motif que de nœuds en-dehors.

La mesure de l'instabilité du groupe permet de renseigner sur sa qualité, et par conséquent est un bon indicateur pour orienter le choix des critères de recherche des groupes communs et des motifs.

6.3 Analyse des tendances dimensionnelles

Dans cette section nous nous intéressons à l'étude des événements agrégés sur une ou plusieurs dimensions. Dans un premier temps, le calcul du *graphe de synthèse* est présenté : il contient les valeurs de synthèse des éléments du graphe sur toutes les dimensions et la propriété de constance des éléments. L'application du *Layout à Contraintes* (LàC) sur l'ensemble des états du GMD à partir de l'étape précédente de réduction des données est détaillée. Pour finir, la détermination et l'impact des paramètres de préparation des données sur l'exploration finale sont discutés.

6.3.1 Réduction des dimensions à un contexte

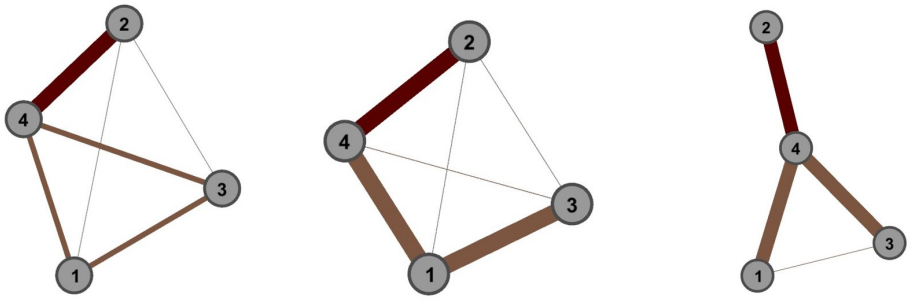
Sur des données de grande taille ou multidimensionnelles, il est impossible soit de visualiser tout à la fois, soit d'analyser la quantité de données pour en extraire une information pertinente. L'intérêt de réduire les données est de permettre à l'utilisateur de comprendre immédiatement les propriétés globales du GMD. Des informations sont perdues pendant le processus, mais la visibilité des données globales est renforcée.

6.3.1.1 Principe du graphe de synthèse

Le *graphe de synthèse* est la porte d'entrée de la méthode d'exploration OCL. Il propose une vue globale du comportement dimensionnel du GMD à l'utilisateur, et le familiarise avec les éléments constants du graphe dont la position va persister d'un état à l'autre lors de l'exploration des données complètes.

Dans le cas où le GMD possède plusieurs dimensions, autant de graphes de synthèse peuvent être calculés que de dimensions, plus un graphe de synthèse de l'ensemble des dimensions (contexte global). L'objectif est de fournir plusieurs niveaux de réduction qui vont permettre à l'utilisateur d'orienter son exploration sur une ou plusieurs dimensions en particulier.

Des graphes de synthèses calculés sur l'exemple du chapitre 5 sont présentés dans la figure 6.13. Sur le graphe de synthèse global, tous les éléments du graphe sont présents (quelque



a- Global b- Sujets-t1 c- Acquisitions-sujet001

FIGURE 6.13 – Graphes de synthèse de l'exemple simple de GMD du chapitre 5

	1-2	1-3	1-4	2-3	2-4	3-4
Global	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	1	$\frac{2}{3}$
Sujets-t1	$\frac{2}{3}$	1	1	$\frac{2}{3}$	1	$\frac{2}{3}$
Acquisitions-sujet001	0	$\frac{1}{2}$	1	0	1	1

TABLE 6.3 – Valeurs du poids de synthèse pour chaque arête du graphe de l'exemple simple de GMD du chapitre 5

que soit leur fréquence d'existence), tandis que le graphe de synthèse d'une dimension peut ne présenter qu'une partie des éléments du graphe total.

6.3.1.2 Calcul du graphe de synthèse

Le calcul du graphe de synthèse s'effectue sur tous les éléments du GMD. La fréquence d'existence est calculée pour chaque élément, et sur les attributs d'intérêt. Pour un graphe de synthèse d'une dimension, le même principe est appliquée sur une partie des états du GMD seulement.

Cela donne tout simplement, pour le calcul de la fréquence d'existence d'un nœud v :

$$\text{fréquence}(v) = \frac{nb\_existence(v)}{nb\_conditions} \tag{6.3}$$

Et pour le calcul d'un attribut de synthèse, par exemple sur le poids d'une arête e :

$$poids_{\text{synthèse}}(e) = \frac{\sum_{i=cond0}^{cond} p_i}{nb\_existence(e)} * \text{fréquence}(e) = \frac{\sum_{i=cond0}^{cond} p_i}{nb\_conditions} \tag{6.4}$$

Si le GMD ne présente pas de poids sur les arêtes, alors poids_synthèse(e) = fréquence(e). La table 6.3 présente les valeurs du poids de synthèse pour l'exemple simple de GMD introduit au chapitre 5.

L'exploration du graphe de synthèse peut se fait selon les techniques classiques de visualisation :

– Layout : physique ("natif" du graphe), LàC calculé pour l'exploration OCL ou layout de

type force calculé sur les éléments du graphe de synthèse.
- Esthétique des nœuds : taille et couleur sur la fréquence d'existence du nœud ou d'un attribut du nœud.
- Esthétique des arêtes : taille et couleur sur poids de synthèse, ainsi que taille et couleur sur la fréquence d'existence du nœud ou d'un attribut du nœud.

Sur la figure 6.13, les graphes de synthèse sont visualisés avec un layout de force simple et l'épaisseur des arêtes représente le poids de synthèse.

6.3.1.3 Algorithme d'identification des éléments constants

La méthode d'exploration OCL repose sur la préservation partielle. Les nœuds *constants* qui vont être fixés dans l'espace d'un état à l'autre sont les nœuds qui subissent peu de changements, c'est-à-dire que leur comportement est stable sur tous les états (voir 6.2.2). Suivant leur impact sur le graphe, les nœuds constants vont être identifiés avec des critères différents :

- **Nœuds actifs**
  - Caractère actif : intervalle itératif défini par un seuil (iThresholdAct)
  - Stabilité : la valeur de Change Centrality au nœud doit être inférieure à une valeur seuil (iThresholdCc) ou bien les écart-type des degrés du nœud et du Change Centrality doivent être inférieurs à des valeurs seuil (respectivement : taille de l'intervalle de stabilité et iThresholdSdCc)
- **Nœuds inactifs**
  - Caractère inactif : intervalle itératif défini par un seuil (iThresholdInact)
  - Stabilité : l'écart-type des degrés du nœud doit être inférieur à une valeur seuil (taille de l'intervalle de stabilité)

L'algorithme d'identification des nœuds constants, qui constitue l'opération 3 de préparation des données (voir figure 6.3), est présenté dans le bloc 3 de l'annexe J. Il est constitué de plusieurs étapes :

1. **Récupération des paramètres** (ligne 1)
   - iPerFixedNodes : pourcentage de nœuds à identifier
   - iPerNodesAct : pourcentage de nœuds actifs dans les nœuds à identifier
   - iThresholdAct : valeur seuil du degré des nœuds pour la détermination des nœuds actifs
   - iThresholdInact : valeur seuil du degré des nœuds pour la détermination des nœuds inactifs
   - iThresholdCc : valeur seuil liée à la mesure du Change Centrality
   - iThresholdSdCc : valeur seuil liée à l'écart-type des mesures de Change Centrality
2. **Initialisation des variables de travail** (lignes 2-6)
3. **Itérations pour atteindre le nombre de nœuds constants à identifier**
   - **Identification des nœuds constants actifs** (lignes 9-31)
     - Intervalle de définition "actif" pour l'itération (lignes 15-16)
     - Intervalle de stabilité (lignes 17-29)

- **Identification des nœuds constants inactifs** (lignes 32-46)
  - Intervalle de définition "inactif" pour l'itération (lignes 36-37)
  - Intervalle de stabilité (lignes 38-44)

Pour le cas où les critères ne permettraient pas la convergence des données, la boucle principale (lignes 7-47) présente une condition d'arrêt supplémentaire à mille itérations. Les valeurs par défaut des paramètres $iThresholdCc$ et $iThresholdSdCc$ sont fixées respectivement à $\frac{1}{2}$ et $\frac{1}{5}$.

6.3.2 Calcul du Layout à Contraintes (LàC)

Les opérations 4 à 8 de la préparation des données pour la méthode d'exploration OCL (voir figure 6.3) permettent de mettre en œuvre la stratégie de préservation partielle de la carte mentale. L'algorithme de détermination de la position des nœuds pour chaque état est appelé *LàC* (*Layout à Contraintes*).

6.3.2.1 Détermination de la position des nœuds constants

Les nœuds actifs sont les nœuds qui ont de l'impact sur le graphe, puisqu'ils sont fortement reliés aux autres nœuds. Ce sont eux qui vont structurer l'espace de visualisation le plus dense. Quant aux nœuds inactifs, ils doivent être fixés pour demeurer en périphérie de l'espace de visualisation dense.

La détermination de la position des nœuds constants s'effectue en deux temps pour préserver les caractéristiques des deux catégories de nœuds :

1. Positionnement des nœuds actifs : graphe de synthèse réduit aux nœuds actifs et leurs arêtes de synthèse.
2. Positionnement des nœuds inactifs : graphe de synthèse complet avec les positions des nœuds actifs fixées.

6.3.2.2 Choix du layout à utiliser

Le layout utilisé pour le calcul du LàC a un impact sur les tendances perçues par l'utilisateur (voir chapitre 3). Pour obtenir un résultat satisfaisant, il faut que le layout soit choisi en fonction des caractéristiques du graphe et du cas d'application, c'est-à-dire en fonction des tâches d'exploration à supporter en priorité. De façon très générale, dans le cadre de la méthode OCL, il faut *a minima* que le layout :

- Soit adapté à la taille du graphe,
- Prenne en compte le poids des arêtes,
- Autorise la fixation de nœuds pendant l'intégralité du temps d'exécution du layout.

Un ensemble de layouts peut être envisagé s'il n'apparait pas judicieux d'utiliser un unique layout aux étapes 4, 6 et 8 de la préparation des données.

6.3.3 Paramètres de la préparation des données

L'exécution de la préparation des données pour l'exploration OCL est conditionnée par un ensemble de paramètres qui agissent à différentes étapes. Quel est l'impact *a priori* de ces paramètres et comment les déterminer de façon optimale ?

6.3.3.1 Paramètres d'entrée

Sept paramètres sont répartis sur l'ensemble de la chaîne de préparation des données :

- **Opération 1**
  - ite : iThresholdEdges – Filtre sur les arêtes, influe sur la quantité de données à explorer.
- **Opération 3**
  - ipdf : iPerFixedNodes – pourcentage de nœuds à identifier, influe sur la quantité d'éléments persistants d'un état du graphe sur l'autre.
  - ipna : iPerNodesAct – pourcentage de nœuds actifs dans les nœuds à identifier, influe sur la fixation des nœuds inactifs.
  - ita : iThresholdAct – valeur seuil du degré des nœuds pour la détermination des nœuds actifs, influe sur le nombre de nœuds identifiés pour une itération.
  - iti : iThresholdInac – valeur seuil du degré des nœuds pour la détermination des nœuds inactifs, influe sur le nombre de nœuds identifiés pour une itération.
  - itCc : iThresholdCc – valeur seuil liée à la mesure du Change Centrality, influe sur le nombre de nœuds actifs identifiés pour une itération.
  - itSc : iThresholdSdCc – valeur seuil liée à l'écart-type des mesures de Change Centrality, influe sur le nombre de nœuds actifs identifiés pour une itération.
- **Opérations 4,6,8**
  - layout – le ou les layouts choisis constituent une variable de la préparation des données, sans compter les paramètres liés au layout lui même. Par exemple pour un layout de force classique, le coefficient de recouvrement des arêtes.

iThesholdAct, iThresholdInac, iThresholdCc et iThresholdSdCc ont un impact sur la convergence de la préparation des données.

6.3.3.2 Détermination des paramètres optimaux

Puisque les résultats visuels de la préservation partielle de la carte mentale peuvent être très différents suivant le jeu de paramètres utilisé, comment déterminer une combinaison de paramètres adaptés au GMD à visualiser ?

Pour aider à déterminer les paramètres optimaux de la préparation des données, nous proposons la démarche empirique suivante (illustrée dans la figure 6.14) :

1. Fixation de $iThresholdEdges$: des hypothèses métier permettent de déterminer un ou des seuils pertinents.
2. Fixation de $iThresholdAct$, $iThresoldInac$, $iThresholdCc$, $iThresholdSdCc$ et $iPerNodesAct$: analyse des caractéristiques topologiques du GMD filtré avec $iThresoldEdges$.

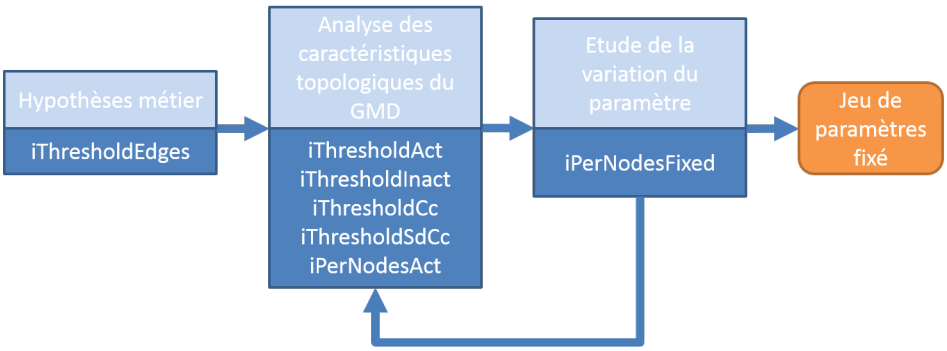


FIGURE 6.14 – Schéma de la détermination empirique des paramètres pour la méthode d'exploration OCL des GMD

3. Fixation de $iPerFixedNodes$: faire varier le pourcentage de nœuds à fixer comparer l'impact dans chaque calcul de la préservation de la carte mentale sur les performances de l'utilisateur.

Si suite à l'étape 3 les résultats ne sont pas satisfaisants dans leur variation, il convient de retourner à l'étape 2 pour recommencer l'analyse des caractéristiques topologiques. Par exemple un seuil $iThesholdAct$ ou $iThresoldInac$ qui fait récupérer trop de nœuds en une seule itération ne permettra pas de déterminer la bonne quantité de nœuds constants, et il sera impossible d'observer l'impact réel de la variation du nombre de nœuds fixes d'un état sur l'autre.

Conclusion du chapitre 6

Nous avons présenté dans ce chapitre la méthode OCL (Overview Constaint Layout), qui permet l'exploration visuelle et interactive des GMD. Elle repose sur la préservation partielle de la carte mentale de l'utilisateur et l'alternance de vues complètes et réduites des données. Avant l'exploration visuelle, les données sont préparées avec l'algorithme LàC : les nœuds constants à travers l'ensemble des états du graphe sont identifiés, puis leur position fixe est déterminée. Les paramètres du LàC, tels que les critères de constance d'un nœud et le nombre de nœuds à identifier, sont à déterminer de façon empirique en fonction des caractéristiques du GMD à explorer. Une fois les données préparées, elles sont approchées visuellement par l'utilisateur en deux étapes, d'abord une vue d'ensemble réduite des données, puis une exploration locale sur les données complètes. La visualisation *en contexte* permet d'identifier les changements dimensionnels au niveau de l'état local, c'est-à-dire vis à vis des états précédants et suivants.

La suite de notre thèse présente l'implémentation des propositions détaillées dans les chapitres 4, 5 et 6 pour l'exploration de données à la provenance complexe d'un laboratoire en neuroimagerie.

Chapitre 7

Application à l'exploration de données en neuroimagerie

Dans les trois chapitres précédents nous avons présenté nos propositions pour permettre l'exploration de relations complexes entre ensembles de données hétérogènes : le modèle de données BMI-LM qui rend possible la gestion et la réutilisation de données en neuroimagerie dans un système PLM (chapitre 4), les GMD pour représenter et analyser sous la forme de graphe les données multidimensionnelles dynamiques (chapitre 5), et la méthode OCL d'exploration visuelle des GMD (chapitre 6).

Dans ce chapitre, un scénario d'utilisation qui reprend les principales étapes d'une étude en neuroimagerie est présenté et appliqué à l'étude de la connectivité fonctionnelle cérébrale. Puis l'implémentation des propositions au sein du laboratoire de neuroimagerie fonctionnelle GIN, sur le jeu de données BIL&GIN, est détaillée : mise en place d'un système PLM et exploration multidimensionnelle de la connectivité fonctionnelle cérébrale humaine.

Sommaire

7.1 Contexte de l'implémentation

Dans cette section est défini le contexte dans lequel l'implémentation de nos propositions est réalisée. Le cas d'utilisation est présenté, suivi du jeu de données utilisé pour sa réalisation. Les infrastructures développées dans le cadre du projet BIOMIST et qui servent mettre en place le cas d'utilisation sont également détaillées.

7.1.1 Cas d'application

Le cas d'application correspond à une problématique courante durant l'étude de la connectivité fonctionnelle au repos : l'exploration des données sous forme de graphes de connectivité. Il couvre les phases d'une étude de recherche : depuis la création d'une étude à la consultation des données dérivées et leur publication. Il a été développé à partir de l'analyse de la bibliographie (chapitre 2) et des besoins identifiés dans le laboratoire GIN (chapitre 4 et annexe C).

Le cas d'utilisation est décrit de façon détaillée dans l'annexe K. Cette section présente les tâches utilisateurs à chaque phase d'une étude, puis la liste des étapes qui constituent le cas d'utilisation.

7.1.1.1 Tâches utilisateurs

Les utilisateurs réalisent des tâches de nature différentes à chaque phase d'une étude :

A. Création d'études
   - Créer la structure de l'étude et les règles d'accès
   - Renseigner les données associées à une étude
   - Associer un statut à l'étude
   - Consulter les données de l'étude

B. Acquisition de données
   - Importer des données acquises
   - Ajouter les données de définition d'une acquisition
   - Requêter les données brutes
   - Analyser la provenance de données brutes
   - Associer un statut à l'étude
   - Consulter les données acquises

C. Traitement de données
   - Lancer un traitement sur des données (d'un sujet ou de plusieurs sujets)
   - Ajouter les données de définition d'un traitement
   - Définir le groupe de sujets d'une analyse
   - Requêter les données dérivées
   - Analyser la provenance de données dérivées
   - Associer un statut au traitement
   - Consulter les données dérivées

D. Publication de données

– Indiquer la provenance
– Garder plusieurs versions de la publication
– Analyser la provenance de données publiées
– Associer un statut à la publication
– Consulter les données publiées

Certaines tâches sont communes à toutes ou plusieurs phases, telles que l'association d'un statut ou la consultation des données.

7.1.1.2 Étude de la connectivité fonctionnelle cérébrale

Le cas d'utilisation retenu est l'étude de la connectivité fonctionnelle cérébrale humaine en fonction des caractéristiques des sujets. Il est constitué de dix étapes :

1. Créer une étude
2. Importer des données brutes acquises lors d'examens d'imagerie et de psychologie
3. Normaliser les acquisitions d'imagerie (spatial et temporel)
4. Calculer les matrices d'adjacence de connectivité fonctionnelle
5. Créer les groupes à analyser
6. Calculer les matrices d'adjacence de connectivité fonctionnelle correspondant aux groupes
7. Créer le GMD de la connectivité fonctionnelle des groupes
8. Préparer l'exploration visuelle du GMD
9. Explorer le GMD avec la méthode OCL
10. Publier des résultats

Les étapes 1 et 2 génèrent des données brutes, les étapes 3 et 4 des données de traitements individuels, les étapes 5 à 8 des données de traitements multi-sujets, l'étape 9 l'exploration des données de connectivité fonctionnelle cérébrale, et l'étape 10 la valorisation des données. Les relations entre les données générées aux étapes 1 à 4 sont présentées dans la figure 7.1 ; les relations montrent notamment la chaîne de traitements nécessaire pour obtenir une matrice d'adjacence de connectivité fonctionnelle par sujet.

7.1.2 Données du cas-test

Le GIN est une unité de recherche multidisciplinaire rassemblant des chercheurs des domaines de l'instrumentation pour l'imagerie médicale, de la médecine nucléaire, du traitement du signal, de la psychiatrie et des neurosciences cognitives.

Si le GIN a été conscient très tôt de la nécessité d'une gestion intégrée de la provenance pour pouvoir explorer des données multimodales, il est toujours confronté à des limites pour effectuer facilement les traitements courants et réutiliser des données. La participation du GIN à l'étude i-Share implique la gestion à partir de fin 2015 de nouvelles acquisitions d'une large cohorte de sujets (plusieurs milliers) et donc de se doter d'un système adapté.

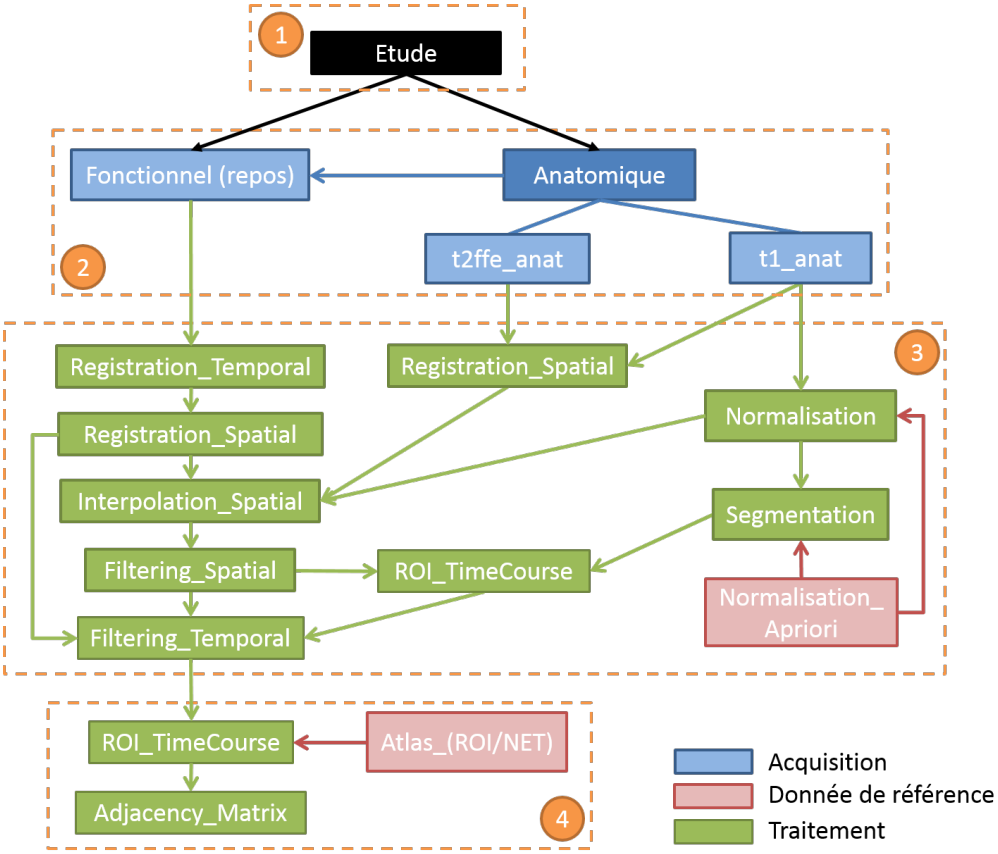


FIGURE 7.1 – Cas d'utilisation : étapes 1-4 sur les données individuelles. 1) Création d'une étude 2) Import d'acquisitions 3) Normalisation des acquisitions d'imagerie 4) Calcul des matrices de connectivité fonctionnelle

7.1.2.1 Le jeu de données BIL&GIN dans la base GINdb

Le jeu de données BIL&GIN – Brain Imaging of Lateralization studied by the Groupe d'Imagerie Neurofonctionnelle – a été créé pour étudier la spécialisation hémisphérique du cerveau des points de vue cognitif, comportemental, génétique et fonctionnel. Il contient des données associées à 453 sujets, dont 303 qui ont passé des examens d'imagerie fonctionnelle (IRMf) pour seize tâches différentes. La proportion de gauchers y est supérieure à la moyenne constatée dans la population (45%). Des données démographiques, psychologiques et génétiques ont également été recueillies. Les méthodes d'acquisition des données de BIL&GIN sont présentées dans Mazoyer *et al.* (2015).

Le jeu de données BIL&GIN est stocké dans la base de données GINdb. Cette base de données a été conçue pour permettre à des utilisateurs non-experts en informatique et en traitement du signal de mener facilement des analyses multimodales (Joliot *et al.*, 2010). Elle a été développée en MySQL : les métadonnées sont stockées dans des tables, et l'adresse des données sur un disque dur dédié est indiquée par un champ dans les tables.

7.1.2.2 Étude de la connectivité fonctionnelle cérébrale au repos

Parmi les 303 sujets du jeu de données BIL&GIN, 231 sujets ont eu des examens fonctionnels de repos qui ont été validés par un contrôle qualité et sont retenus pour le cas-test.

Pour chaque sujet, les données de travail sont :

- L'âge : calculé à partir des dates de naissance et de passage de l'examen d'imagerie.
- Le genre : {homme,femme} ({H,F})
- La latéralité : mesure binaire qui indique la prédominance de latéralité {gauche,droite} ({G,D}) chez le sujet
- La matrice de connectivité fonctionnelle, appelée matrice d'adjacence

Nous nous intéressons aux corrélations entre les caractéristiques des sujets et les relations entre les régions du cerveau à l'état de repos. Pour améliorer la validité statistique des résultats, nous groupons les sujets par caractéristique : l'âge, le genre ou de la latéralité. La répartition des sujets en fonction de ces trois caractéristiques est présentée dans le tableau 7.1. Les classes d'âge sont obtenues par tranches de cinq ans, avec le début de la première classe à 18 ans inclus ; les classes sont identifiées par le milieu de leur intervalle de définition, par exemple 20.5 pour la première classe dont l'intervalle vaut [18,23[.

Classe		**20.5**	**25.5**	**30.5**	**35.5**	**40.5**	**45.5**	**50.5**	**55.5**	*Total latéralité*	*Total genre*
Total		93	82	27	16	7	4	1	1	231	
H	**G**	28	21	2	3	0	1	0	1	56	114
	D	14	19	10	9	5	1	0	0	58	
F	**G**	28	20	4	1	1	0	1	0	55	117
	D	23	22	11	3	1	2	0	0	62	

TABLE 7.1 – Répartition des effectifs selon les caractéristiques des sujets

De façon générale, il est important que les effectifs d'une cohorte d'étude soient équilibrés, afin de présenter un effectif minimal dans chaque classe. Nous constatons dans le tableau que 78% de l'effectif se concentre sur les deux premières classes d'âge (20.5 et 25.5), et 94% sur les quatre premières (20.5, 25.5, 30.5 et 35.5). Les deux dernières classes (50.5 et 55.5) qui ne contiennent qu'un seul sujet ne peuvent pas être prises en compte pour l'étude de l'impact de l'âge. Les quatre classes de genre et de latéralité présentent globalement le même nombre de sujet chacune (écart maximal d'effectif de 11%).

Durant le processus de création des matrices d'adjacence, un atlas est utilisé. Il s'agit de l'atlas AICHA – Atlas of Intrinsic Connectivity of Homotopic Areas – qui est adapté à l'étude de la spécialisation hémisphérique du cerveau. Cet atlas a été calculé sur les acquisitions IRM au repos de 281 sujets, et est constitué de 192 paires de régions homotopiques. L'atlas AICHA et son processus d'obtention sont présentés dans Joliot *et al.* (2015).

7.1.3 Environnement du projet BIOMIST

Les objectifs du projet BIOMIST ont déjà été présentés dans la section 1.3.2.2. Dans cette section les aspects organisationnels et l'infrastructure technique sont développés.

7.1.3.1 Organisation et temporalité du projet

Le projet se déroule sur quarante-deux mois, de décembre 2013 à mai 2017. Les méthodes agiles (MA) permettent une définition continue des besoins et des technologies tout au long du projet. L'approche s'appuie sur des feedbacks et des ajustements permanents, ce qui est adapté à la recherche scientifique et aux démarches d'innovation. Les MA ont été créées dans le domaine du génie logiciel, et sont désormais adaptées au management de projets industriels, comme l'a montré le projet Wikispeed ou des projets de recherche tels que le projet européen FP6 EURACE (Marchesi *et al.*, 2007). Les MA permettent une meilleure appropriation du projet par les utilisateurs, car ces derniers sont d'avantage impliqués dans le processus en donnant régulièrement leur feedback.

Le projet est découpé en six lots de travail (WP) qui sont présentés dans la section 1.3.2.2.

- **WP1** : Gestion du projet et coordination
- **WP2** : PLM pour l'imagerie biomédicale
- **WP3** : Visualisation et comparaison de graphes
- **WP4** : Apports sémantiques pour la traçabilité et la réutilisation en recherche biomédicale
- **WP5** : Intégration et contrôle qualité
- **WP6** : Exploitation et dissémination des résultats

Le système PLM de gestion des données de neuroimagerie (WP2) est le squelette du projet, par conséquent la définition du modèle de données BMI-LM était un pré-requis à la suite des travaux. Les autres éléments sont développés de façon concourante, en satellite autour du système PLM.

Les travaux présentés dans ce manuscrit s'inscrivent principalement dans les WP 2 et 3.

7.1.3.2 Infrastructure du projet

L'infrastructure du projet est constituée de plusieurs briques qui vont interagir à la demande de l'utilisateur :

- **Interface de saisie des données** : saisie des données d'examens (par exemple examens psychologiques). Moyen de saisie : tablette avec un système d'exploitation Android.
- **Interface de visualisation de graphes** : requête, analyse de graphe, lancement de workflow et sauvegarde des graphes dans le PLM. Cette interface est basée sur l'interface SwoViewer et customisée pour les besoins du projet (requêtes et liens avec le PLM).
- **Serveur PLM** : gère les données et leur provenance grâce aux concepts PLM et au modèle de données BMI-LM (voir le chapitre 4). La solution PLM retenue est Teamcenter [1] version 10, avec une base Oracle. Cette solution propose à la fois un client riche et un client web.

1. Édité par Siemens.

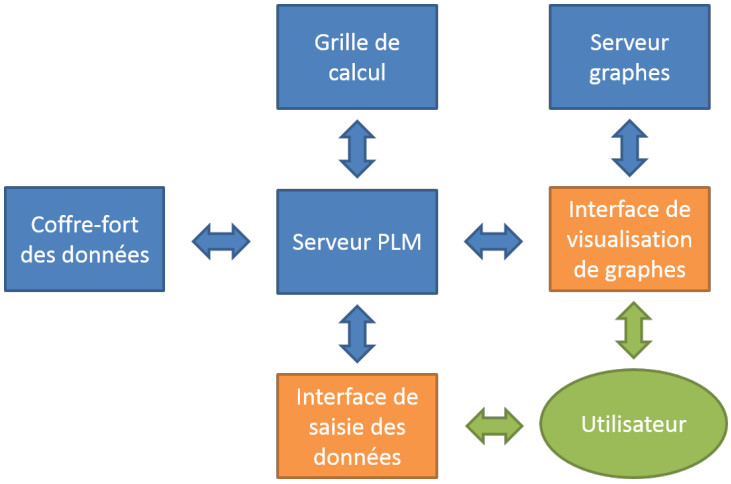


FIGURE 7.2 – Schéma théorique de l'infrastructure du projet BIOMIST représentant les relations entre les briques.

Le modèle de données et les préférences du système PLM sont customisées selon les besoins du projet ; ils sont présentés respectivement dans l'annexe D.

- **Coffre-fort des données** : espace de sauvegarde des fichiers, géré par la solution Teamcenter.
- **Grille de calcul** : élément préexistant, permet de calculer plus efficacement des calculs longs et coûteux. La grille de calcul Unix existant au GIN est utilisée.
- **Serveur graphes** : exécute des calculs de présentation des graphes pour les retourner dans l'interface de visualisation de graphes. Les algorithmes d'analyse et de présentation des graphes sont développés en Java.

Les relations entre les différentes briques du projet sont illustrées dans la figure 7.2.

7.2 Mise en place du système PLM

Afin de tester le modèle de données proposé dans le chapitre 4, un système PLM a été mis au place au GIN. Les caractéristiques de l'implémentation sont décrites dans la section 7.2.1, puis les résultats obtenus sont exposés dans la section 7.2.2. Pour finir, les retours d'expérience des chercheurs sur la mise en place du système PLM sont détaillés dans la section 7.2.3.

7.2.1 Description de l'implémentation

Comme dit précédemment, le système PLM choisi est Teamcenter, pour des raisons pratiques : d'une part les facilités de personnalisation et la modularité du logiciel et d'autre part un accès facile au logiciel lui-même et à du personnel formé à son administration. La version de travail est Teamcenter 10.3, la dernière à être sortie au moment du début de l'implémentation. Les auteurs des travaux de recherche présentés dans ce manuscrit n'ont aucun intérêt lié à la solution Teamcenter, ni dans tout autre logiciel utilisé pour l'implémentation. Il n'y a à notre

connaissance aucune raison de penser que les résultats présentés dans cette section auraient été différents en utilisant un autre système PLM du marché.

7.2.1.1 Implémentation du modèle de données

Dans Teamcenter les objets du modèle sont appelés Business Object – BO –, et sont organisés hiérarchiquement en fonction de leurs caractéristiques et de leurs fonctionnalités. Le module BMIDE – Business Modeler IDE – de Teamcenter permet de personnaliser le modèle de données par défaut de la base PLM. Les principaux éléments du modèle de données sont personnalisés dans le cadre du projet BIOMIST sont :

- Objets du modèle de données : un BO pour chaque objet de BMI-LM. Pour bénéficier de toutes les fonctionnalités associées aux articles dans le système PLM, les BO issus de BMI-LM héritent tous du BO article. BOs versionnables (*Software Tool*, *Acquisition Device*) versus tous les autres non-versionnables.
- Relations entre les objets : les relations héritant du BO *IMANrelation* – la relation par défaut entre deux BOs *article* – ont été ajoutées au modèle de données. Les relations possibles entre BOs sont ensuite indiquées pour chaque BO dans le module *GRMrule*.
- Datasets : afin de prendre en compte les formats et extensions de fichiers spécifiques au domaine de la neuroimagerie, des datasets adaptés sont ajoutés au modèle de données Teamcenter. Une vingtaine de datasets ont été rajoutés au modèle au moment de la rédaction de ce chapitre. Ce nombre est amené à grandir si de nouveaux formats font leur apparition ou si un nouveau domaine d'étude est ajouté à la base de données.
- Outils : ces objets servent à indiquer avec quels logiciels vont être ouvert les fichiers d'un dataset. Ils servent à associer une extension avec un programme exécutable dans l'environnement du client. Une vingtaine d'outils personnalisés existent actuellement dans le modèle de données. Comme pour les datasets, leur nombre est susceptible d'augmenter.

Le modèle de données peut être importé et exporté dans BMIDE au format XML. Lorsqu'une modification a été apportée au modèle de données, il est nécessaire de déployer celui-ci sur le serveur pour que les clients – riches ou web – puissent en bénéficier.

Le client riche peut également être personnalisé par l'administrateur du serveur Teamcenter.

- Classification : module de l'application cliente dans Teamcenter qui est utilisé dans l'industrie manufacturière pour classer les produits en familles. Les classes peuvent être importées et exportées depuis le client sous format XML. Aucun déploiement n'est nécessaire lors de modifications de la classification, ce qui permet une grande flexibilité pour les utilisateurs.
- Préférences : décrit les façons d'afficher les relations entre instances de BOs.
- Feuilles de style : défini les façons d'afficher les informations d'une instance de BO.
- Organisation : permet de définir les utilisateurs, les rôles et les groupes.
- Projet : permet de définir un projet et l'équipe qui aura des droits sur les données du projet. Ce module va servir à la définition des études.

A cause de la quantité inhabituelle de types de BOs *article* dans la base de données, un jeu d'icônes a été créé pour aider l'utilisateur à identifier plus facilement chaque type de BO.

7.2.1.2 Mise en place de l'étude

La mise en place de l'étude dans le système PLM Teamcenter du laboratoire GIN a nécessité trois phases :

1. Création de l'étude : l'étude GINT1 est crée dans Teamcenter. Les droit d'accès à l'étude sont définis suivant les règles présentées dans le chapitre 4. Les données liées à l'étude sont stockées dans un objet de type *Study*.
2. Import des données de définition des données brutes. Ces données ne sont pas décrites de façon formelle dans le jeu de données BIL&GIN. Il a fallut les compiler depuis l'expertise des chercheurs du laboratoire. Environ trois cent objets de définition ont été créés pour décrire les données brutes du jeu de données BIL&GIN, ils ne sont pas rattachés à l'étude GINT1 mais sont disponibles pour l'ensemble des utilisateurs.
3. Import des données brutes acquises lors d'examens d'imagerie et de psychologie à partir du jeu de données BIL&GIN. Les données sont migrées depuis la base de données GINdb vers le système PLM Teamcenter grâce au logiciel Talend qui permet de faire un mapping entre les deux modèles de données, et de générer des fichiers PLMXML qui vont importer les données au bon format dans la base Teamcenter. Pour les 231 sujets, plus de soixante mille objets (hors datasets) contenant des données brutes ont été créés de cette façon dans la base de données Teamcenter, et rattachés à l'étude GINT1. La figure 7.3 résume la migration des données au GIN.

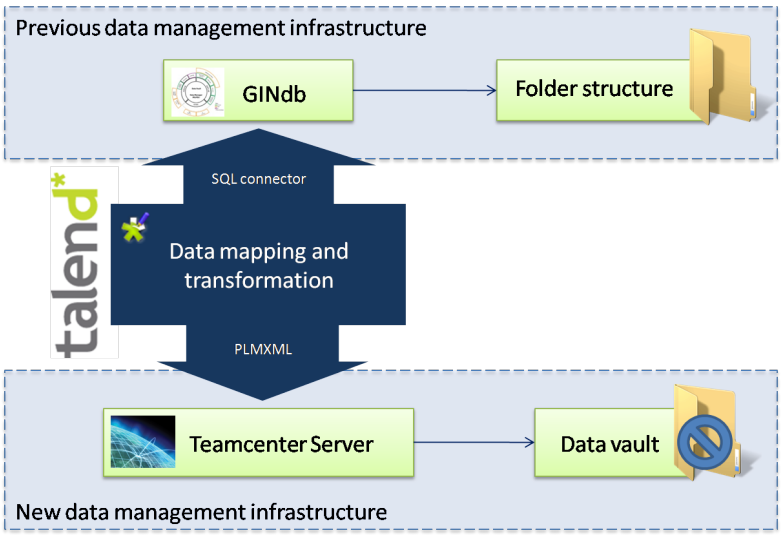


FIGURE 7.3 – Migration des données depuis GINdb vers Teamcenter. Le logiciel Talend permet d'établir un mapping entre les modèles de données des deux systèmes. Dans la base de données GINdb, les fichiers sont stockés à l'extérieur de la base dans une structure de dossiers, tandis que dans un système PLM les fichiers sont stockés dans un coffre-fort qui n'est accessible que par connexion à la base de données.

7.2.1.3 Transformation des données

Pour transformer les données brutes en données dérivées, il est nécessaire d'appliquer des traitements spécifiques. L'enjeu est double pour ce procesus : un utilisateur doit pouvoir lancer facilement de nouveaux traitements et il est nécessaire que la gestion de la provenance des données soit assurée automatiquement.

Pour opérer une chaîne de traitement sur une grille de calcul externe à Teamcenter, l'utilisateur lance un workflow depuis l'interface du système PLM. L'objet *WFI* (BO *WorkflowInput*) a été ajouté au modèle de données de Teamcenter pour regrouper toutes les informations nécessaires au lancement d'un traitement sur une grille de calcul externe :

- L'objet de définition de la chaîne de traitement (BO *Processing Definition*)
- Les objets de définition des paramètres associés à chaque étape de la chaîne de traitement (BOs *Processing Parameter*)
- Les objets de définition des données d'entrée de la chaîne de calcul, qui vont permettre de requêter dans la base ces données à partir du groupe de sujets qui sera indiqué en parallèle par l'utilisateur

L'objet WFI peut être réutilisé ultérieurement pour lancer la même chaîne de traitement sur des données différentes. Lorsqu'un workflow de traitement est demandé par l'utilisateur, il doit renseigner le WFI choisi et le groupe de sujets (BO *Subject Group*) ou le groupe de groupe de sujets.

Une fois le workflow lancé, une série d'opérations est effectuée sans nécessiter l'action de l'utilisateur. Les étapes sont présentées dans la figure 7.4 :

1. Requête sur les données d'entrée dans Teamcenter : une requête est formée et exécutée à partir des objets de définition renseignés au niveau de l'objet WFI.

2. Téléchargement des données d'entrée en local sur le serveur : les données d'entrée ainsi que les données des objets de définition sont téléchargés dans une structure en dossiers compréhensible par Nipype – un framework de traitements de données pour la neuroimagerie (Gorgolewski *et al.*, 2011).

3. Lancement du traitement sur la grille de calcul avec Nipype : le framework gère les entrées et sorties de la chaîne, ainsi que l'appel aux logiciels et l'envoi des codes de calcul.

4. Retour des données calculées en local sur le serveur : les données sont organisées par dossier (un dossier = une brique du traitement) et un fichier d'information sur le calcul est généré depuis Nipype.

5. Remontée des données calculées dans Teamcenter : les objets correspondant à la chaîne de traitement sont créés, les datasets sont stockés dans le coffre-fort de données, et les relations de provenance sont mises en place.

6. Envoi d'une alerte au commanditaire du calcul : un mail est envoyé et le workflow se termine.

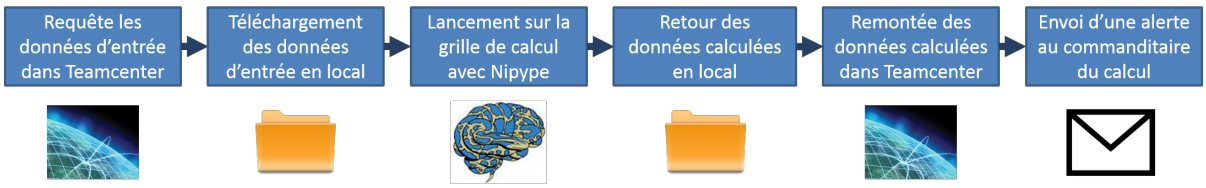


FIGURE 7.4 – Étapes de l'exécution d'un workflow de traitement.

7.2.1.4 Préparation des GMD

Les algorithmes proposés dans cette thèse ont été implémentés en java par nos soins. Une routine permet de créer des GMD statiques et dynamiques au format JGEX à partir des matrices de connectivité fonctionnelle, d'un atlas et des données associées aux sujets et aux groupes.

La structure de données java a été développée pour manipuler facilement les concepts des GMD, appliquer les algorithmes de préparation OCL et se connecter à des librairies existantes (notamment la librairie Gephi Toolkit [2] pour le calcul des propriétés et des layouts).

7.2.2 Présentation des résultats

La réalisation des étapes 1 à 8 du cas d'utilisation introduit en début de chapitre (voir le paragraphe 7.1.1) est présentée dans cette sous-section : de la création d'un étude au calcul d'un GMD pour analyser l'impact d'une caractéristique des sujets (par exemple l'âge, la latéralité ou le genre) sur la connectivité fonctionnelle au repos.

7.2.2.1 Données brutes

Les données brutes sont générées aux étapes 1 et 2 du cas d'utilisation : création d'une étude et import des données brutes. Un contrôle qualité est effectué sur les données brutes auxquelles sont associées à un statut {unverified, verified,rejected}. Les datasets peuvent être ouverts dans l'interface (fichiers textes, pdf...) ou en local sur la machine de l'utilisateur, grâce aux connexions à des logiciels externes. Les métadonnées de la classification sont affichés dans un onglet de l'interface PLM. La figure 7.5 montre dans l'interface de Teamcenter les données brutes associées à un sujet : examens, acquisitions et unités de données.

Pour l'étude GINT1, tous les examens d'imagerie IRM fonctionnelle à l'état de repos conscient ont été acquis suivant le même protocole qui est décrit dans la base de données PLM. La figure 7.6 présente la structure des objets de définition correspondants dans la base PLM implémentée au GIN.

7.2.2.2 Calcul de la matrice de connectivité fonctionnelle d'un sujet ayant passé un examen IRMf au repos

Les données dérivées sont calculées dans les étapes 3 à 8 de cas d'utilisation. Les étapes 3 et 4 permettent de calculer les matrice de connectivité fonctionnelle pour chaque sujet du

2. http://gephi.github.io/toolkit/

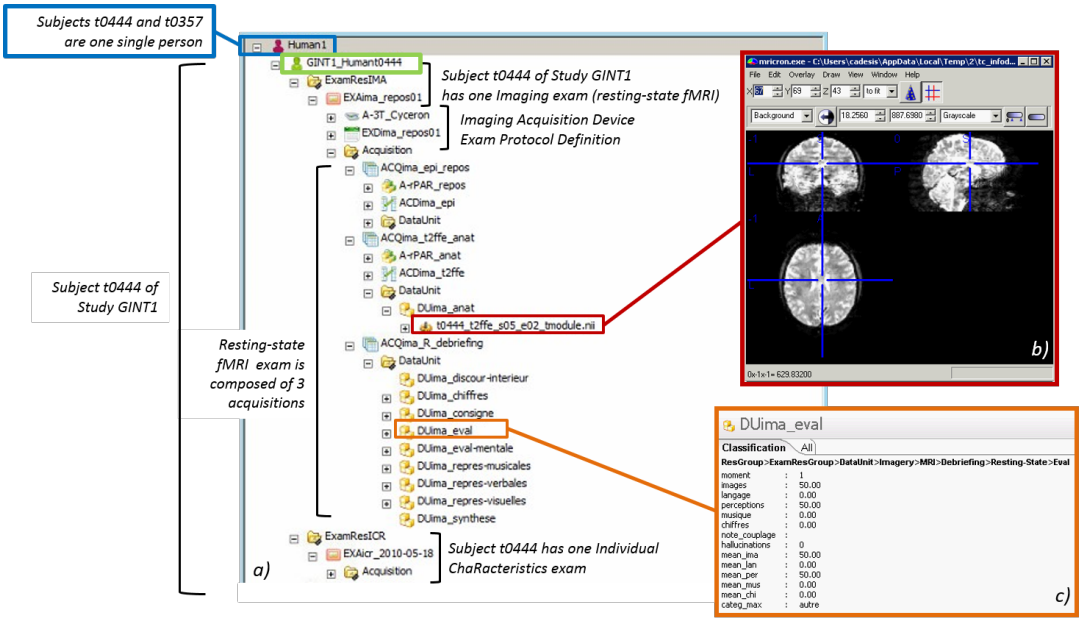


FIGURE 7.5 – Examens passés par le sujet *t0444* dans le cadre de l'étude *GINT1* dans Teamcenter. a) Arbre des données appartenant au sujet *t0444*, b) Image IRM anatomique visualisée depuis Teamcenter, c) Valeurs des attributs de classification pour l'objet *DUima_anat*.

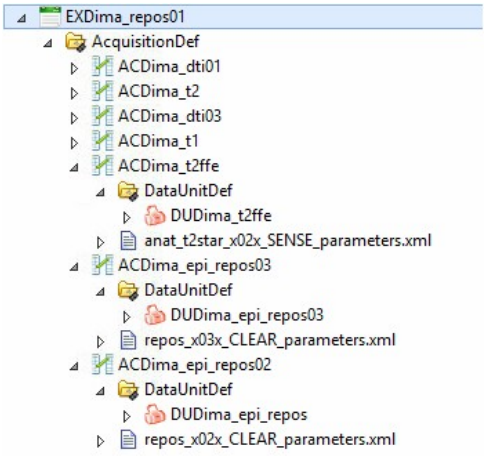


FIGURE 7.6 – Objets de définition d'un examen d'imagerie IRM depuis le module *relation browser* de Teamcenter.

jeu de données. Les chaînes de traitement ne peuvent être lancée que sur des données brutes et des données dérivées dont le statut est *verified*. L'utilisateur indique au workflow de calcul les données d'entrée et la définition de la chaîne de traitements à appliquer. L'ensemble des données est envoyé sur des grilles de calcul externes au système PLM, puis lorsque les calculs sont terminées, les données résultats sont remontées dans Teamcenter où la provenance est crée automatiquement. La dernière étape du workflow consiste à prévenir l'utilisateur que les résultats sont arrivés dans la base de données.

La figure 7.7 présente les résultats de la chaîne de traitement dans Teamcenter, et illustre les différentes façons de naviguer à travers la provenance des données.

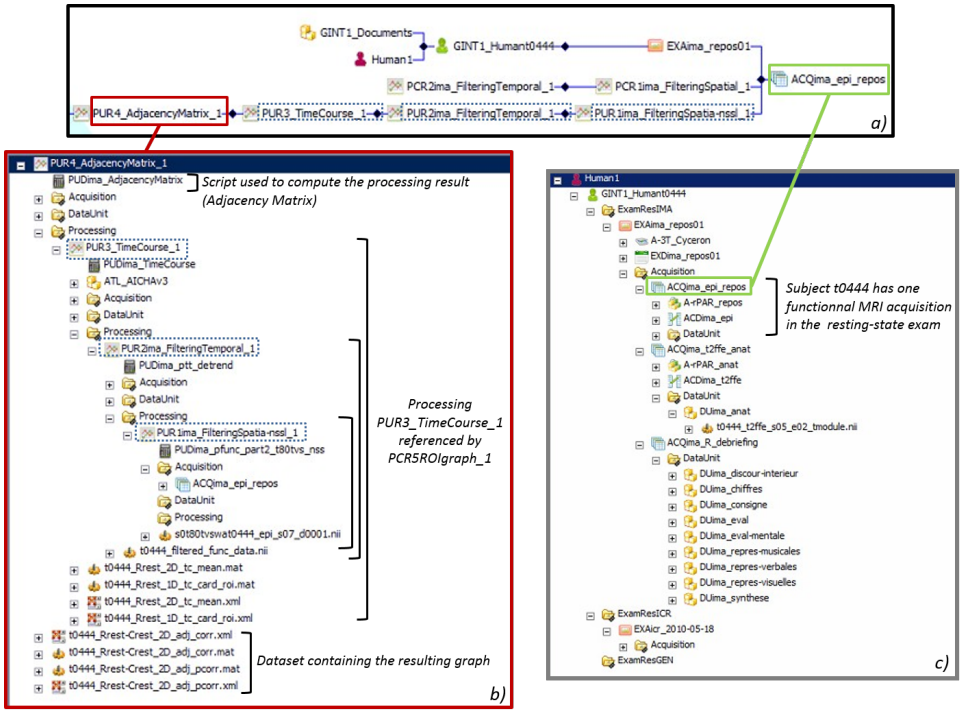


FIGURE 7.7 – Navigation dans Teamcenter : a) visualisation de la chaîne de calcul qui utilise les données d'imagerie de repos *ACQima_epi_repos*, b) visualisation de la provenance du calcul de création de la matrice d'adjacence de connectivité fonctionnelle *PUR4_AdjacencyMatrix_1*, c) Arbre des données appartenant au sujet *t0444* dans lequel est stocké l'acquisition d'imagerie de repos *ACQima_epi_repos*.

7.2.2.3 Création d'un GMD sur des données de groupe

Les étapes 5 à 8 permettent la création d'un GMD prêt pour l'exploration visuelle de la connectivité fonctionnelle cérébrale selon plusieurs dimensions (l'étape 9 – l'exploration OCL – du cas d'utilisation est développée dans la section 7.3 de ce chapitre). Dans un premier temps, les groupes de sujets sont crées à partir de requêtes sur des paramètres d'intérêt. Les systèmes PLM permettent de construire des requêtes personnalisées sur des relations complexes ; deux exemples de requêtes sont présentés dans la figure 7.8. Une fois les groupes créés, une chaîne de traitements est appliquée en parallèle sur tous les groupes de sujets pour calculer les matrices de connectivité de chaque groupe. Un GMD est créé à partir des matrices de connectivité de

chaque groupe, puis la préparation OCL des données est calculée pour obtenir un GMD prêt pour l'exploration visuelle.

La chaîne de traitement dans le logiciel Teamcenter, depuis la matrice de connectivité fonctionnelle d'un sujet individuel jusqu'au GMD préparé pour la méthode de visualisation OCL, est présentée dans la figure 7.9.

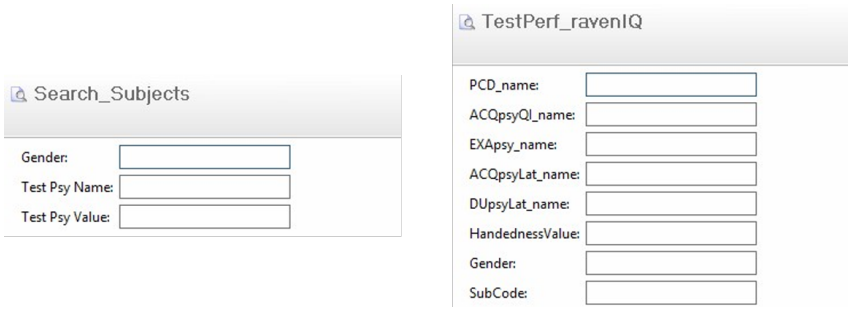


FIGURE 7.8 – Exemples de requêtes personnalisées dans Teamcenter ; a) Recherche de sujet en fonction du genre et du résultat à un test psychologique et b) Recherche de donénes dérivées en fonction de plusieurs critères (sujets, acquisition d'entrée, etc).

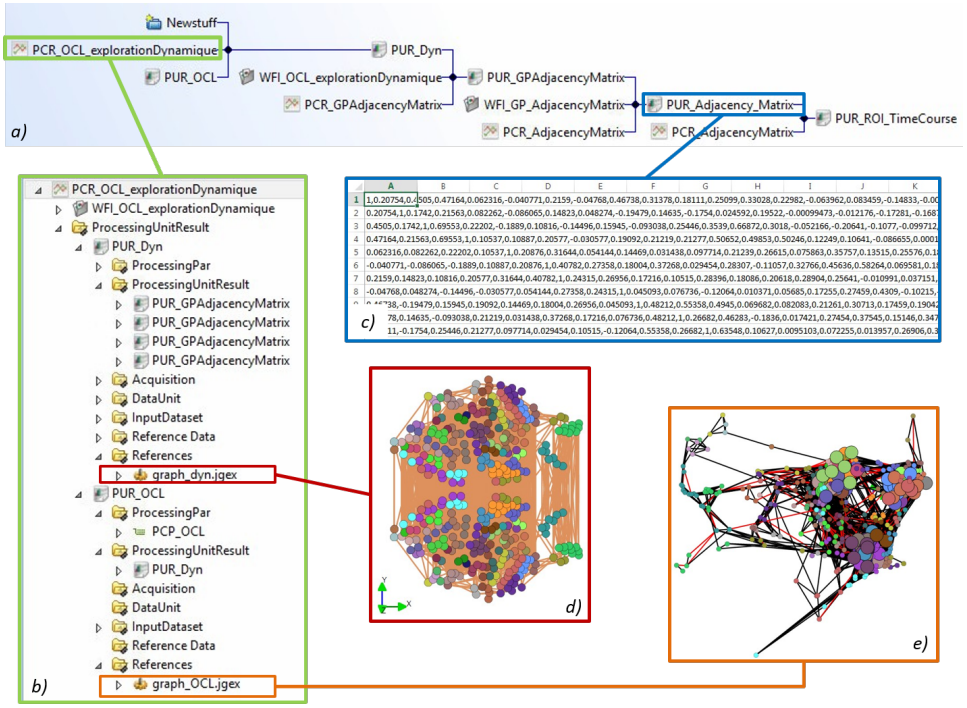


FIGURE 7.9 – Provenance d'un GMD préparé pour l'exploration dynamique de la connectivité fonctionnelle cérébrale avec la méthode OCL. a) Chaîne de traitement depuis la matrice de connectivité d'un sujet, b) Provenance de la chaîne de traitement qui créé le GMD puis le prépare pour l'exploration OCL, c) Matrice de connectivité, d) GMD créé à partir des matrices d'adjacence des quatre groupes, e) GMD préparé pour la visualisation OCL affiché dans le logiciel SwoViewer.

7.2.2.4 Publication des résultats

L'étape 10 du cas d'utilisation concerne la publication des résultats d'une étude. Un statut est associé aux objets *référence bibliographique* : {draft,review,rejected,published}. Lors de la création d'une publication dans la base, son statut est *draft*. Si la publication n'est pas associée à une étude, l'utilisateur lance un workflow qui va modifier le statut à *published*. Si au contraire la publication est un article en ébauche, elle devra passer par le statut *review* quand l'article est soumis à un éditeur, et prendra le statut final de *rejected* si l'article est refusé, et de *published* s'il est accepté.

Le suivi d'un article scientifique depuis son ébauche à sa publication n'a pas pu être testée, car aucune publication n'était prévue dans l'intervalle d'expérimentation. Les acquisitions du projet I-share prévues pour fin 2015 vont permettre de tester l'intégralité de la chaîne depuis l'acquisition directe des données jusqu'à la publication des résultats issus de l'analyse des données. Pour le moment, les objets *référence bibliographique* sont utilisés pour des publications existantes qui vont être référencées par des objets de définition. Un exemple est présenté dans la figure 7.10 ; le test psychologique d'Edinburgh, qui permet de mesurer la latéralité d'un sujet, a été décrit dans un article de Oldfield en 1971.

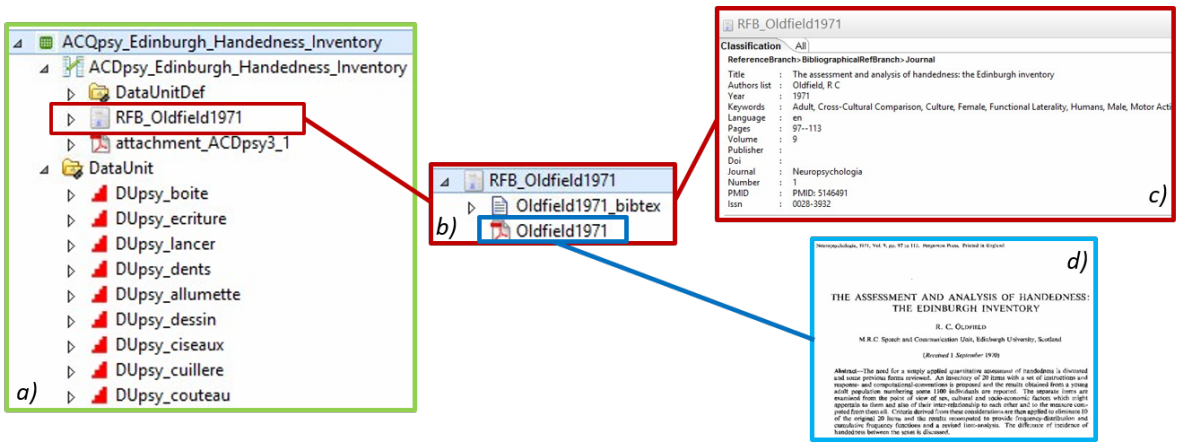


FIGURE 7.10 – La publication (Oldfield, 1971) dans Teamcenter avec le modèle de données BMI-LM. a) La publication est référencée par l'objet de définition du test d'Edinburgh dans la base de données, b) L'objet *référence bibliographique* représentant la publication, c) Métadonnées associées à la publication, d) Fichier pdf de l'article stocké dans la base de données.

7.2.3 Critique de l'implémentation

L'implémentation dans Teamcenter a été présentée aux membres du GIN – le même panel que celui qui avait participé aux interviews de définition du besoin (annexe C). Les chercheurs reconnaissent dans le modèle de données BMI-LM les concepts qui leur sont familiers. Ils valident la nouvelle structuration de leurs données, et accueillent avec enthousiasme les apports de la classification. En effet, la classification leur permet d'apprivoiser rapidement la base de données et d'adapter facilement la structuration des méta-données à leurs pratiques de recherche, tandis que le modèle générique BMI-LM constitue le squelette de la gestion des données.

Les critiques principales concernent l'interface du logiciel (affichage du client riche de Teamcenter) : il y a trop de menus, d'icones et de fenêtres dans l'environnement. Par conséquent, retrouver visuellement les information n'est pas immédiat : la navigation d'objets en objets n'est pas possible au sein d'une seule fenêtre. Les chercheurs souhaitent utiliser une interface ergonomique et simplifiée qui ne nécessite que peu de choix à faire pour chaque option. Ils souhaitent prendre en main le logiciel sans pratiquement aucune phase d'apprentissage. Sur ce point, les chercheurs préfèrent donc leur ancien système de gestion des données (GINdb), qui possède une interface simplifier pour la consultation de données. Par ailleurs, les chercheurs souhaitent effectuer des requêtes complexes – ce qui signifie des requêtes incluant des informations sur la provenance –, mais l'interface du module qui permet de créer ces requêtes dans Teamcenter (QueryBuilder) n'est pas très ergonomique et nécessite de connaître finement le modèle de données du système. Ces constatations ne favorisent pas l'usage occasionnel de la base pour chercher et réutiliser des informations.

Une synthèse des critiques de l'implémentation est présentée dans la table 7.2.

Retours positifs	**Points d'amélioration**
concepts familiers flexibilité du modèle	interface des systèmes PLM – navigation – requêtes

TABLE 7.2 – Synthèse des critiques de l'implémentation PLM.

7.3 Exploration dynamique de réseaux cérébraux

Dans cette section nous appliquons la méthode d'exploration OCL aux GMD créés à partir de matrices de connectivité fonctionnelle de groupes de sujets définis par l'utilisateur. La méthode OCL est successivement appliquée à un GMD dont la dimension est l'âge des sujets, et un GMD dont les dimensions sont le genre et la latéralité des sujets.

7.3.1 Application de l'OCL à l'étude de la connectivité fonctionnelle cérébrale

7.3.1.1 Dimensions de l'exploration

La connectivité fonctionnelle cérébrale des 231 sujets du jeu de données va être étudiée selon plusieurs critères :

- L'âge : les sujets sont répartis en classes d'intervalle de cinq ans qui sont ordonnées de façon croissante.
- Le genre et la latéralité : de façon arbitraire, les dimensions sont respectivement ordonnées {Homme,Femme} et {Gauche,Droite}.

Les GMD à explorer présentent les caractéristiques suivantes :

- 384 nœuds (un nœud par région du cerveau en suivant l'atlas AICHA), ce qui est supérieur à la limite commune de 50 nœuds pour les graphes de petite taille, mais inférieur aux

milliers de nœuds de graphes de grande taille.
– graphe complet et pondéré par les valeurs de connectivité.

7.3.1.2 Préparation des données

La méthode d'exploration OCL nécessite de préparer le GMD selon huit étapes (voir chapitre 6. Les détails de la préparation pour notre cas d'étude sont présentés ci-dessous pour chaque étape :

1. Filtre sur le poids des arêtes : dans un premier temps, nous choisissons d'éliminer 95% des arêtes du graphe. Nous allons donc obtenir un graphe avec un nombre d'arêtes constant sur l'ensemble des états.
2. Calcul des éléments du graphe de synthèse : moyenne des éléments. Dans notre cas les nœuds sont identiques sur l'ensemble des états, par conséquent seul le poids de synthèse (qui correspond au produit de la moyenne des poids et de la fréquence d'existence) est calculé.
3. Calcul des éléments constants : les paramètres optimaux sont déterminés expérimentalement, puis les éléments constants sont calculés selon l'algorithme introduit dans le chapitre 6.
4. Layout sur les éléments constants actifs et leurs arêtes : le layout OpenOrd (Martin *et al.*, 2011) est appliqué sur le graphe de synthèse partiel (éléments actifs). Ce layout a été initialement développé en C++ [3], mais nous avons utilisé le plugin java *OpenOrd Layout* dévelopé pour le logiciel Gephi [4]. Cet algorithme multi-niveaux est basé sur le layout de force de Frutcherman-Reingold (Fruchterman & Reingold, 1991) et un clustering d'arêtes moyennes (utilisant le poids des arêtes et la distance aux clusters), ce qui le rend particulièrement adapté aux grands graphes. L'algorithme est constitué de cinq phases auxquelles un poids est associé et détermine leur importance relative : liquid (liquide), expansion (extension), cool-down (ralentissement), crunch (crise), simmer (frémissement).
5. Fixation des positions des éléments constants actifs : les positions calculées à l'étape précédente (éléments actifs constants) sont sauvegardées dans le GMD à préparer.
6. Layout sur tous les éléments : le layout forceAtlas2 (Jacomy *et al.*, 2011) est appliqué sur le graphe de synthèse complet, pour lequel les positions des nœuds constants actifs ont été rendues fixes.
7. Fixation des positions des éléments constants inactifs : les positions calculées à l'étape précédente (éléments inactifs inconstants) sont sauvegardées dans le GMD à préparer.
8. Layout sur chaque état du GMD : le layout forceAtlas2 est appliqué sur chaque état du GMD, pour lesquels les positions des nœuds constants (actifs et inactifs) ont été rendues fixes.

3. http ://www.cs.sandia.gov/ smartin/software.html
4. https ://marketplace.gephi.org/plugin/openord-layout/

L'ensemble de la chaîne de préparation des données a été développée en java, comme présenté dans le paragraphe 7.2.1.4.

7.3.2 Exploration suivant l'âge des sujets

Nous souhaitons étudier l'influence de l'âge sur la connectivité fonctionnelle des sujets du jeu de données. Des intervalles de cinq ans sont utilisés pour former les classes d'étude. Seules les quatre premières classes ({20.5}, {25.5}, {30.5} et {35.5}) contiennent suffisamment de sujets pour être exploitées, ce qui rapporte notre effectif total d'étude à 218 sujets (au lieu des 231 initiaux). Les effectifs pour chaque classe et les proportions suivant le genre et la latéralité des sujets sont présentés dans la table ci-dessous :

Classe		**20.5**	**25.5**	**30.5**	**35.5**
Effectif		93	82	27	16
Homme	Gauche	30.1%	25.6%	7.4%	18.8%
	Droite	15.1%	23.2%	37.0%	56.2%
Femme	Gauche	30.1%	24.4%	14.8%	6.2%
	Droite	24.7%	26.8%	40.7%	18.8%

TABLE 7.3 – Répartition des effectifs des classes de sujets suivant l'âge.

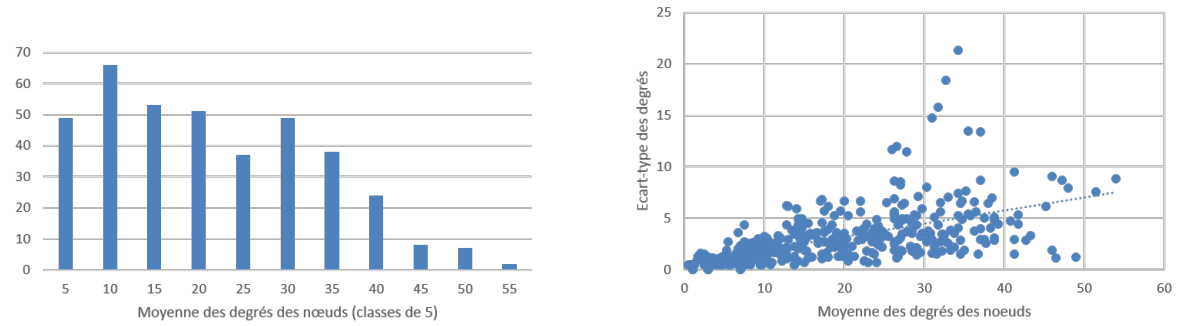
Quatre groupes de sujets correspondant aux classes sont créés, et les matrices d'adjacence de connectivité fonctionnelle correspondantes sont calculées par la médiane de la connectivité de tous les sujets d'un groupe. Pour finir, le graphe JGEX dynamique est calculé à partir des matrices d'adjacence des groupes. L'intégralité des résultats est présenté dans annexe L.

7.3.2.1 Optimisation des paramètres

L'étude est menée sur les 5% d'arêtes aux poids les plus élevées pour chaque état du graphe dynamique. A partir du graphe filtré, le graphe est analysé d'un point de vue topologique :

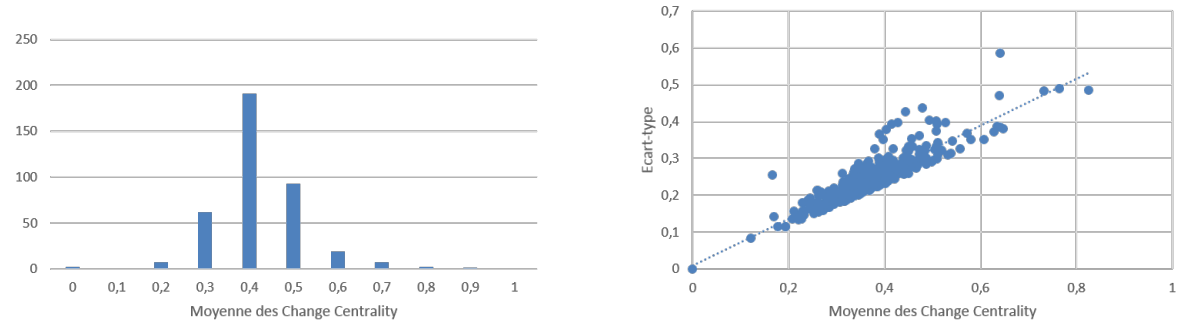
- Degré des nœuds : La répartition de la moyenne du degré des nœuds est donnée dans la figure 7.11.
- Change Centrality des nœuds : La répartition de la moyenne du degré des nœuds est donnée dans la figure 7.12.

Nous observons que le degré maximal moyen sur les quatre états liés aux classes d'âge est inférieur à 55 (soit 14% du nombre total de nœuds) et que plus de 50% des nœuds a un degré moyen inférieur à 20 (soit 5% du nombre total de nœuds). 49 nœuds ont un degré inférieur à 5 (soit 13% du nombre total de nœuds) et 115 nœuds ont un degré inférieur à 10 (soit 30% du nombre total de nœuds). La moitié des nœuds présente une valeur de Change Centrality moyenne inférieure à 0,4.



a) b)

FIGURE 7.11 – Répartition de la moyenne du degré des nœuds pour 4 classes d'âge (20.5, 25.5, 30.5 et 35.5) d'un graphe filtré à 95%



a) b)

FIGURE 7.12 – Répartition de la moyenne de la mesure du Change Centrality pour 4 classes d'âge (20.5, 25.5, 30.5 et 35.5) d'un graphe filtré à 95%

Nous en déduisons un jeu de paramètres à tester :

- Valeur seuil du degré des nœuds pour la détermination des nœuds actifs $iThesholdAct$: 5.0
- Valeur seuil du degré des nœuds pour la détermination des nœuds inactifs $iThresholdInac$: 0.5
- Valeur seuil liée à la mesure du Change Centrality $iThresholdCc$: 0.4
- Valeur seuil liée à l'écart-type des mesures de Change Centrality $iThresholdSdCc$: 0.25

A partir de ces premiers paramètres, nous allons essayer de déterminer les pourcentages optimaux de nœuds fixes $iPerFixedNodes$ et de nœuds actifs parmi ces nœuds fixes $iPerNodesAct$. Nous lançons le calcul de détermination des nœuds constants en faisant varier ces paramètres respectivement de 10 à 50%, et de 50 à 70%. Le tableau 7.4 présente le nombre de nœuds constants (actifs et inactifs) obtenus.

Nœuds constants				Nœuds actifs					Nœuds inactifs				
%th	th	re	e(%)	%th	th	re	e(%)	iter	%th	th	re	e(%)	iter
10	38.4	41	6.8	70	26.9	38	41.4	7	30	11.5	3	-74.0	7
10	38.4	45	17.2	60	23.0	24	4.2	6	40	15.4	21	36.7	14
10	38.4	45	17.2	50	19.2	24	25.0	6	50	19.2	21	9.4	14
20	76.8	77	0.3	70	53.8	56	4.2	8	30	23.0	21	-8.9	7
20	76.8	77	0.3	60	46.1	56	21.5	8	40	30.7	21	-31.6	8
20	76.8	77	0.3	50	38.4	38	-1.0	7	50	38.4	39	1.6	10
30	115.2	132	14.6	70	80.6	93	15.3	10	30	34.6	39	12.8	7
30	115.2	117	1.6	60	69.1	70	1.3	9	40	46.1	47	2.0	8
30	115.2	115	-0.2	50	57.6	56	-2.8	8	50	57.6	59	2.4	10
40	153.6	165	7.4	70	107.5	118	9.7	11	30	46.1	47	2.0	6
40	153.6	152	-1.0	60	92.2	93	0.9	10	40	61.4	59	-4.0	8
40	153.6	154	0.3	50	76.8	70	-8.9	9	50	76.8	84	9.4	10
50	192	197	2.6	70	134.4	138	2.7	12	30	57.6	59	2.4	6
50	192	202	5.2	60	115.2	118	2.4	11	40	76.8	84	9.4	8
50	192	189	-1.6	50	96.0	93	-3.1	10	50	96.0	96	0.0	9

TABLE 7.4 – Répartition des nœuds constants identifiés pour la méthode OCL appliquée à l'étude de l'âge, avec les paramètres fixes $iThesholdAct$=5.0, $iThresholdInac$=0.5, $iThresholdCc$=0.4, $iThresholdSdCc$=0.25, et les paramètres $iPerFixedNodes$ (variant de 10 à 50%) et $iPerNodesAct$ (variant de 50 à 70%). *Légende :* %th=pourcentage théorique de nœuds, th=nombre théorique de nœuds, re=nombre réel de nœuds, e(%)=écart en pourcentage de la valeur réel à la valeur théorique, iter=nombre d'itérations nécessaires pour le calcul.

Les nœuds actifs sont déterminés en premier, donc si le nombre total de nœuds est atteint, le nombre de nœuds actifs sera inférieur au nombre théorique prévu. Le nombre d'itérations maximales est de 14, ce qui reste faible. Cela suggère que $iThesholdAct$ et $iThesholdInact$ pourraient être diminués pour permettre une détermination plus fine du nombre de nœuds constants identifiés qui tendent vers la valeur recherche du nombre de nœuds constants.

Les nœuds actifs structurent et orientent le layout, tandis que les nœuds inactifs sont rejetés en périphérie du layout. Trop de nœuds actifs rendus fixes empêchent d'identifier les changements topologiques globaux, tandis que trop de nœuds inactifs fixes ne permet pas de stabiliser les structures en cluster récurrentes. En effet, les nœuds qui ne sont pas identifiés comme constants sont des nœuds moins stables que les autres sur la dimension, c'est-à-dire que leurs connexions, et donc leur rôle au sein du graphe, varie en fonction des états.

L'impact du pourcentage de nœuds actifs par rapport au pourcentage fixe de nœuds constants ($iPerFixedNodes$) sur le layout final est présenté dans la figure 7.13 pour l'état correspondant à la classe d'âge {30,5}, et l'impact du pourcentage de nœuds constants par rapport au pourcentage fixe de nœuds actifs ($iPerNodesAct$) sur le layout final est présenté dans la figure 7.13 pour l'état correspondant à la classe d'âge {30,5}. Des figures supplémentaires sont données dans l'annexe L. Nous observons d'un point de vue qualitatif que plus le pourcentage de noeuds constants à fixer est élevé et plus le pourcentage de noeuds actifs à fixer doit être faible pour pouvoir constater une stabilité du layout "suffisante".

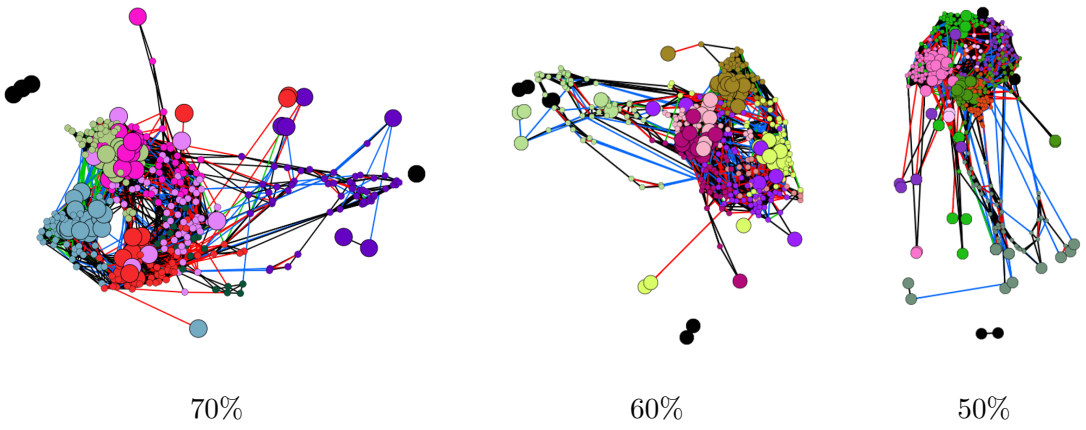


FIGURE 7.13 – Impact du pourcentage de nœuds actifs par rapport à un pourcentage fixe de nœuds constants sur le layout final de l'état correspondant à la classe d'âge {30.5}, $iPerFixedNodes$=20%. *Légende* : layout LàC, grands nœuds = nœuds constants, couleurs = clusters, les trois résultats sont indépendants.

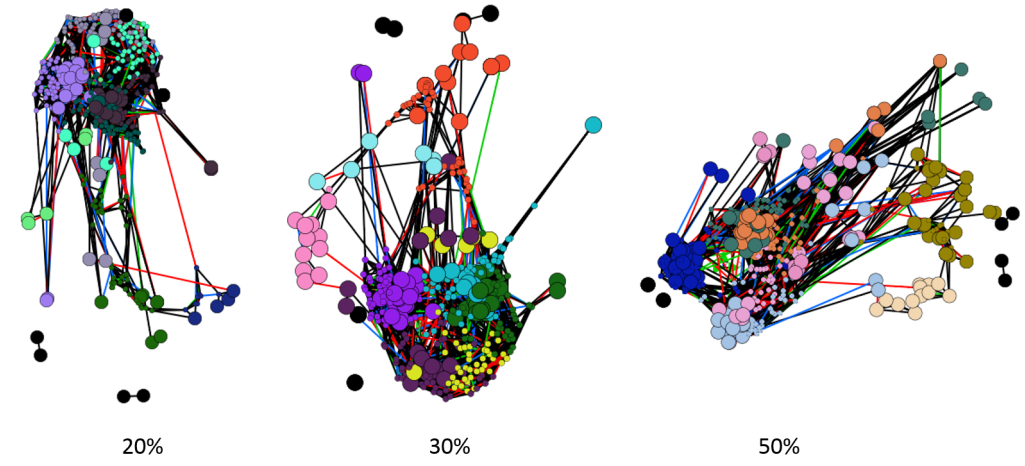


FIGURE 7.14 – Impact du pourcentage de nœuds constants par rapport à un pourcentage fixe de nœuds actifs sur le layout final de l'état correspondant à la classe d'âge {25.5}, $iPerNodesAct$=50%. *Légende* : layout LàC, grands nœuds = nœuds constants, couleurs = clusters, les trois résultats sont indépendants.

Nous fixons donc pour la suite $iPerFixedNodes$ à 30% et $iPerNodesAct$ à 60%, ce qui donne les paramètres retenus :

Paramètre	Valeur
$iThresholdEdges$	95%
$iPerFixedNodes$	30%
$iPerNodesAct$	60%
$iThesholdAct$	5.0
$iThresholdInac$	0.5
$iThresholdCc$	0.4
$iThresholdSdCc$	0.25

TABLE 7.5 – Paramètres finaux pour l'exploration OCL suivant l'âge.

7.3.2.2 Résultats du LàC

La liste des 117 nœuds constants obtenus avec leurs caractéristiques (degré, change centrality, position) est donnée dans l'annexe L, de même que des figures complémentaires à ce qui est présenté dans cette section. La répartition finale des nœuds est données dans la table 7.6 : 117 nœuds constants ont été identifiés, soit 30,5% du nombre total de nœuds dans le graphe.

	théorique	résultat	itérations
nœuds actifs	69	70	9
nœuds inactifs	46	47	8
total	115	117	9

TABLE 7.6

Le graphe de synthèse obtenu est présenté dans la figure 7.15 : les nœuds constants identifiés pendant la préparation des données sont reliés par les arêtes de synthèse du GMD. Nous observons trois clusters clairement mis en évidence. L'identification des régions anatomiques concernées est facilitée grâce à la juxtaposition de deux vues – le graphe de synthèse et le graphe entier avec son layout anatomique 2D – et à l'utilisation de codes de couleur anatomiques – une couleur par zone englobant plusieurs régions de l'atlas AICHA.

Nous observons une relative symétrie hémisphérique chez les nœuds constants. Cette symétrie n'est pas exacte, ce qui pourrait s'expliquer par le filtre appliqué aux arêtes.

La figure 7.16 montre les configurations de graphe obtenues pour chaque classe d'âge avec le layout LàC. A titre de comparaison, le layout "libre" calculé avec l'algorithme de force multi-niveaux OpenORD est également présenté pour chaque classe d'âge. Des couleurs aléatoires sont attribuées aux clusters des configurations de graphe afin d'aider à repérer les changements topologiques globaux d'un état à l'autre. Nous constatons que malgré la coloration des clusters, il est difficile d'identifier les parties communes des configurations de graphe sans le LàC : soit parce que l'orientation aléatoire de la configuration est aléatoire, soit parce que l'allure du graphe est très différente – par exemple entre les classes {20.5} et {35.5}.

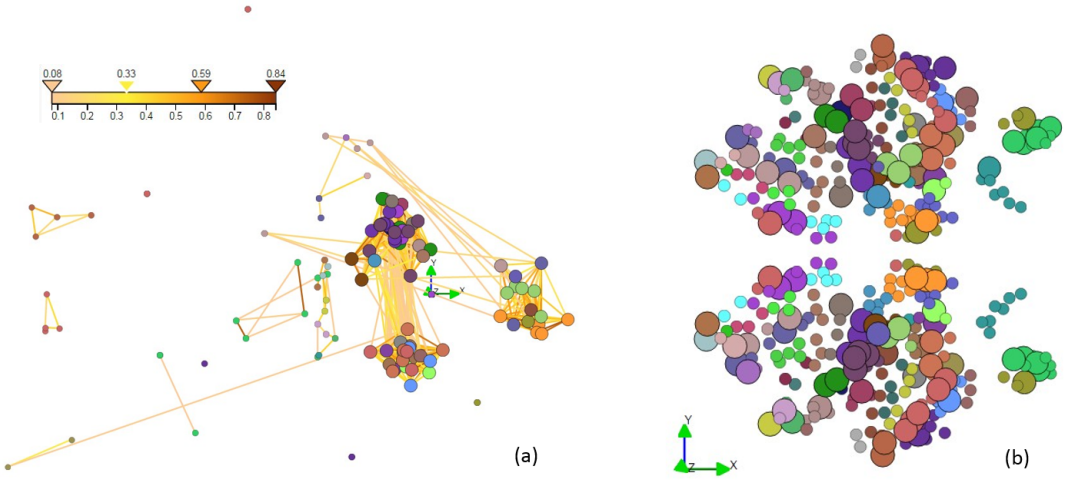


FIGURE 7.15 – Nœuds constants identifiés sur les quatre classes d'âge : (a) Graphe de synthèse (la valeur du poids de synthèse est donnée dans l'échelle de couleur) (b) Layout anatomique 2D (nœuds constants = grands nœuds). Les couleurs des nœuds représentent une segmentation anatomique.

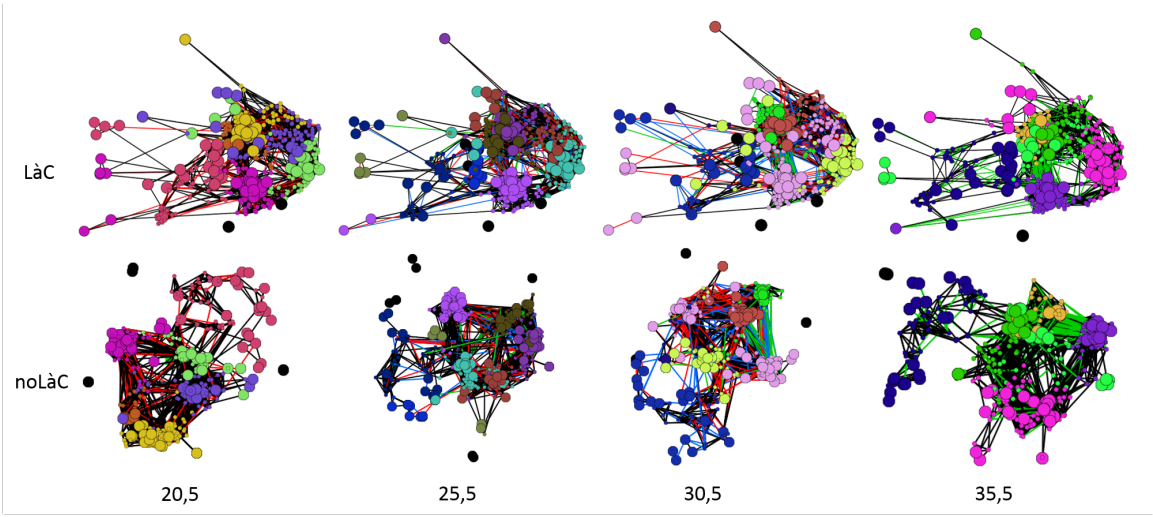


FIGURE 7.16 – Comparaison du layout avec (LàC) et sans contraintes (noLàC) pour les quatre classes d'âge.

7.3.2.3 Résultats de l'exploration

Nous présentons les fonctionnalités interactives proposées pour l'exploration des GMD appliquées à l'étude de l'impact de l'âge sur la connectivité fonctionnelle cérébrale. Les variables visuelles sont limitées – couleur, taille et visibilité des éléments du graphe –, et nous privilégions une combinaison de vues inter-connectées du GMD.

Grâce aux vues multiples connectées du visualiseur de graphes SwoViewer, il est possible de mettre en exergue un nœud sélectionné et les nœuds qui lui sont adjacents. Un exemple est présenté dans la figure 7.17 : la sélection d'un nœud dans une vue *en contexte* permet de mettre

en évidence les nœuds qui lui sont adjacents et propage l'affichage dans la ou les autres vues de l'interface, comme par exemple une vue anatomique 2D des régions du cerveau.

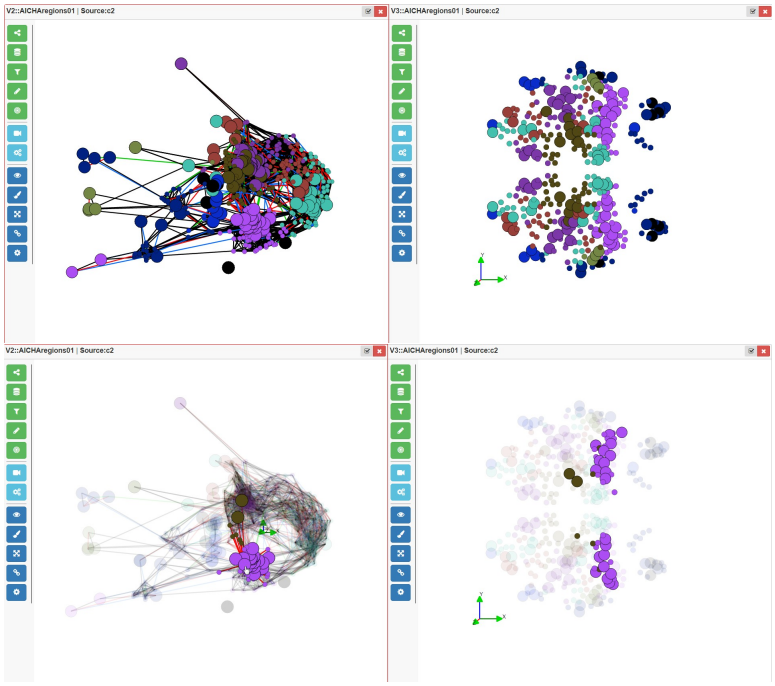


FIGURE 7.17 – Exemple d'exploration multi-vues : la classe d'âge {20.5} est affiché avec le layout LàC sur la gauche, et un layout anatomique 2D sur la droite. Les deux vues partagent des couleurs aléatoires associées aux clusters de la configuration de graphe. (Bas) La sélection d'un nœud sur l'une des vues le met en exergue ainsi que les nœuds qui lui sont adjacents, et ce sur les deux vues.

Les vues *en contexte* permettent d'approcher l'étude des GMD de façon statique : ces vues agissent comme des lentilles fisheye, mais sur un état dimensionnel. La visualisation se focalise sur un état, mais indique les changements vis à vis des états adjacents suivant l'ordre de la dimension. Les arêtes *en contexte* pour les quatre états sont présentées avec le layout LàC dans la figure 7.18 dans une présentation small multiples ; la couleur des arêtes représente les évolutions des états adjacents. Nous pouvons pas exemple observer que des beaucoup de connexions apparaissent entre deux clusters à l'état {30.5} et que de nouvelles connexions apparaissent encore à cet endroit à l'état {35.5}.

7.3.3 Exploration suivant le genre et la latéralité des sujets

Nous souhaitons à présent explorer la connectivité des sujets suivant leur genre {Femme,Homme} et leur latéralité {Gauche,Droite}, ce qui signifie que le GMD résultant présente deux dimensions, que nous choisissons arbitrairement d'ordonner dans l'ordre ci-dessus. Sur les 231 sujets du jeu de données, nous obtenons quatre classes dont les effectifs sont similaires à 89%, comme le montre la table 7.7. L'intégralité des résultats est présenté dans l'annexe M.

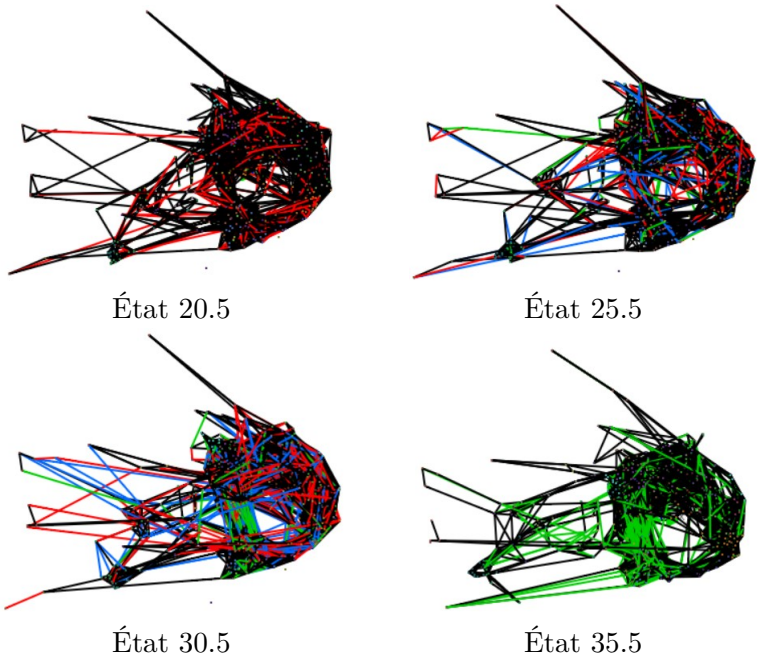


FIGURE 7.18 – Comparaison en contexte sur l'âge. Rouge : l'arête disparait à l'état suivant, Vert : l'arête apparait à l'état courant, Bleu : l'arête apparait à l'état courant et disparait à l'état suivant.

Genre	**Latéralité**	**Effectifs**
Femme	Gauche	56
	Droite	58
Homme	Gauche	55
	Droite	62

TABLE 7.7 – Répartition des effectifs des classes de sujets suivant le genre et la latéralité.

7.3.3.1 Optimisation des paramètres

Nous suivons le même raisonnement que pour l'étude suivant l'âge ; après avoir étudié la répartition de la moyenne et de l'écart-type des degrés et de la mesure du Change Centrality, nous en déduisons un premier jeu de paramètres à tester :

- Valeur seuil du degré des nœuds pour la détermination des nœuds actifs $iThesholdAct$: 5.0
- Valeur seuil du degré des nœuds pour la détermination des nœuds inactifs $iThresholdInac$: 0.5
- Valeur seuil liée à la mesure du Change Centrality $iThresholdCc$: 0.4
- Valeur seuil liée à l'écart-type des mesures de Change Centrality $iThresholdSdCc$: 0.25

Le tableau 7.8 présente le nombre de nœuds et le nombre d'itérations nécessaires pour les nœuds actifs et les nœuds inactifs en fonction de la variation des paramètres $iPerFixedNodes$ et $iPerNodesAct$.

Nœuds constants				Nœuds actifs					Nœuds inactifs				
%th	th	re	e(%)	%th	th	re	e(%)	iter	%th	th	re	e(%)	iter
10	38.4	38	-1.0	70	26.9	27	0.4	6	30	11.5	11	-4.5	9
10	38.4	38	-1.0	60	23.0	27	14.7	6	40	15.4	11	-28.4	9
10	38.4	38	-1.0	50	19.2	27	28.9	6	50	19.2	11	-42.7	9
20	76.8	77	0.3	70	53.8	56	4.0	8	30	23.0	21	-8.9	7
20	76.8	77	0.3	60	46.1	56	17.7	8	40	30.7	21	-31.6	8
20	76.8	74	-3.6	50	38.4	36	-6.7	7	50	38.4	38	-1.0	10
30	115.2	141	22.4	70	80.6	103	21.7	10	30	34.6	38	10.0	7
30	115.2	119	3.3	60	69.1	71	2.6	9	40	46.1	48	4.2	8
30	115.2	110	-4.5	50	57.6	56	-2.9	8	50	57.6	54	-6.3	10
40	153.6	151	-1.7	70	107.5	103	-4.4	10	30	46.1	48	4.2	6
40	153.6	170	10.7	60	92.2	103	10.5	10	40	61.4	67	9.0	9
40	153.6	152	-1.0	50	76.8	71	-8.2	9	50	76.8	81	5.5	10
50	192	185	-3.6	70	134.4	131	-2.6	11	30	57.6	54	-6.3	6
50	192	212	10.4	60	115.2	131	12.1	11	40	76.8	81	5.5	8
50	192	198	3.1	50	96.0	103	6.8	10	50	96.0	95	-1.0	9

TABLE 7.8 – Répartition des nœuds constants identifiés pour la méthode OCL appliquée à l'étude du genre et de la latéralité des sujets, avec les paramètres fixes $iThesholdAct$=5.0, $iThresholdInac$=0.5, $iThresholdCc$=0.4, $iThresholdSdCc$=0.25, et les paramètres $iPerFixedNodes$ (variant de 10 à 50%) et $iPerNodesAct$ (variant de 60 à 70%). *Légende :* %th=pourcentage théorique de nœuds, th=nombre théorique de nœuds, re=nombre réel de nœuds, e(%)=écart en pourcentage de la valeur réel à la valeur théorique, iter=nombre d'itérations nécessaires pour le calcul.

Nous fixons donc pour la suite $iPerFixedNodes$ à 20% et $iPerNodesAct$ à 70%, ce qui donne les paramètres retenus :

Paramètre	Valeur
$iThresholdEdges$	95%
$iPerFixedNodes$	20%
$iPerNodesAct$	70%
$iThesholdAct$	5.0
$iThresholdInac$	0.5
$iThresholdCc$	0.4
$iThresholdSdCc$	0.25

TABLE 7.9 – Paramètres finaux pour l'exploration OCL suivant le genre et la latéralité.

7.3.3.2 Résultats du LàC

La liste des 77 nœuds constants obtenus avec leurs caractéristiques (degré, change centrality, position) est donnée dans l'annexe M, de même que l'intégralité des figures résultat. La répartition finale des nœuds est données dans la table 7.10 : 77 nœuds constants ont été identifiés, soit 20% du nombre total de nœuds dans le graphe.

	théorique	résultat	itérations
nœuds actifs	46	56	8
nœuds inactifs	31	21	8
total	77	77	8

TABLE 7.10

Le graphe de synthèse obtenu est présenté dans la figure 7.19 : les nœuds constants identifiés pendant la préparation des données sont reliés par les arêtes de synthèse du GMD. Nous observons trois clusters clairement mis en évidence. L'identification des régions anatomiques concernées est facilitée grâce à la juxtaposition de deux vues – le graphe de synthèse et le graphe entier avec son layout anatomique 2D – et à l'utilisation de codes de couleur anatomiques – une couleur par zone englobant plusieurs régions de l'atlas AICHA.

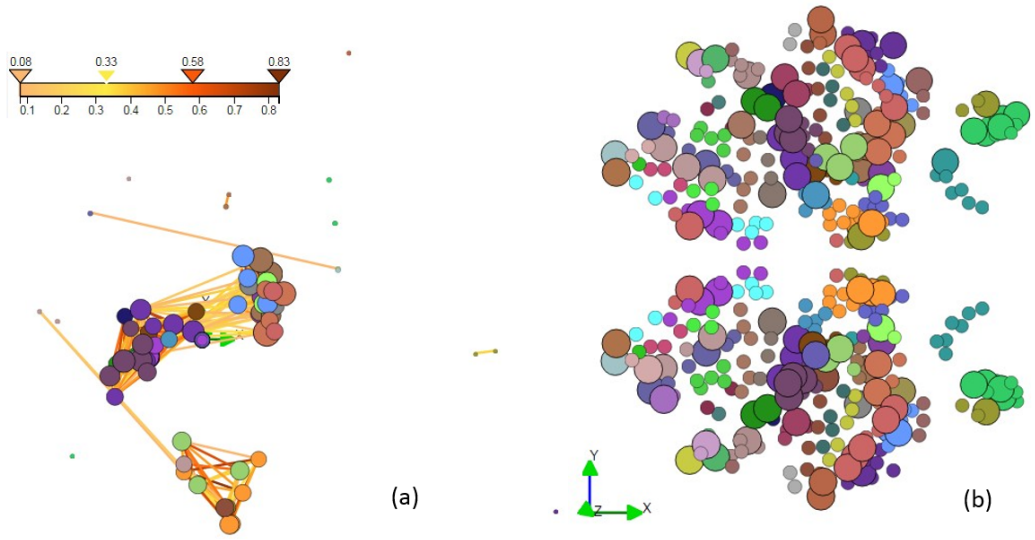


FIGURE 7.19 – Nœuds constants identifiés sur les quatre classes de genre et de latéralité : (a) Graphe de synthèse (la valeur du poids de synthèse est donnée dans l'échelle de couleur) (b) Layout anatomique (nœuds constants = grands nœuds). Les couleurs des nœuds représentent une segmentation anatomique.

La figure 7.20 montre les configurations de graphe obtenues pour chaque classe d'âge avec le layout LàC. A titre de comparaison, le layout "libre" calculé avec l'algorithme de force multi-niveaux OpenORD est également présenté pour chaque classe d'âge. De même que pour l'étude suivant l'âge, des couleurs aléatoires sont attribuées aux clusters des configurations de graphe afin d'aider à repérer les changements topologiques globaux d'un état à l'autre.

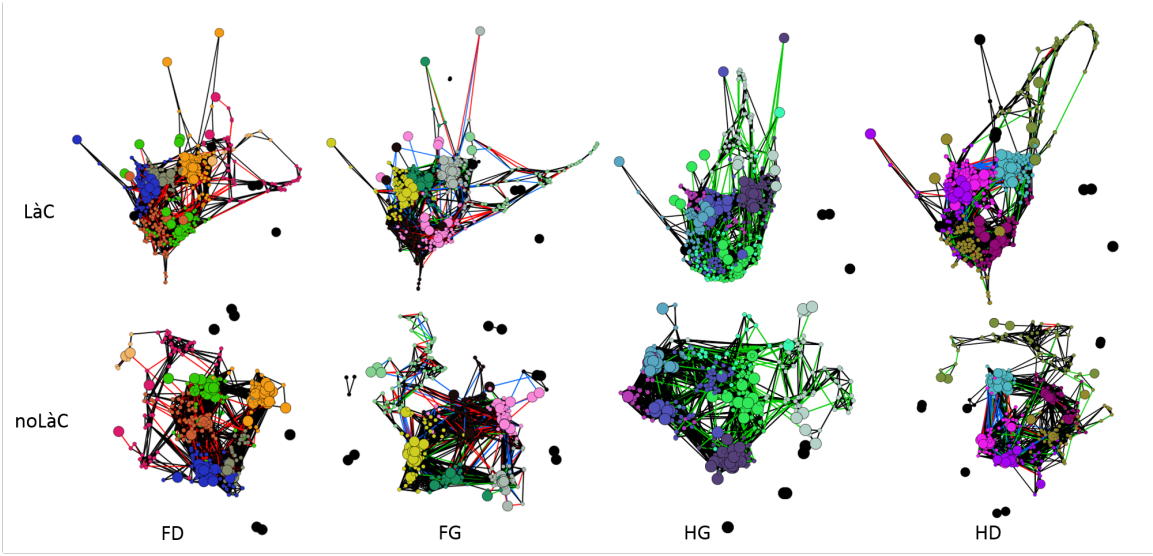
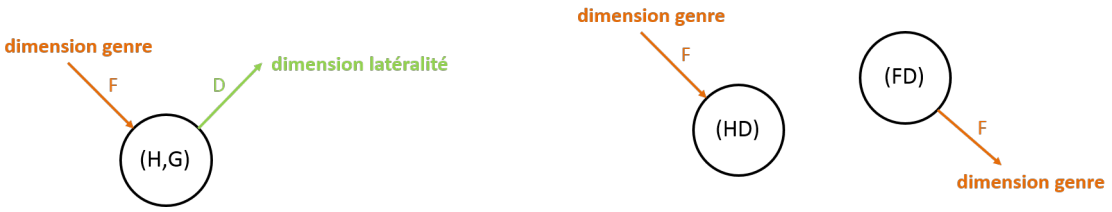


FIGURE 7.20 – Comparaison du layout avec (LàC) et sans contraintes (noLàC) pour les quatre classes de genre et de latéralité.

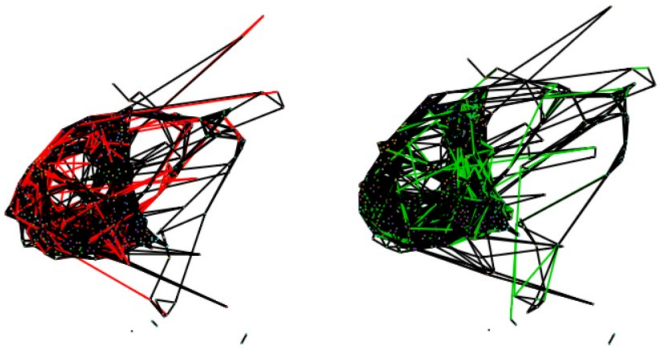
7.3.3.3 Résultats de l'exploration

Le GMD qui représente la connectivité fonctionnelle cérébrale en fonction du genre et de la latéralité possède deux dimensions. La représentation d'un homme gaucher *en contexte* est proposée dans la figure 7.21.(a). Avec plusieurs dimensions, il est utile de pouvoir interagir sur la ou les dimensions visualisées. Plutôt que de visualiser le contexte par rapport à toutes les dimensions, un utilisateur pourrait ne vouloir afficher que le contexte d'une dimension, comme illustré dans la figure 7.21.(b) : les droitiers en contexte du genre. Les arêtes *en contexte* pour la dimension genre (avec une latéralité fixée à "droitier") sont présentées avec le layout LàC dans la figure 7.22 dans une présentation small multiples ; la couleur des arêtes représente les évolutions des états adjacents.



(a) (b)

FIGURE 7.21 – Représentation d'un état *en contexte* : (a) exemple de contexte : un homme gaucher, (b) les contextes pour des droitiers.



État FD État HD

FIGURE 7.22 – Comparaison en contexte sur le genre pour une latéralité fixe (*droitier*). Rouge : l'arête disparait à l'état suivant, Vert : l'arête apparait à l'état courant.

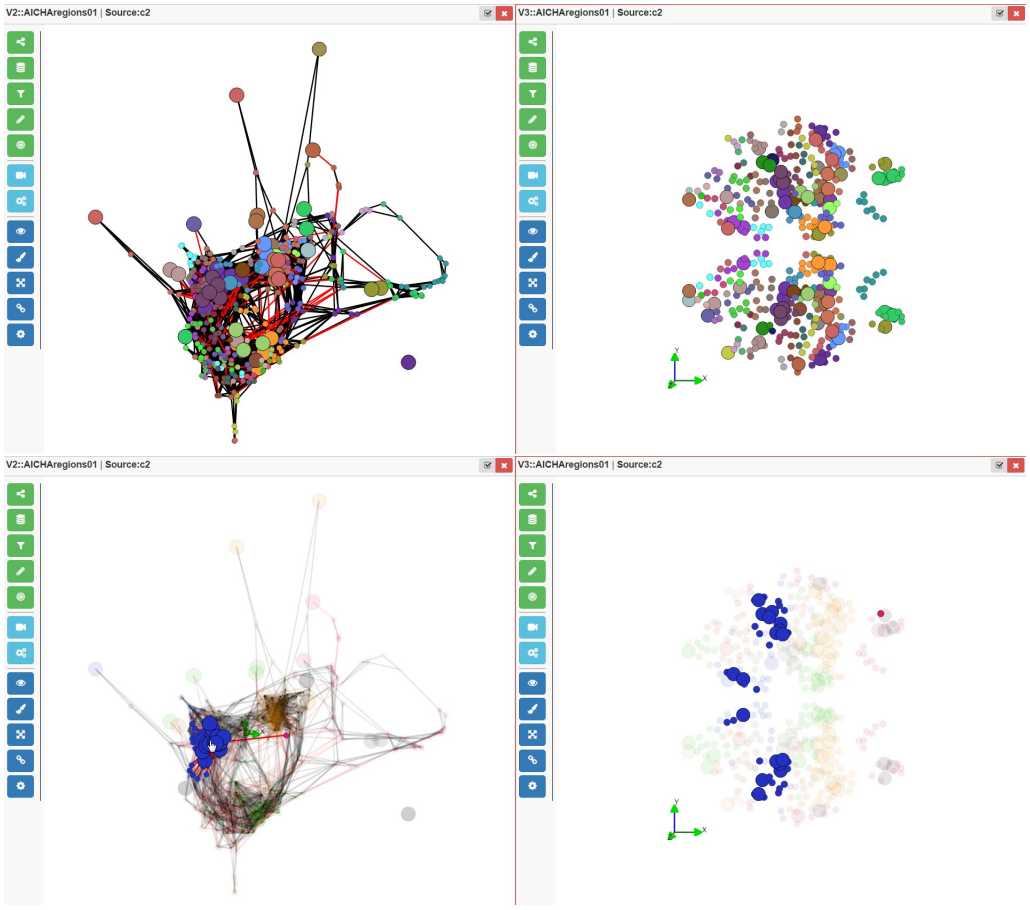


FIGURE 7.23 – Exemple d'exploration multi-vues : la classe de genre et de latéralité {FD} est affiché avec le layout LàC sur la gauche, et un layout anatomique 2D sur la droite. Les deux vues partagent des couleurs aléatoires associées aux clusters de la configuration de graphe. La sélection d'un nœud sur l'une des vues le met en exergue ainsi que les nœuds qui lui sont adjacents, et ce sur les deux vues.

7.3.4 Critique de la méthode OCL

La méthode OCL permet d'exploration des GMD grâce à la préservation partielle de la carte mentale de l'utilisateur qui permet de mettre en relief les éléments constants sur la ou les dimensions considérées. Le graphe de synthèse permet à l'utilisateur de comprendre la structure des nœuds fixes et d'aborder l'exploration interactive en connaissant déjà les structures topologiques principales du GMD. Le mode de visualisation *en contexte* permet de focaliser la visualisation sur un état et sur les changements qui se produisent à sa proximité (dimensionnelle), à manière d'un fisheye.

Nous n'avons pas pu expérimenter la méthode sur un panel étendu de testeurs, car les tâches d'exploration requièrent l'expertise de chercheurs en neuroimagerie sensibilisés à la visualisation de graphes, et un tel groupe est difficile à constituer.

La détermination des paramètres optimaux semble être un point difficile pour l'utilisateur, car il doit actuellement aborder cet aspect en dehors de l'interface de visualisation de graphes, de façon empirique, et apprécier les résultats de façon qualitative uniquement. Il nous faudra donc retravailler l'interface pour intégrer cette fonctionnalité et proposer un algorithme de détermination automatique des paramètres.

D'autre part, il manque au visualiseur SwoViewer des fonctionnalités d'interaction qui permettraient d'améliorer la méthode OCL, en facilitant l'étude locale d'une configuration de graphe, ainsi que l'étude de groupes de nœuds :

- La surbrillance du graphe de synthèse sur d'autres vues.
- Une lentille simple qui permet d'offrir à l'utilisateur un zoom local sans qu'il ait à modifier les paramètres d'affichage du graphe ou de la vue.
- La visualisation des arêtes d'une sélection (groupe) de nœuds.
- La surbrillance d'une sélection (groupe) de nœuds et de leurs arêtes communes entre plusieurs vues.
- Le calcul de layouts locaux au sein d'un groupe de nœud.

Conclusion du chapitre 7

Nous avons présenté dans ce chapitre l'application des propositions développées dans les chapitres 4, 5 et 6. Un système PLM pour la gestion des données de neuroimagerie a été implémenté au sein du laboratoire GIN : il permet aux chercheurs de gérer leurs données tout au long d'une étude, et de conserver automatiquement la provenance. La méthode OCL de visualisation de GMD a été testée dans l'interface du logiciel SwoViewer. Les algorithmes utilisés pour la préparation des données et l'analyse interactive ont été développés en java. La méthode permet d'explorer des GMD tout en préservant la carte mentale de l'utilisateur et en proposant l'alternance interactive de vues complètes et réduites des données. Les configurations de graphes sont mises en relief à la façon d'un fisheye dimensionnel.

La méthode OCL a été testée sur une partie du jeu de données BIL&GIN du groupe de recherche GIN, afin d'explorer la connectivité fonctionnelle cérébrale en fonction de l'âge, du

genre et de la latéralité.

La méthode OCL permet à l'utilisateur d'identifier les structures constantes tout au long des dimensions d'un GMD, afin de mieux analyser les changements entre états. Elle permet également d'explorer les données sous forme de vues interactives. Les principales critiques de la méthode OCL sont la difficulté de détermination des paramètres d'entrée de l'algorithme, et le manque de fonctionnalités cernant la sélection et la manipulation de groupes dans le visualiseur SwoViewer.

Dans la suite de notre manuscrit, le chapitre 8 conclut et discute les propositions de la thèse afin de présenter les perspectives de recherche futures.

Chapitre 8

Conclusion

8.1 Synthèse

Nous nous sommes intéressés dans notre thèse à l'exploration de relations complexes entre ensembles de données hétérogènes et multidimensionnels. La figure 8.1 synthétise la démarche et les résultats obtenus. Le manuscrit a été découpée en sept chapitres, dont nous rappelons le contenu.

Chapitre 1 Le cadre et le positionnement scientifique ont été présentés. Ce chapitre a permis d'exposer les questions de recherche de la thèse :

- Comment gérer les données hétérogènes et leur provenance ?
- Comment visualiser les structures de données multidimensionnelles et dynamiques ?

A partir de la comparaison des caractéristiques des domaines de l'industrie manufacturière et de la neuroimagerie, une première hypothèse ont été posée : les solutions de gestion des données développées pour l'industrie manufacturière peuvent être appliquées à la gestion des données en neuroimagerie.

Les domaines de recherche et d'application de la thèse ont été identifiés : ingénierie collaborative, gestion des données, visualisation de l'information, théorie des graphes et visualisation des réseaux biologiques.

Chapitre 2 Nous avons présenté les concepts liés à la gestion du cycle de vie du produit dans l'industrie manufacturière, ainsi que les apports et limites des systèmes PLM (Product Lifecycle Management). Ces derniers permettent d'apporter la bonne information à la bonne personne et au bon moment tout au long du cycle de vie du produit. Les interfaces des systèmes PLM rendent cependant difficile l'exploration de la provenance des données.

Les problématiques du domaine de la neuroimagerie en terme de partage des données ont été introduites, et les solutions de gestion des données développées pour la neuroimagerie ont été comparées. La pluridisciplinarité des études, le temps et les coûts d'acquisition et la reproduction des résultats incitent les chercheurs en neuroimagerie à partager les données existantes, cependant des limites techniques et sociales empêchent encore aujourd'hui le partage et la réuti-

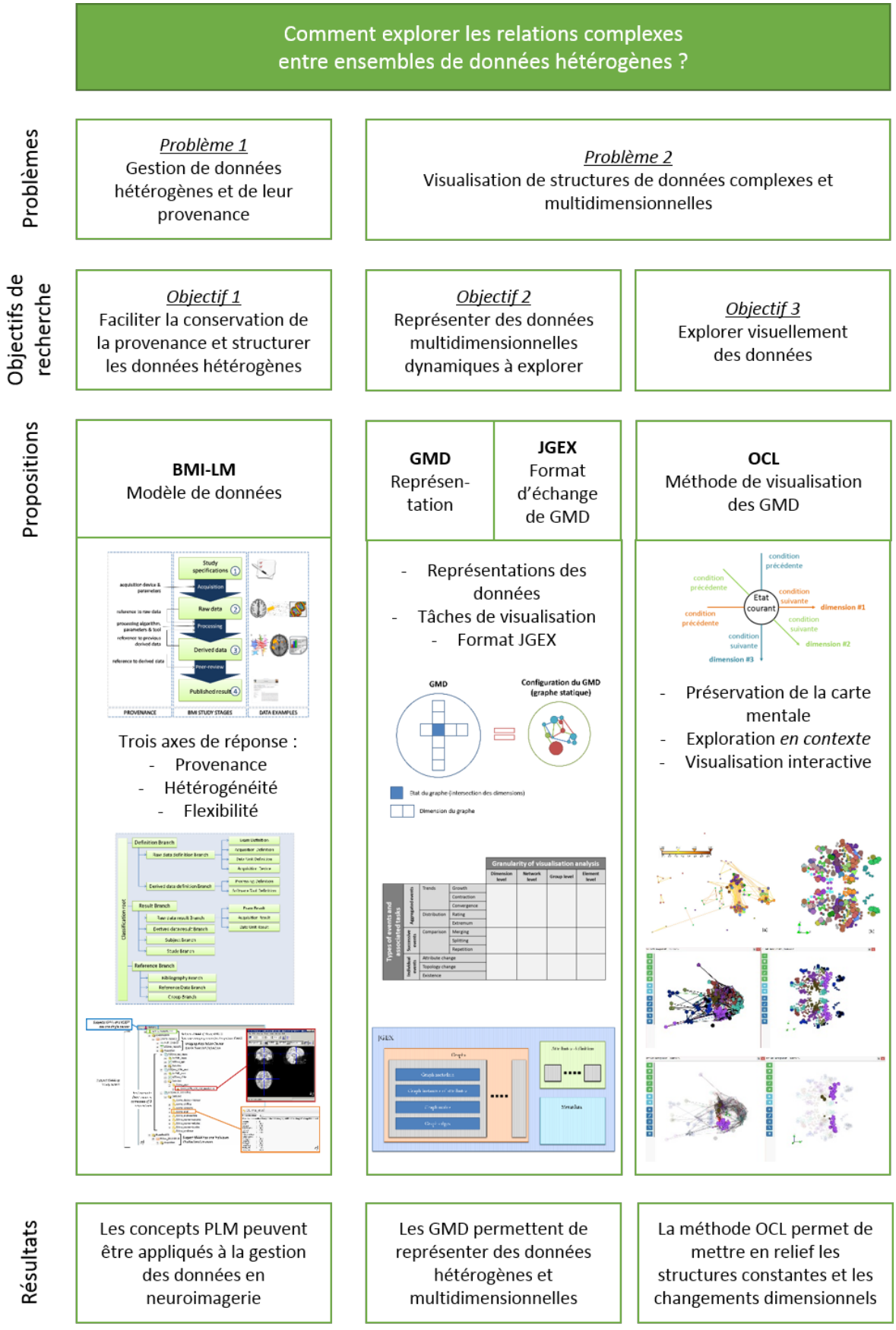


FIGURE 8.1 – Schéma de synthèse

lisation des données à grande échelle. Les systèmes de gestion des données développés pour la neuroimagerie présentent des limites en terme de gestion des données hétérogènes tout au long d'une étude, de gestion de la provenance et de flexibilité des modèles de données.

Nous avons identifié trois verrous à lever pour une gestion adaptée des données de neuroimagerie tout au long d'une étude pour faciliter leur partage et leur réutilisation :

- **Provenance** – gérer l'intégralité des données d'une étude depuis ses spécifications jusqu'aux publications et capturer la provenance associée, pour un partage et une réutilisation optimale des données.
- **Hétérogénéité** – accepter tous les formats de données et gérer les concepts de plusieurs disciplines.
- **Flexibilité** –permettre les évolutions du modèle de données sans conséquences sur les données déjà présentes dans la base de données.

Chapitre 3 Dans ce chapitre nous avons introduit les concepts de la théorie des graphes ainsi que les techniques de visualisation de graphes statiques et dynamiques. La technique la plus adaptée dépend des tâches à effectuer par l'utilisateur. Le rôle de la carte mentale dans la visualisation de graphes dynamiques a été discuté, et plusieurs études montrent que sa préservation d'un moment temporel sur l'autre pourrait aider à l'exploration de graphes dynamiques.

L'interaction est nécessaire à l'exploration de données complexes. L'utilisateur peut interagir avec la visualisation selon trois axes : les données elles-mêmes, la représentation des données et la vue des données. Les taxonomies des tâches en visualisation servent à faciliter le travail des concepteurs de techniques de visualisation, en particulier pour la composante interactive.

L'étude de la bibliographie nous a amené à formuler **deux objectifs** pour résoudre le problème 2 :

1. **Quelle représentation** pour les données hétérogènes et complexes (multidimensionnelles, hiérarchiques, évolutives) ?
2. **Comment explorer** et donc interagir avec ces données ?

Il n'existe pas actuellement de représentation sous forme de graphe des données que nous cherchons à explorer. Quelle forme conceptuelle pourrait prendre ce graphe ? Quelle structure de graphe créer qui s'appuierait sur les structures existantes (arbres, graphes génériques statiques et dynamiques, graphes composés...) ?

Pour explorer visuellement la nouvelle structure de graphe, nous avons choisi de nous appuyer sur les représentations visuelles dynamiques et les techniques d'interaction existantes. A l'étude de l'état de l'art, les diagrammes node-link et les techniques small multiples semblent de bonnes pistes. Cependant, nous avons identifié **trois verrous à lever** pour permettre l'exploration de données multidimensionnelles et dynamiques :

- **Stabilité du layout** – Comment préserver la carte mentale tout en mettant en exergue les changements ?

– **Stabilité des données** – Comment mettre en évidence des structures constantes et des structures variables ?
– **Interaction** – Quel processus d'interaction pour aborder les données ?

Le chapitre 3 nous a également permis d'introduire le cas d'application de notre thèse, la connectivité fonctionnelle cérébrale en neuroimagerie, que les chercheurs en neuroimagerie étudient à l'aide de la théorie des graphes et de techniques du domaine de la visualisation d'informations.

Chapitres 4,5,6 Les problèmes de recherche établis dans le chapitre 1 ont été segmentés en trois objectifs, qui sont chacun abordés dans un chapitre de notre thèse.

Le chapitre 4 s'est intéressé à l'objectif n°1 : *faciliter la conservation de la provenance et structurer les données hétérogènes*. Nous avons proposé le **modèle de données Bio-Medical Imaging – Lifecycle Management (BMI-LM)** pour la gestion des données en neuroimagerie. Il s'articule autour de trois axes principaux :
– *provenance* garantie pour l'ensemble des étapes d'une étude de recherche et entre les études,
– gestion de l'*hétérogénéité* des données au sein de concepts génériques,
– *flexibilité* du modèle de données à l'aide de classes pour permettre l'évolution de l'organisation des données au grès des avancées de la recherche.

Dans le chapitre 5 sont présentées nos propositions relatives à l'objectif n°2 : *représenter des données multidimensionnelles et dynamiques à explorer*. Les **Graphes Multidimensionnels Dynamiques (GMD)** ont été définis ; ils servent à représenter des relations complexes entre des données, notamment des données qui évoluent selon plusieurs dimensions. Le vocabulaire associé aux GMD a été précisé, et une **taxonomie des tâches visuelles** à été développée pour aider à concevoir dans le futur des méthodes d'exploration visuelle adaptées aux spécificités des GMD. Afin de permettre l'échange et le stockage de GMD, par exemple entre une base de données PLM et une interface de visualisation de graphes, **le format JSON Graph EXchange (JGEX)** a été présenté.

Le chapitre 6 a porté sur l'objectif n°3 : *explorer visuellement des données*. La méthode **Overview Constaint Layout** (OCL) a été développée pour l'exploration des GMD. Elle s'articule autour d'une *persistance partielle de la carte mentale* sur l'ensemble des états du GMD grâce à l'algorithme de **Layout à Contraintes (LàC)**, d'une visualisation *en-contexte* basée sur le principe du fisheye et l'alternance *interactive* de vues complètes et réduites.

Chapitre 7 Les propositions développées dans notre thèse ont été appliquées à l'étude de la connectivité fonctionnelle cérébrale.

Un système PLM pour la gestion des données de neuroimagerie a été implémenté au sein du laboratoire GIN : il permet aux chercheurs de gérer leurs données tout au long d'une étude, et

de conserver automatiquement la provenance. Ce résultat confirme notre hypothèse de départ selon laquelle les solutions de gestion des données développées pour l'industrie manufacturière peuvent être appliquées à la gestion des données en neuroimagerie.

La méthode OCL de visualisation de GMD a été testée dans l'interface du logiciel Swo-Viewer. La méthode permet de mettre en relief les structures constantes et les changements dimensionnels pour explorer des GMD. D'autre part les GMD se sont révélés adaptés pour la représentation de données hétérogènes et multidimensionnelles.

8.2 Discussion

8.2.1 Gestion des données et de leur provenance

Acquisition de données brutes L'utilisation d'un système PLM associé au modèle de données BMI-LM pour gérer les données de neuroimagerie a uniquement été testée avec un jeu de données pré-existant issu du laboratoire GIN. La base données pour la neuroimagerie installée au GIN est actuellement connectée à une grille de calculs locale pour l'exécution de chaînes de traitements de façon automatisée. Cependant, elle n'est pas encore intégrée avec des appareils externes d'acquisition de données brutes, comme une machine IRM pour les données d'imagerie ou une tablette notamment pour le recueil des données sujets et des tests de psychologie. Il reste donc à tester l'installation en contexte d'import de données directement depuis la source d'acquisition. L'utilisation d'un système PLM pour gérer des données nouvellement acquises est planifié dans l'étude *I-Share*[1] fin 2015.

Publications scientifiques liées à une étude Le modèle de données BMI-LM permet de gérer les articles scientifiques, à la fois comme référence de données de définition (protocoles d'acquisition, algorithmes de traitement, etc) et comme publication de résultats à la suite d'une étude. Ce dernier aspect permettrait de garantir la reproductibilité des résultats scientifiques, puisque tous les traitements effectués sur la données brutes ont été tracés et identifiés depuis la phase d'acquisition. Actuellement des publications sont gérées dans le système PLM installé au GIN, mais elles servent uniquement de référence à des protocoles. La gestion des nouvelles publications va être testée dans un futur proche, notamment dans le cadre de l'étude *I-Share*.

Partage Une autre limitation des travaux présentés dans ce manuscrit concerne le partage public des données. Les règles d'accès sur chaque étude et dépendant du groupe et du rôle des utilisateurs sont administrés grâce au module de gestion des accès du PLM Teamcenter, ce qui facilite la collaboration au sein d'une équipe et entre partenaires, mais pas au sein d'une communauté de chercheurs. Bien que les clients web pour le PLM existent, une licence et un apprentissage minimal sont nécessaires, ce qui empêche les utilisations occasionnelles du système. Les bases de données en accès ouvert sur internet sont un standard implicite dans le domaine de la neuroimagerie depuis la première base de données du projet HBP en 1993.

1. http://www.i-share.fr/

Il apparait donc indispensable de proposer dans le futur une plateforme en ligne de la base de données PLM plus adaptée à la neuroimagerie . Dans une première étape des travaux de recherche présentés dans notre manuscrit, un faible nombre d'utilisateurs ont participé à la clarification des besoins et aux tests sur la base PLM. Les systèmes PLM ont été validés pour la gestion de données en neuroimagerie. A présent les futurs travaux doivent porter sur les spécifications d'un client web dédié qui pourrait se connecter à d'autres bases de données de neuroimagerie, notamment grâce à l'emploi d'ontologies du domaine. Une tendance émergente du domaine est de permettre le partage entre différents systèmes de bases de données à travers la médiation, grâce à l'utilisation d'ontologies de mapping (Pham *et al.*, 2015). Il apparait pertinent de prévoir le moyen de connecter les systèmes PLM avec à des bases de données externes de type XNAT ou encore PubMed pour relier la gestion bibliographique telle que défnie dans le modèle de données BMI-LM avec la bibliographie la plus complète du monde médical. Plusieurs travaux sur le sujet constituent une base de réflexion intéressante, telle que l'initiative FBIRN (Ashish *et al.*, 2010). Des ontologies ont été utilisées pour aider à la conception de la classification pour la neuroimagerie, cependant un enrichissement sémantique plus conséquent pourrait améliorer les relations entre objets dans les systèmes PLM (Assouroko *et al.*, 2014), en particulier pour les fonctionnalités de recherche de données.

Interface Le retour des utilisateurs du GIN a mis en avant que l'interface utilisateur du système PLM n'est adaptée à leurs besoins, à cause d'un nombre excessif de fenêtres et de menus qui empêche un usage implicite du système. Par conséquent, il apparait nécessaire de développer dans le futur une interface spécifique, et qui serait disponible sur le web. Les clients web deviennent une évolution naturelle des systèmes PLM dans l'industrie manufacturière afin de pouvoir accéder aux données de partout et avec le support de nouveaux médiums comme les tablettes. Des clients PLM web intéressants ont été développés, mais ils proposent toujours une interface surchargée – par exemple ARAS ou Windchill – et complexe – Active Workspace –, ce qui est inadapté à une utilisation occasionnelle. Les relations entre objets sont difficiles à appréhender, considérant la nature même du travail de recherche en neuroimagerie. La navigation dans les données est par conséquent critique. Cependant, les systèmes PLM actuels ne proposent pas de module satisfaisant pour l'exploration des relations. Par conséquent, un intérêt majeur dans les travaux de recherche à venir est la visualisation des relations, sous forme de graphe, afin d'améliorer l'exploration des données et de leur provenance. Les concepts du PLM sont appliqués à un nombre croissant de nouveaux domaines, la communauté du PLM devrait profiter de cette opportunité pour faire évoluer les fonctionnalités des systèmes PLM, en particulier du point de vue de leur interface.

Maintenance La maintenance d'un système PLM demande une expertise et des ressources. C'est également le cas d'autres systèmes de gestion des données du domaine de la neuroimagerie. Si les bénéfices sont démontrés sur une étude à grande échelle, l'investissement humain et financier devient envisageable sur un groupement de laboratoires et d'institutions.

Application à d'autres domaines Parce que le modèle de données BMI-LM est générique et suffisamment flexible pour être facilement étendu, l'approche présentée dans notre thèse pourrait être utilisée par d'autres domaines présentant des données hétérogènes et complexes. Par exemple, la mécatronique est un domaine combinant mécanique, électronique, automatique et informatique. La conception d'un système mécatronique est complexe à cause des interactions multi-physiques entre les différents composants (Lefèvre *et al.*, 2012). Par conséquent une approche similaire à celle présentée dans notre manuscrit pourrait être intéressante pour la communauté mécatronique (Bricogne *et al.*, 2012). Jusqu'à présent, la plupart des travaux de recherche dans la communauté PLM ont porté sur la première étape du cycle de vie des produits (BOL). Cependant, un mécatronique système – par exemple un contrôleur industriel – va générer durant sa vie (MOL) des données hétérogènes à travers des interactions physiques et numériques – par exemple une boucle de contrôle entre des capteurs physiques, une unité de calcul et un actionneur physique. Ces données peuvent être stockées dans un système de gestion des données, potentiellement un système PLM pour tracer la provenance des actions réalisées par les systèmes mécatroniques. Il apparait intéressant d'appliquer la démarche présentée dans cette thèse à d'autres domaines aux problématiques similaires à la neuroimagerie, comme la mécatronique. L'ouverture du PLM à de nouveaux domaines d'application promet de nouveaux marchés pour les entreprises qui gravitent autour de la technologie (éditeurs, consultants, etc).

La Plateforme d'Imagerie du Petit Animal (PIPA) [2] de l'Université Paris Descartes regroupe les données hétérogènes d'imagerie de plusieurs laboratoires (imagerie par résonance magnétique, imagerie ultrasonore, imagerie optique de bioluminescence et de fluorescence, imagerie scanner X et imagerie par résonance paramagnétique électronique), ce qui implique des besoins en gestion du partage et de la réutilisation de données entre plusieurs laboratoires. Le projet *DRIVE* de Sorbonne-Paris-Cité prévoit de déployer la solution PLM développée dans le projet BIOMIST en 2016 au sein du Réseau d'Images du Vivant pour les plateformes d'imagerie Experimentale pour gérer les données de la plateforme PIPA.

8.2.2 Exploration visuelle de données multidimensionnelles dynamiques

Préparation des données La détermination des paramètres optimaux semble être un point difficile pour l'utilisateur, car il doit actuellement aborder cet aspect en dehors de l'interface de visualisation de graphes, de façon empirique, et apprécier les résultats de façon qualitative uniquement. Il nous faudra donc retravailler l'interface pour intégrer cette fonctionnalité et proposer un algorithme de détermination automatique des paramètres.

Préservation de la carte mentale L'intérêt du LàC pour les performances de l'utilisateur à réaliser des tâches complexes n'a pas pu être prouvé. Pour le moment, seules des appréciations qualitatives ont été effectuées sur la méthode OCL. Il faudrait mesurer les performances d'un groupe de personnes lors de l'exploration d'un GMD avec deux ou trois dimensions. Il serait également intéressant de déterminer avec plus de précision les pourcentages de nœuds fixes

2. http://piv.parisdescartes.fr/

idéals.

Méthode extensible La méthode OCL n'a pu être testée que sur un jeu de données GMD de petite taille (quatre états). Il faudra tester de nouveau la méthode sur un jeu de données présentant davantage d'états (une dizaine ou plusieurs dizaines), afin de vérifier si la méthode est extensible (scalable) malgré le changement d'échelle. En effet, la détermination des nœuds constants pourrait être biaisée par de trop grandes différences entre les états, et il faudra peut-être songer à faire évoluer la méthode afin de rendre la définition de la stabilité plus souple, par exemple en introduisant des critères qu'un nœud ne devra pas respecter sur la totalité des états, mais sur un nombre minimal d'états.

Interaction Il existe trois niveaux d'abstraction de la visualisation avec lesquels un utilisateur peut interagir : les paramètres visuels, les schémas visuels et le filtrage des données (Von Landesberger *et al.*, 2011). Nous avons testé la méthode OCL dans le visualiseur de graphes SwoViewer ; l'interface propose d'interagir avec les paramètres visuels et le filtrage des données, mais très peu avec les schémas visuels. En particulier, il n'est pas possible de définir un schéma visuel pour un groupe de nœuds. Dans un futur proche, nous travaillerons à développer des fonctionnalités autour de l'interaction avec des groupes de nœuds (sous-graphes) dans le cadre de l'étude locale d'une configuration de graphe :

- Une lentille simple qui permet d'offrir à l'utilisateur un zoom local sans qu'il ait à modifier les paramètres d'affichage du graphe ou de la vue.
- La visualisation des arêtes d'une sélection de nœuds.
- La surbrillance d'une sélection de nœuds et de leurs arêtes communes entre plusieurs vues.
- L'application aux groupes d'un schéma visuel indépendant.

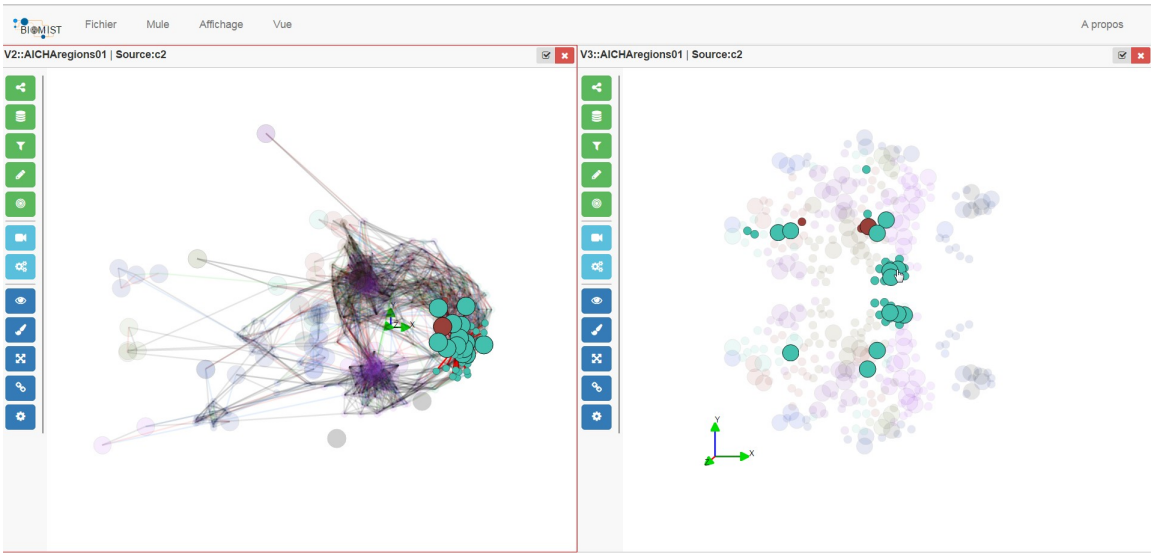


FIGURE 8.2 – Fisheye des arêtes et des nœuds adjacents d'un nœud sélectionné, avec propagation dans une autre vue du graphe dans SwoViewer. Il serait intéressant de propager ce type de fonctionnalité à des groupes de nœuds définis par l'utilisateur.

Application à d'autres domaines Dans le cadre de notre thèse, nous avons appliqué nos propositions pour la visualisation de structures de données multidimensionnelles et dynamiques à l'étude de la connectivité fonctionnelle cérébrale. Nous avons ouvert ce manuscrit sur la difficulté de trouver les informations pertinentes dans le monde actuel submergé par les données. Le monde est dynamique, multivarié et profondément relationnel : les réseaux sont partout. Nous pensons donc que la démarche présentée dans notre thèse pourrait être réutilisée dans de nombreux domaines qui présentent des caractéristiques identiques à celles de la neuroimagerie. En particulier, nous avons relevé dans notre thèse les limites des interfaces des systèmes PLM en terme de navigation. Un des axes futurs de nos travaux sera de proposer un moyen de visualiser de la provenance complexe, comme par exemple les BOM produit, qui sont des structures hiérarchiques naturellement évolutives.

Un autre exemple d'application est la visualisation d'information pour l'aide à la décision. Le logiciel *SwoDir*[3] propose de naviguer dans plusieurs sources d'information – CRM (Customer Relationship Management), réseaux sociaux professionnels, bases de données sur des entreprises, etc – sous forme de graphes. L'utilisateur n'a plus besoin de se connecter à plusieurs bases de données pour chercher des informations sur une sociétés ou une personne, car le logiciel requête lui-même les sources de données et les présente au sein d'une même interface après avoir réalisé les jointures appropriées. Une application permet de suivre l'évolution des données dans le temps, par exemple les relations Des copies d'écran du logiciel SwoDir sont données à titre d'illustration dans la figure 8.3 ; les vues sont centrées sur des nœuds sélectionnés par l'utilisateur, et les relations sont affichées sur plusieurs niveaux en fonction de leur nature.

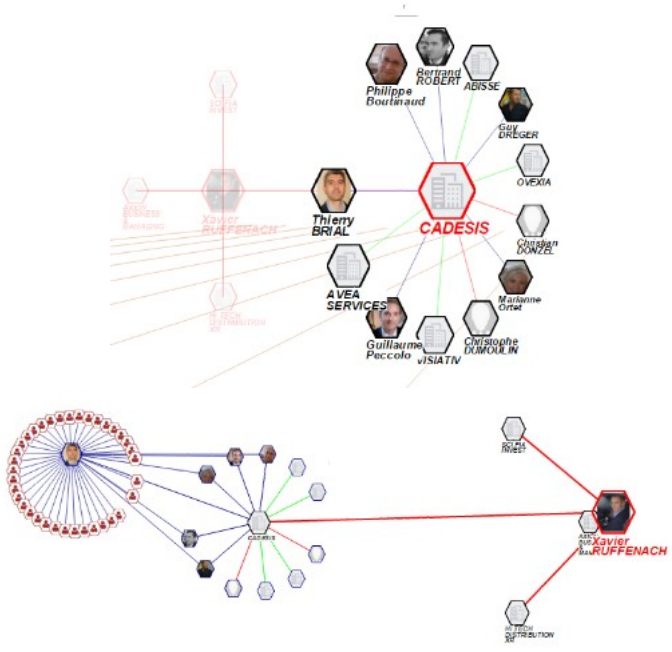


FIGURE 8.3 – Copies d'écran du logiciel SwoDir. Les nœuds à partir desquels sont centrées les vues sont entourés de rouge. La couleur des arêtes et la profondeur du layout indiquent la nature des relations.

3. Logiciel en cours de développement par la société CADESIS.

Bibliographie

Abello, James, Kobourov, Stephen G, & Yusufov, Roman. 2005. Visualizing large graphs with compound-fisheye views and treemaps. *Pages 431–441 of : Graph Drawing*. Springer.

Abello, James, Archambault, Daniel, Kennedy, Jessie, Kobourov, SG, Ma, Kwan Liu, Miksch, Silvia, Muelder, Chris, & Telea, Alexandru. 2014. Temporal multivariate networks. *Multivariate Network Visualization*, 151–175.

Adamson, Christopher L, & Wood, Amanda G. 2010. DFBIdb : a software package for neuroimaging data management. *Neuroinformatics*, **8**(4), 273–284.

Ahlberg, Christopher, Williamson, Christopher, & Shneiderman, Ben. 1992. Dynamic queries for information exploration : An implementation and evaluation. *Pages 619–626 of : Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM.

Ahn, Jae-wook, Plaisant, Catherine, & Shneiderman, Ben. 2014. A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics*, **20**(3), 365–376.

Aigner, Wolfgang, Miksch, Silvia, Schumann, Heidrun, & Tominski, Christian. 2011. *Visualization of time-oriented data*. Springer Science & Business Media.

Alper, Basak, Bach, Benjamin, Henry Riche, Nathalie, Isenberg, Tobias, & Fekete, Jean-Daniel. 2013. Weighted graph comparison techniques for brain connectivity analysis. *Pages 483–492 of : Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.

Alstott, Jeffrey, Breakspear, Michael, Hagmann, Patric, Cammoun, Leila, & Sporns, Olaf. 2009. Modeling the impact of lesions in the human brain. *PLoS Comput Biol*, **5**(6), e1000408–e1000408.

Amar, Robert, Eagan, James, & Stasko, John. 2005. Low-level components of analytic activity in information visualization. *Pages 111–117 of : IEEE Symposium on Information Visualization, INFOVIS 2005*. IEEE.

Ameri, Farhad, & Dutta, Deba. 2005. Product Lifecycle Management : Closing the Knowledge Loops. *Computer-Aided Design & Application*, **2**(5), 577–590.

Anscombe, Francis J. 1973. Graphs in statistical analysis. *The American Statistician*, **27**(1), 17–21.

Archambault, Daniel. 2009. Structural differences between two graphs through hierarchies. *Pages 87–94 of : Proceedings of Graphics Interface 2009*. Canadian Information Processing Society.

Archambault, Daniel, & Purchase, Helen C. 2012. The mental map and memorability in dynamic graphs. *Pages 89–96 of : Proceedings of Pacific Visualization Symposium (PacificVis 2012)*. IEEE.

Archambault, Daniel, & Purchase, Helen C. 2013a. The "map" in the mental map : Experimental results in dynamic graph drawing. *International Journal of Human-Computer Studies*, **71**(11), 1044–1055.

Archambault, Daniel, & Purchase, Helen C. 2013b. Mental map preservation helps user orientation in dynamic graphs. *Pages 475–486 of : Graph Drawing*. Springer.

Archambault, Daniel, Purchase, Helen C, & Pinaud, Bruno. 2011a. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, **17**(4), 539–552.

Archambault, Daniel, Purchase, Helen C, & Pinaud, Bruno. 2011b. Difference map readability for dynamic graphs. *Graph drawing*, 50–61.

Ascoli, Giorgio A. 2006. The ups and downs of neuroscience shares. *Neuroinformatics*, **4**(3), 213–215.

Ashish, Naveen, Ambite, José Luis, Muslea, Maria, & Turner, Jessica A. 2010. Neuroscience data integration through mediation : an (F) BIRN case study. *Frontiers in neuroinformatics*, **4**.

Assouroko, Ibrahim. 2012. *Gestion de données et dynamique des connaissances en ingénierie numérique – contribution à l'intégration de l'ingénierie des exigences, de la conception numérique et de la simulation numérique*. Ph.D. thesis, Université de Technologie de Compiègne.

Assouroko, Ibrahim, Ducellier, Guillaume, Eynard, Benoît, & Boutinaud, Philippe. 2012. Semantic relationship knowledge management and reuse in collaborative product development. *9th International Conference on Product Lifecycle Management, 9th-11th July 2012, Québec*.

Assouroko, Ibrahim, Ducellier, Guillaume, Boutinaud, Philippe, & Eynard, Benoît. 2014. Knowledge management and reuse in collaborative product development–a semantic relationship management-based approach. *International Journal of Product Lifecycle Management*, **7**(1), 54–74.

Auber, David. 2004. Tulip—A huge graph visualization framework. *Graph Drawing Software*, 105–126.

Bach, Benjamin. 2014. *Connections, changes, and cubes : unfolding dynamic networks for visual exploration.* Ph.D. thesis, Université Paris 11.

Bach, Benjamin, Pietriga, Emmanuel, & Fekete, Jean-Daniel. 2014a. GraphDiaries : animated transitions andtemporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, **20**(5), 740–754.

Bach, Benjamin, Dragicevic, Pierre, Archambault, Daniel, Hurter, Christophe, & Carpendale, Sheelagh. 2014b. A review of temporal data visualizations based on space-time cube operations. *In : Eurographics Conference on Visualization (EUROVIS 2014).*

Barillot, Christian, Bannier, Elise, Commowick, Olivier, Corouge, Isabelle, Guillaumont, Justine, Yao, Yao, & Kain, Michael. 2015. Shanoir : Software as a Service Environment to Manage Population Imaging Research Repositories. *Page 23 of : 1 st MICCAI Workshop on MAnagement and Processing of images for Population ImagiNG.*

Barsky, Aaron, Gardy, Jennifer L, Hancock, Robert EW, & Munzner, Tamara. 2007. Cerebral : a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. *Bioinformatics*, **23**(8), 1040–1042.

Bastian, Mathieu, Heymann, Sebastien, Jacomy, Mathieu, *et al.* 2009. Gephi : an open source software for exploring and manipulating networks. *ICWSM*, **8**, 361–362.

Batagelj, Vladimir, & Mrvar, Andrej. 1998. Pajek-program for large network analysis. *Connections*, **21**(2), 47–57.

Beck, Fabian, Burch, Michael, & Diehl, Stephan. 2009. Towards an aesthetic dimensions framework for dynamic graph visualisations. *Pages 592–597 of : 13th International Conference on Information Visualisation, 2009.* IEEE.

Beck, Fabian, Burch, Michael, Diehl, Stephan, & Weiskopf, Daniel. 2014. The State of the Art in Visualizing Dynamic Graphs. *In : EuroVis STAR, 2014 9th-13th June.*

Beck, Fabio, Burch, Michel, & Diehl, Stephan. 2013. Matching application requirements with dynamic graph visualization profiles. *Pages 11–18 of : 17th International Conference on Information Visualisation (IV).* IEEE.

Becker, Stefanie I, Harris, Anthony M, Venini, Dustin, & Retell, James D. 2014. Visual search for color and shape : When is the gaze guided by feature relationships, when by feature values ? *Journal of Experimental Psychology : Human Perception and Performance*, **40**(1), 264.

Belkadi, Farouk, Troussier, Nadège, Eynard, Benoit, & Bonjour, Eric. 2010. Collaboration based on product lifecycles interoperability for extended enterprise. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, **4**(3), 169–179.

Bennett, Chris, Ryall, Jody, Spalteholz, Leo, & Gooch, Amy. 2007. The aesthetics of graph visualization. *Pages 57–64 of : Proceedings of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Eurographics Association.

Benson, D a, Boguski, M S, Lipman, D J, Ostell, J, Ouellette, B F, Rapp, B a, & Wheeler, D L. 1999. GenBank. *Nucleic acids research*, **27**(1), 12–7.

Benson, DA, Karsch-Mizrachi, I, Lipman, DJ, Ostell, J, & Sayers, EW. 2010. GenBank. *Nucleic acids research*, **38**(suppl 1), D46–D51.

Bertin, Jacques. 1973. Sémiologie graphique : Les diagrammes-Les réseaux-Les cartes.

Bezerianos, Anastasia, Chevalier, Fanny, Dragicevic, Pierre, Elmqvist, Niklas, & Fekete, Jean-Daniel. 2010. Graphdice : A system for exploring multivariate social networks. *Computer Graphics Forum*, **29**(3), 863–872.

Birney, Ewan, Hudson, Thomas J, Green, Eric D, Gunter, Chris, Eddy, Sean, Rogers, Jane, Harris, Jennifer R, Ehrlich, S Dusko, Apweiler, Rolf, Austin, Christopher P, *et al.* 2009. Prepublication data sharing. *Nature*, **461**(7261), 168–170.

Blessing, Lucienne TM, & Chakrabarti, Amaresh. 2009. *DRM : A Design Reseach Methodology*. Springer.

Blythe, Jim, McGrath, Cathleen, & Krackhardt, David. 1996. The effect of graph layout on inference from social network data. *Pages 40–51 of : Graph Drawing*. Springer.

Bondy, John Adrian, & Murty, Uppaluri Siva Ramachandra. 1976. *Graph theory with applications*. Macmillan London.

Book, Gregory A, Anderson, Beth M, Stevens, Michael C, Glahn, David C, Assaf, Michal, & Pearlson, Godfrey D. 2013. Neuroinformatics Database (NiDB)–a modular, portable database for the storage, analysis, and sharing of neuroimaging data. *Neuroinformatics*, **11**(4), 495–505.

Boothroyd, Geoffrey. 1994. Product design for manufacture and assembly. *Computer-Aided Design*, **26**(7), 505–520.

Borgatti, Stephen P, Everett, Martin G, & Freeman, Linton C. 2002. Ucinet for Windows : Software for social network analysis.

Boujut, Jean-François, & Blanco, Eric. 2003. Intermediary objects as a means to foster cooperation in engineering design. *Computer Supported Cooperative Work (CSCW)*, **12**(2), 205–219.

Bourqui, Romain, & Jourdan, Fabien. 2008. Revealing subnetwork roles using contextual visualization : comparison of metabolic networks. *Pages 638–643 of : 12th International Conference on Information Visualisation (IV'08)*. IEEE.

Boutang, Yann Moulier. 2007. *Le capitalisme cognitif : la nouvelle grande transformation.* Amsterdam.

Brehmer, Matthew, & Munzner, Tamara. 2013. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, **19**(12), 2376–2385.

Bricogne, Matthieu, Rivest, Louis, Troussier, Nadège, & Eynard, Benoît. 2012. Towards PLM for mechatronics system design using concurrent software versioning principles. *Product Lifecycle Management – Towards Knowledge-Rich Enterprises*, 339–348.

Brown, Jesse a., Rudie, Jeffrey D., Bandrowski, Anita, Van Horn, John D., & Bookheimer, Susan Y. 2012. The UCLA multimodal connectivity database : a web-based platform for brain connectivity matrix sharing and analysis. *Frontiers in Neuroinformatics*, **6**(November), 1–17.

Buckow, Karoline, Quade, Matthias, Rienhoff, Otto, & Nussbeck, Sara Y. 2014. Changing requirements and resulting needs for IT-infrastructure for longitudinal research in the neurosciences. *Neuroscience research.*

Bullmore, Ed, & Sporns, Olaf. 2009. Complex brain networks : graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, **10**(3), 186–198.

Bullmore, Edward T, & Bassett, Danielle S. 2011. Brain graphs : graphical models of the human brain connectome. *Annual review of clinical psychology*, **7**, 113–140.

Burch, Michael, & Diehl, Stephan. 2008. TimeRadarTrees : Visualizing dynamic compound digraphs. *Computer Graphics Forum*, **27**(3), 823–830.

Chen, Peng, Plale, Beth, Cheah, Y, Ghoshal, Devarshi, Jensen, Soren, & Luo, Yuan. 2012. Visualization of network data provenance. *Pages 1–9 of : High Performance Computing (HiPC), 2012 19th International Conference on.* IEEE.

Chen, Yuh-Min, & Jan, Yann-Daw. 2000. Enabling allied concurrent engineering through distributed engineering information management. *Robotics and Computer-Integrated Manufacturing*, **16**(1), 9–27.

Chi, Ed H. 2000. A taxonomy of visualization techniques using the data state reference model. *Pages 69–75 of : Information Visualization, 2000. InfoVis 2000. IEEE Symposium on.* IEEE.

Coleman, Michael K, & Parker, D Stott. 1996. Aesthetics-based Graph Layout for Human Consumption. *Software : Practice and Experience*, **26**(12), 1415–1438.

Collins, Francis S, & Tabak, Lawrence A. 2014. NIH plans to enhance reproducibility. *Nature*, **505**(7485), 612.

Conte, Donatello, Foggia, Pasquale, Sansone, Carlo, & Vento, Mario. 2004. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, **18**(03), 265–298.

Cox, Richard. 1999. Representation construction, externalised cognition and individual differences. *Learning and instruction*, **9**(4), 343–363.

Das, Samir, Zijdenbos, Alex P, Harlap, Jonathan, Vins, Dario, & Evans, Alan C. 2011. LORIS : a web-based data management system for multi-center studies. *Frontiers in neuroinformatics*, **5**, 37.

Davidson, Ron, & Harel, David. 1996. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics (TOG)*, **15**(4), 301–331.

Debaecker, Denis. 2004. *PLM : la gestion collaborative du cycle de vie des produits (Product Life-Cycle Management)*. Hermès Science.

Di Battista, Giuseppe, Eades, Peter, Tamassia, Roberto, & Tollis, Ioannis G. 1994. Algorithms for drawing graphs : an annotated bibliography. *Computational Geometry*, **4**(5), 235–282.

Dickson, James, Drury, Heather, & Van Essen, David C. 2001. The surface management system (SuMS) database : a surface–based database to aid cortical surface reconstruction, visualization and analysis. *Philosophical Transactions of the Royal Society of London. Series B : Biological Sciences*, **356**(1412), 1277–1292.

Diehl, Stephan, & Görg, Carsten. 2002. Graphs, they are changing. *Pages 23–31 of : Graph drawing*. Springer.

Dojat, Michel, Pélégrini-Issac, Mélanie, Ahmad, Farooq, Barillot, Christian, Batrancourt, Bénédicte, Gaignard, Alban, Gibaud, Bernard, Girard, Pascal, Godard, David, Kassel, Gilles, *et al.* 2011. NeuroLOG : a framework for the sharing and reuse of distributed tools and data in neuroimaging. *Pages 26–30 of : Proceedings of the 17th Annual Meeting of the Organization for Human Brain Mapping held in Québec*.

Doucet, Gaëlle, Naveau, Mikaël, Petit, Laurent, Delcroix, Nicolas, Zago, Laure, Crivello, Fabrice, Jobard, Gael, Tzourio-Mazoyer, Nathalie, Mazoyer, Bernard, Mellet, Emmanuel, *et al.* 2011. Brain activity at rest : a multiscale hierarchical functional organization. *Journal of neurophysiology*, **105**(6), 2753–2763.

Ducellier, Guillaume. 2008. *Gestion de règles expertes en ingénierie collaborative : applications aux plateformes PLM*. Ph.D. thesis, Troyes.

Dwyer, Tim, Marriott, Kim, Schreiber, Falk, Stuckey, Peter J, Woodward, Michael, & Wybrow, Michael. 2008. Exploration of networks using overview+ detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, **14**(6), 1293–1300.

Dwyer, Tim, Marriott, Kim, & Wybrow, Michael. 2009. Topology preserving constrained graph layout. *Pages 230–241 of : Graph Drawing*. Springer.

Eades, Peter. 1984. A heuristics for graph drawing. *Congressus numerantium*, **42**, 146–160.

Eades, Peter, Lai, Wei, Misue, Kazuo, & Sugiyama, Kozo. 1991. *Preserving the mental map of a diagram.* International Institute for Advanced Study of Social Information Science, Fujitsu Limited.

Elmqvist, Niklas, & Fekete, Jean-Daniel. 2010. Hierarchical aggregation for information visualization : Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, **16**(3), 439–454.

Elmqvist, Niklas, Do, Thanh-Nghi, Goodell, Howard, Henry, Nathalie, & Fekete, Jean-Daniel. 2008. ZAME : Interactive large-scale graph visualization. *Pages 215–222 of : Visualization Symposium (PacificVIS'08).* IEEE.

Erten, Cesim, Harding, Philip J, Kobourov, Stephen G, Wampler, Kevin, & Yee, Gary. 2004. GraphAEL : Graph animations with evolving layouts. *Pages 98–110 of : Graph Drawing.* Springer.

Euler, Leonhard. 1741. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, **8**, 128–140.

Eynard, Benoît, Gallet, Thomas, Nowak, Pierre, & Roucoules, Lionel. 2004. UML based specifications of PDM product structure and workflow. *Computers in industry*, **55**(3), 301–316.

Eynard, Benoît, Liénard, Sébastien, Charles, Sébastien, & Odinot, Aurélien. 2005. Web-based collaborative engineering support system : applications in mechanical design and structural analysis. *Concurrent engineering*, **13**(2), 145–153.

Farrugia, Michael, & Quigley, Aaron. 2011. Effective temporal graph layout : A comparative study of animation versus static display methods. *Information Visualization*, **10**(1), 47–64.

Federico, Paolo, Pfeffer, Jurgen, Aigner, Wolfgang, Miksch, Silvia, & Zenk, Lukas. 2012. Visual analysis of dynamic networks using change centrality. *Pages 179–183 of : Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).* IEEE Computer Society.

Fenves, Steven J, Foufou, Sebti, Bock, Conrad, & Sriram, Ram D. 2008. CPM2 : a core model for product data. *Journal of Computing and Information Science in Engineering*, **8**(1), 014501.

Ferguson, Adam R, Nielson, Jessica L, Cragin, Melissa H, Bandrowski, Anita E, & Martone, Maryann E. 2014. Big data from small data : data-sharing in the 'long tail' of neuroscience. *Nature neuroscience*, **17**(11), 1442–1447.

Fielding, Emanuela AS, McCardle, John R, Eynard, Benoit, Hartman, Nathan, & Fraser, Alister. 2014. Product lifecycle management in design and engineering education : International perspectives. *Concurrent Engineering : Research and Applications*, **22**(2), 123–134.

Fox, Peter T, & Lancaster, Jack L. 2002. Mapping context and content : the BrainMap model. *Nature reviews. Neuroscience*, **3**(4), 319–21.

Fox, Peter T, Laird, Angela R, Fox, Sarabeth P, Fox, P Mickle, Uecker, Angela M, Crank, Michelle, Koenig, Sandra F, & Lancaster, Jack L. 2005. BrainMap taxonomy of experimental design : description and evaluation. *Human brain mapping*, **25**(1), 185–98.

Frishman, Yaniv, & Tal, Ayellet. 2008. Online dynamic graph drawing. *Visualization and Computer Graphics, IEEE Transactions on*, **14**(4), 727–740.

Friston, Karl J, *et al.* 1994. Functional and effective connectivity in neuroimaging : a synthesis. *Human brain mapping*, **2**(1-2), 56–78.

Fruchterman, Thomas MJ, & Reingold, Edward M. 1991. Graph drawing by force-directed placement. *Software : Practice and Experience*, **21**(11), 1129–1164.

Gadde, Syam, Aucoin, Nicole, Grethe, Jeffrey S, Keator, David B, Marcus, Daniel S, Pieper, Steve, FBIRN, MBIRN, *et al.* 2012. XCEDE : an extensible schema for biomedical data. *Neuroinformatics*, **10**(1), 19–32.

Gardan, Nicolas. 2005. *Proposition d'une méthodologie de travail collaboratif : Concepts et applications*. Ph.D. thesis, Universié de Reims.

Gerhard, Stephan, Daducci, Alessandro, Lemkaddem, Alia, Meuli, Reto, Thiran, Jean-Philippe, & Hagmann, Patric. 2011. The connectome viewer toolkit : an open source framework to manage, analyze, and visualize connectomes. *Frontiers in neuroinformatics*, **5**(June), 3.

Ghoniem, Mohammad, Fekete, Jean-Daniel, & Castagliola, Philippe. 2004. A comparison of the readability of graphs using node-link and matrix-based representations. *Pages 17–24 of : IEEE Symposium on Information Visualization (INFOVIS 2004)*. Ieee.

Gibson, Helen, Faith, Joe, & Vickers, Paul. 2013. A survey of two-dimensional graph layout techniques for information visualisation. *Information visualization*, **12**(3-4), 324–357.

Goble, Carole, & Stevens, Robert. 2008. State of the nation in data integration for bioinformatics. *Journal of biomedical informatics*, **41**(5), 687–93.

Gomez-Marin, Alex, Paton, Joseph J, Kampff, Adam R, Costa, Rui M, & Mainen, Zachary F. 2014. Big behavioral data : psychology, ethology and the foundations of neuroscience. *Nature neuroscience*, **17**(11), 1455–1462.

Gorgolewski, Krzysztof, Burns, Christopher D, Madison, Cindee, Clark, Dav, Halchenko, Yaroslav O, Waskom, Michael L, & Ghosh, Satrajit S. 2011. Nipype : A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python. *Frontiers in Neuroinformatics*, **5**, 13.

Grebici, Khadidja. 2007. *La maturité de l'information et le processus de conception collaborative*. Ph.D. thesis, Institut National Polytechnique de Grenoble-INPG.

Grieves, Michael. 2005. *Product Lifecycle Management : Driving the Next Generation of Lean Thinking.* McGraw Hill Professional.

Hadlak, Steffen, Schulz, H, & Schumann, Heidrun. 2011. In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, **17**(12), 2334–2343.

Harris, Paul A, Taylor, Robert, Thielke, Robert, Payne, Jonathon, Gonzalez, Nathaniel, & Conde, Jose G. 2009. Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics*, **42**(2), 377–381.

He, Weiqing, & Marriott, Kim. 1997. Constrained graph layout. *Pages 217–232 of : Graph Drawing.* Springer.

He, Yong, & Evans, Alan. 2010. Graph theoretical modeling of brain connectivity. *Current opinion in neurology*, **23**(4), 341–350.

Henry, Nathalie, Fekete, Jean-Daniel, & McGuffin, Michael J. 2007. NodeTrix : a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), 1302–1309.

Herman, Ivan, Melançon, Guy, & Marshall, M Scott. 2000. Graph visualization and navigation in information visualization : A survey. *IEEE Transactions on Visualization and Computer Graphics*, **6**(1), 24–43.

Heymann, Sébastien. 2013. *Exploratory link stream analysis for event detection.* Ph.D. thesis, Université Pierre et Marie Curie-Paris VI.

Holten, Danny. 2006. Hierarchical edge bundles : Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, **12**(5), 741–748.

Hutchison, R Matthew, Womelsdorf, Thilo, Allen, Elena A, Bandettini, Peter A, Calhoun, Vince D, Corbetta, Maurizio, Della Penna, Stefania, Duyn, Jeff H, Glover, Gary H, Gonzalez-Castillo, Javier, *et al.* 2013. Dynamic functional connectivity : promise, issues, and interpretations. *Neuroimage*, **80**, 360–378.

Jacomy, Mathieu, Heymann, Sebastien, Venturini, Tommaso, & Bastian, Mathieu. 2011. Forceatlas2, a continuous graph layout algorithm for handy network visualization. *Medialab center of research*, **560**.

Jagou, Patrick. 1993. *Concurrent engineering : la maîtrise des coûts, des délais et de la qualité.* Hermès.

Joliot, Marc, Delcroix, Nicolas, Zago, Laure, Vigneau, M, Crivello, Fabrice, Simon, G, Petit, Laurent, Turbelin, M. R, Naveau, Mickaël, Lamberton, F, Hervé, Pierre-Yves, Jobard, Gaël, Mellet, E, Mazoyer, Bernard, & Tzourio-Mazoyer, Nathalie. 2010. GINdb : portable database

for the storage and processing of human functional brain imaging data. *In : Proceedings of the 16th Annual Meeting of the Organization for Human Brain Mapping Barcelona, Spain [Poster].*

Joliot, Marc, Jobard, Gaël, Naveau, Mikaël, Delcroix, Nicolas, Petit, Laurent, Zago, Laure, Crivello, Fabrice, Mellet, Emmanuel, Mazoyer, Bernard, & Tzourio-Mazoyer, Nathalie. 2015. AICHA : An atlas of intrinsic connectivity of homotopic areas. *Journal of neuroscience methods*, **254**, 46–59.

Jusufi, Ilir. 2013. Multivariate Networks : Visualization and Interaction Techniques.

Kamada, Tomihisa, & Kawai, Satoru. 1989. An algorithm for drawing general undirected graphs. *Information processing letters*, **31**(1), 7–15.

Kaye, Jane, Heeney, Catherine, Hawkins, Naomi, De Vries, Jantina, & Boddington, Paula. 2009. Data sharing in genomics—re-shaping scientific practice. *Nature Reviews Genetics*, **10**(5), 331–335.

Keator, David B, Grethe, Jeffrey S, Marcus, D, Ozyurt, B, Gadde, Syam, Murphy, Sean, Pieper, Steve, Greve, D, Notestine, R, Bockholt, H Jeremy, *et al.* 2008. A national human neuroimaging collaboratory enabled by the Biomedical Informatics Research Network (BIRN). *Pages 162–172 of : IEEE Transactions on Information Technology in Biomedicine*, vol. 12. IEEE.

Keator, David B, Helmer, K, Steffener, Jason, Turner, Jessica A, Van Erp, Theo GM, Gadde, Syam, Ashish, N, Burns, GA, & Nichols, B Nolan. 2013. Towards structured sharing of raw and derived neuroimaging data across existing resources. *Neuroimage*, **82**, 647–661.

Keim, Daniel, *et al.* 2002. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, **8**(1), 1–8.

Keller, Tanja, & Tergan, Sigmar-Olaf. 2005. Visualizing knowledge and information : An introduction. *Knowledge and information visualization*, 1–23.

Kiritsis, Dimitris, Bufardi, Ahmed, & Xirouchakis, Paul. 2003. Research issues on product lifecycle management and information tracking using smart embedded systems. *Advanced Engineering Informatics*, **17**(3-4), 189–202.

Knickmeyer, Rebecca C, Gouttard, Sylvain, Kang, Chaeryon, Evans, Dianne, Wilber, Kathy, Smith, J Keith, Hamer, Robert M, Lin, Weili, Gerig, Guido, & Gilmore, John H. 2008. A structural MRI study of human brain development from birth to 2 years. *The Journal of Neuroscience*, **28**(47), 12176–12182.

Kobourov, Stephen G. 2012. Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv :1201.3011.*

Konstantinov, G. 1988. Emerging standards for design management systems. *Pages 16–21 of : Proceedings of the Computer Standards Conference – Computer Standards Evolution : Impact and Imperatives*. IEEE.

Kruggel, Frithjof, Turner, Jessica, Muftuler, L Tugan, Initiative, Alzheimer's Disease Neuroimaging, *et al.* 2010. Impact of scanner hardware and imaging protocol on image quality and compartment volume precision in the ADNI cohort. *Neuroimage*, **49**(3), 2123–2133.

Kruja, Eriola, Marks, Joe, Blair, Ann, & Waters, Richard. 2002. A short note on the history of graph drawing. *Graph Drawing*, 272–286.

Lantada, Andrés Díaz, & Morgado, Pilar Lafont. 2013. *Computer-Aided Design (CAD) Technologies for Biodevices*. Springer.

Larkin, Jill H. 1981. *The role of problem representation in physics*. Ph.D. thesis, Carnegie-Mellon University, Department of Psychology Pittsburgh, PA.

Lee, Bongshin, Plaisant, Catherine, Parr, Cynthia Sims, Fekete, Jean-Daniel, & Henry, Nathalie. 2006. Task taxonomy for graph visualization. *Pages 1–5 of : Proceedings of the 2006 AVI workshop on BEyond time and errors : novel evaluation methods for information visualization*. ACM.

Lefèvre, J, Charles, S, Bosch-Mauchand, M, Eynard, B, & Padiolleau, E. 2012. Towards Multidisciplinary Modeling and Simulation : interoperability issues and challenges for Mechatronic Engineering. *In : International Conference on Tools and Methods of Competitive Engineering*.

Lefèvre, Jérémy, Charles, Sébastien, Bosch-Mauchand, Magali, Eynard, Benoît, & Padiolleau, Éric. 2014. Multidisciplinary modelling and simulation for mechatronic design. *Journal of Design Research 9*, **12**(1-2), 127–144.

Lerman, Kristina, Ghosh, Rumi, & Kang, Jeon Hyung. 2010. Centrality metric for dynamic networks. *Pages 70–77 of : Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM.

Liu, D Tony, & Xu, X William. 2001. A review of web-based product data management systems. *Computers in industry*, **44**(3), 251–262.

Liu, Shixia, Wu, Yingcai, Wei, Enxun, Liu, Mengchen, & Liu, Yang. 2013. Storyflow : Tracking the evolution of stories. *Visualization and Computer Graphics, IEEE Transactions on*, **19**(12), 2436–2445.

MacKenzie-Graham, Allan J, Van Horn, John D, Woods, Roger P, Crawford, Karen L, & Toga, Arthur W. 2008. Provenance in neuroimaging. *Neuroimage*, **42**(1), 178–195.

Marchesi, Michele, Mannaro, Katiuscia, Uras, Selene, & Locci, Mario. 2007. Distributed Scrum in research project management. *Pages 240–244 of : Agile Processes in Software Engineering and Extreme Programming*. Springer.

Marcus, Daniel S, Olsen, Timothy R, Ramaratnam, Mohana, & Buckner, Randy L. 2007. The Extensible Neuroimaging Archive Toolkit and Sharing Neuroimaging Data. *Neuroinformatics*, **00**, 11–34.

Martin, Shawn, Brown, W Michael, Klavans, Richard, & Boyack, Kevin W. 2011. OpenOrd : an open-source toolbox for large graph layout. *Pages 786806–786806 of : IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics.

Mazoyer, B, Mellet, E, Perchey, G, Zago, L, Crivello, F, Jobard, G, Delcroix, N, Vigneau, M, Leroux, G, Petit, L, *et al.* 2015. BIL&GIN : a neuroimaging, cognitive, behavioral, and genetic database for the study of human brain lateralization. *NeuroImage*.

Ming, XG, Yan, JQ, Lu, WF, & Ma, DZ. 2005. Technology solutions for collaborative product lifecycle management–status review and future trend. *Concurrent Engineering Research & Application*, **13**(4), 311–319.

Moen, Sven. 1990. Drawing dynamic trees. *Software, IEEE*, **7**(4), 21–28.

Naveau, Mikaël. 2012. *Connectivité fonctionnelle cérébrale pendant l'état de repos : modélisation multi-échelle*. Ph.D. thesis, Université de Caen.

Negri, Antonio, & Vercellone, Carlo. 2008. Le rapport capital/travail dans le capitalisme cognitif. *Multitudes*, **32**(1), 39–50.

Nguyen Van, T. 2006. Ingénierie système appliqué à la gestion des données techniques en entreprise étendue : Application aux boucles de conception/simulation. *Ecole Centrale Paris*.

Nichols, B Nolan, & Pohl, Kilian M. 2015. Neuroinformatics software applications supporting electronic data capture, management, and sharing for the neuroimaging community. *Neuropsychology review*, **25**(3), 356–368.

Nicosia, Vincenzo, Tang, John, Mascolo, Cecilia, Musolesi, Mirco, Russo, Giovanni, & Latora, Vito. 2013. Graph metrics for temporal networks. *Pages 15–40 of : Temporal Networks*. Springer.

Nielsen, Finn Årup. 2009. Brede Wiki : Neuroscience data structured in a wiki. *Pages 129–133 of : Proceedings of the Fourth Workshop on Semantic Wikis–The Semantic Wiki Web*, vol. 464.

Noack, Andreas. 2004. An energy model for visual graph clustering. *Pages 425–436 of : Graph Drawing*. Springer.

Oldfield, RC. 1971. The assessment and analysis of handedness : the Edinburgh inventory. *Neuropsychologia*, **9**(1), 97–113.

Palla, Gergely, Barabási, Albert-László, & Vicsek, Tamás. 2007. Quantifying social group evolution. *Nature*, **446**(7136), 664–667.

Papenberg, Goran, Salami, Alireza, Persson, Jonas, Lindenberger, Ulman, & Bäckman, Lars. 2015. Genetics and functional imaging : Effects of APOE, BDNF, COMT, and KIBRA in aging. *Neuropsychology review*, **25**(1), 47–62.

Pavlopoulos, Georgios A, Wegener, Anna-Lynn, & Schneider, Reinhard. 2008. A survey of visualization tools for biological network analysis. *BioData Min*, **1**(1), 12.

Petit, Laurent, Crivello, Fabrice, Mellet, E, Jobard, Gaël, Zago, Laure, Joliot, Marc, Mazoyer, Bernard, & Tzourio-Mazoyer, Nathalie. 2012. BIL&GIN : a database for the study of hemispheric specialization. *Proceedings of the 18th Annual Meeting of the Organization for Human Brain Mapping, Beijing, China.*

Pham, Cong Cuong, Matta, Nada, Durupt, Alexandre, Eynard, Benoit, Ducellier, Guillaume, *et al.* 2015. KNOWLEDGE SHARING IN HETEROGENEOUS DATA CONTEXT : APPLICATION IN PLM. *In : DS 80-10 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 10 : Design Information and Knowledge Management Milan, Italy, 27-30.07. 15.*

Pohl, Mathias, Reitz, Florian, & Birke, Peter. 2008. As time goes by : integrated visualization and analysis of dynamic networks. *Pages 372–375 of : Proceedings of the working conference on Advanced visual interfaces.* ACM.

Poldrack, Russell A, Fletcher, Paul C, Henson, Richard N, Worsley, Keith J, Brett, Matthew, & Nichols, Thomas E. 2008. Guidelines for reporting an fMRI study. *NeuroImage*, **40**(2), 409–14.

Poldrack, Russell A, Barch, Deanna M, Mitchell, Jason P, Wager, Tor D, Wagner, Anthony D, Devlin, Joseph T, Cumba, Chad, Koyejo, Oluwasanmi, & Milham, Michael P. 2013. Toward open sharing of task-based fMRI data : the OpenfMRI project. *Frontiers in neuroinformatics*, **7**.

Poline, Jean-Baptiste, Breeze, Janis L, Ghosh, Satrajit, Gorgolewski, Krzysztof, Halchenko, Yaroslav O, Hanke, Michael, Haselgrove, Christian, Helmer, Karl G, Keator, David B, Marcus, Daniel S, Poldrack, Russell a, Schwartz, Yannick, Ashburner, John, & Kennedy, David N. 2012. Data sharing in neuroimaging research. *Frontiers in neuroinformatics*, **6**(Apr.), 9.

Purchase, Helen C. 2002. Metrics for graph drawing aesthetics. *Journal of Visual Languages & Computing*, **13**(5), 501–516.

Purchase, Helen C, & Samra, Amanjit. 2008. Extremes are better : Investigating mental map preservation in dynamic graphs. *Pages 60–73 of : Diagrammatic Representation and Inference.* Springer.

Purchase, Helen C, Hoggan, Eve, & Görg, Carsten. 2007. How important is the "mental map" ?– an empirical investigation of a dynamic graph layout algorithm. *Pages 184–195 of : Graph drawing.* Springer.

Randoing, Jean-Martial. 1995. *Les SGDT.* Hermès.

Reitz, Florian, Pohl, Mathias, & Diehl, Stephan. 2009. Focused animation of dynamic compound graphs. *Pages 679–684 of : Information Visualisation, 2009 13th International Conference.* IEEE.

Robertson, George, Fernandez, Roland, Fisher, Danyel, Lee, Bongshin, & Stasko, John. 2008. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, **14**(6), 1325–1332.

Rosenman, Mike A, & Gero, John S. 1999. Purpose and function in a collaborative CAD environment. *Reliability Engineering & System Safety*, **64**(2), 167–179.

Rufiange, Sébastien, & McGuffin, Michael J. 2013. DiffAni : Visualizing dynamic graphs with a hybrid of difference maps and animation. *Visualization and Computer Graphics, IEEE Transactions on*, **19**(12), 2556–2565.

Saaksvuori, Antti, & Immonen, Anselmi. 2008. *Product lifecycle management.* Springer Science & Business Media.

Sackett, PJ. 1990. Using your resources effectively. *Proc. Computers in Manufacturing, Birmingham, UK.*

Saffrey, Peter, & Purchase, Helen. 2008. The mental map versus static aesthetic compromise in dynamic graphs : a user study. *Pages 85–93 of : Proceedings of the ninth conference on Australasian user interface-Volume 76.* Australian Computer Society, Inc.

Schaeffer, Satu Elisa. 2007. Graph clustering. *Computer Science Review*, **1**(1), 27–64.

Scott, Adam, Courtney, Will, Wood, Dylan, de la Garza, Raul, Lane, Susan, King, Margaret, Wang, Runtang, Roberts, Jody, Turner, Jessica a, & Calhoun, Vince D. 2011. COINS : An Innovative Informatics and Neuroimaging Tool Suite Built for Large Heterogeneous Datasets. *Frontiers in neuroinformatics*, **5**(December), 33.

SDK, & PLMXML. *V6. 0.1-PLM XML Schema Functional Description, November 2005.*

Shen, Weiming. 2003. Editorial of the special issue on knowledge sharing in collaborative design environments. *Computers in Industry*, **52**(1), 1–3.

Shen, Weiming, Hao, Qi, & Li, Weidong. 2008. Computer supported collaborative design : Retrospective and perspective. *Computers in Industry*, **59**(9), 855–862.

Shneiderman, Ben. 1996. The eyes have it : A task by data type taxonomy for information visualizations. *Pages 336–343 of : Visual Languages, 1996. Proceedings., IEEE Symposium on.* IEEE.

Shneiderman, Ben, & Aris, Aleks. 2006. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics*, **12**(5), 733–740.

Simmhan, Yogesh L, Plale, Beth, & Gannon, Dennis. 2005. A survey of data provenance in e-science. *ACM Sigmod Record*, **34**(3), 31–36.

Smith, Barry. 1988. Foundations of Gestalt theory.

Smith, Marc A, Shneiderman, Ben, Milic-Frayling, Natasa, Mendes Rodrigues, Eduarda, Barash, Vladimir, Dunne, Cody, Capone, Tony, Perer, Adam, & Gleave, Eric. 2009. Analyzing (social media) networks with NodeXL. *Pages 255–264 of : Proceedings of the fourth international conference on Communities and technologies*. ACM.

Smoot, Michael E, Ono, Keiichiro, Ruscheinski, Johannes, Wang, Peng-Liang, & Ideker, Trey. 2011. Cytoscape 2.8 : new features for data integration and network visualization. *Bioinformatics*, **27**(3), 431–432.

Sohlenius, Gunnar. 1992. Concurrent engineering. *CIRP Annals-Manufacturing Technology*, **41**(2), 645–655.

Sriti, Mohamed-Foued, & Boutinaud, Philippe. 2012. PLMXQuery : Towards a Standard PLM Querying Approach. *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, 379–388.

Stark, John. 2004. *Product lifecycle management : paradigm for 21st century product realization*. Springer, Berlin.

Sudarsan, Rachuri, Fenves, Steven J, Sriram, Ram D, & Wang, Fujun. 2005. A product information modeling framework for product lifecycle management. *Computer-aided design*, **37**(13), 1399–1411.

Sugiyama, Kozo, Tagawa, Shojiro, & Toda, Mitsuhiko. 1981. Methods for visual understanding of hierarchical system structures. *Pages 109–125 of : IEEE Transactions on Systems, Man and Cybernetics*, vol. 11. IEEE.

Sweller, John. 1994. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, **4**(4), 295–312.

Sylvester, John Joseph. 1878. Chemistry and algebra. *Nature*, **17**, 284.

Tang, John, Leontiadis, Ilias, Scellato, Salvatore, Nicosia, Vincenzo, Mascolo, Cecilia, Musolesi, Mirco, & Latora, Vito. 2013. Applications of temporal graph metrics to real-world networks. *Pages 135–159 of : Temporal Networks*. Springer.

Teeters, Jeffrey L, Harris, Kenneth D, Millman, K Jarrod, Olshausen, Bruno A, & Sommer, Friedrich T. 2008. Data sharing for computational neuroscience. *Neuroinformatics*, **6**(1), 47–55.

Tekušová, Tatiana, & Schreck, Tobias. 2008. Visualizing time-dependent data in multivariate hierarchic plots-design and evaluation of an economic application. *Pages 143–150 of : 12th International Conference on Information Visualisation (IV'08)*. IEEE.

Temal, Lynda, Dojat, Michel, Kassel, Gilles, & Gibaud, Bernard. 2008. Towards an ontology for sharing medical images and regions of interest in neuroimaging. *Journal of Biomedical Informatics*, **41**(5), 766–778.

Terzi, Sergio, Panetto, Hervé, Morel, Gérard, & Garetti, Marco. 2007. A holonic metamodel for product traceability in Product Lifecycle Management. *International Journal of Product Lifecycle Management*, **2**(3), 253–289.

Terzi, Sergio, Abdelaziz, Bouras, Butta, Bebashi, Garetti, Marco, & Kiritsis, Dimitri. 2010. Product lifecycle management – from its history to its new role. *International Journal of Product Lifecycle Management*, **4**(4), 360–389.

Teyseyre, Alfredo R, & Campo, Marcelo R. 2009. An overview of 3D software visualization. *Visualization and Computer Graphics, IEEE Transactions on*, **15**(1), 87–105.

Tominski, Christian, Abello, James, Van Ham, Frank, & Schumann, Heidrun. 2006. Fisheye tree views and lenses for graph visualization. *Pages 17–24 of : Tenth International Conference on Information Visualization (IV 2006)*. IEEE.

Tominski, Christian, Abello, James, & Schumann, Heidrun. 2009. CGV—An interactive graph visualization system. *Computers & Graphics*, **33**(6), 660–678.

Trier, Matthias. 2006. D. 7 Towards a Social Network Intelligence Tool for visual Analysis of Virtual Communication Networks.

Troussier, Nadège. 2010. De la conception mécanique à la conception collaborative et robuste des systèmes mécatroniques : une approche combinant maîtrise des performances des systèmes techniques et gestion des connaissances. *Habilitation à Diriger des Recherches, Université de Technologie de Compiègne*.

Tufte, Edward R. 1991. Envisioning information. *Optometry & Vision Science*, **68**(4), 322–324.

Tversky, Barbara, Morrison, Julie Bauer, & Betrancourt, Mireille. 2002. Animation : can it facilitate ? *International journal of human-computer studies*, **57**(4), 247–262.

Uddin, Shahadat, Hossain, Liaquat, & Wigand, Rolf T. 2014. New Direction in Degree Centrality Measure : Towards a Time-Variant Approach. *International Journal of Information Technology & Decision Making*, **13**(04), 865–878.

Van Ham, Frank, & Perer, Adam. 2009. "Search, Show Context, Expand on Demand" : Supporting Large Graph Exploration with Degree-of-Interest. *Visualization and Computer Graphics, IEEE Transactions on*, **15**(6), 953–960.

Van Horn, JD, Grethe, JS, Kostelec, P, Woodward, JB, Aslam, JA, Rus, D, Rockmore, D, & Gazzaniga, MS. 2001. The Functional Magnetic Resonance Imaging Data Center (fMRIDC) : the challenges and rewards of large–scale databasing of neuroimaging studies. *Philosophical*

Transactions of the Royal Society of London. Series B : Biological Sciences, **356**(1412), 1323–1339.

Van Horn, John Darrell, & Toga, Arthur W. 2009. Is it time to re-prioritize neuroimaging databases and digital repositories ? *NeuroImage*, **47**(4), 1720–34.

Van Horn, John Darrell, & Toga, Arthur W. 2014. Human neuroimaging as a "Big Data" science. *Brain imaging and behavior*, **8**(2), 323–331.

Van Wijk, Jarke J, & Van de Wetering, Huub. 1999. Cushion treemaps : Visualization of hierarchical information. *Pages 73–78 of : IEEE Symposium on Information Visualization (Info Vis' 99)*. IEEE.

Von Landesberger, Tatiana, Kuijper, Arjan, Schreck, Tobias, Kohlhammer, Jörn, van Wijk, Jarke J, Fekete, J-D, & Fellner, Dieter W. 2011. Visual analysis of large graphs : state-of-the-art and future research challenges. *Computer graphics forum*, **30**(6), 1719–1749.

Walter, Thomas, Shattuck, David W, Baldock, Richard, Bastin, Mark E, Carpenter, Anne E, Duce, Suzanne, Ellenberg, Jan, Fraser, Adam, Hamilton, Nicholas, Pieper, Steve, Ragan, Mark a, Schneider, Jurgen E, Tomancak, Pavel, & Hériché, Jean-Karim. 2010. Visualization of image data from cells to organisms. *Nature methods*, **7**(3 Suppl), S26–41.

Wang, Chin-Bin, Chen, Yuh-Min, Chen, Yuh-Jen, & Ho, Chengter. 2005. Methodology and system framework for knowledge management in allied concurrent engineering. *International Journal of Computer Integrated Manufacturing*, **18**(1), 53–72.

Ware, Colin. 2005. Visual queries : The foundation of visual thinking. *Pages 27–35 of : Knowledge and information visualization*. Springer.

Ware, Colin. 2012. *Information visualization : perception for design*. Elsevier.

Wattenberg, Martin. 2006. Visual exploration of multivariate graphs. *Pages 811–819 of : ACM Proceedings of the SIGCHI conference on Human Factors in computing systems*.

Webber, Jim. 2012. A programmatic introduction to neo4j. *Pages 217–218 of : Proceedings of the 3rd annual conference on Systems, Programming, and Applications : Software for Humanity*. ACM.

Weiner, Michael W, Aisen, Paul S, Jack, Clifford R, Jagust, William J, Trojanowski, John Q, Shaw, Leslie, Saykin, Andrew J, Morris, John C, Cairns, Nigel, Beckett, Laurel A, *et al.* 2010. The Alzheimer's disease neuroimaging initiative : progress report and future plans. *Alzheimer's & Dementia*, **6**(3), 202–211.

Weiner, Michael W, Veitch, Dallas P, Aisen, Paul S, Beckett, Laurel A, Cairns, Nigel J, Green, Robert C, Harvey, Danielle, Jack, Clifford R, Jagust, William, Liu, Enchi, *et al.* 2013. The Alzheimer's Disease Neuroimaging Initiative : A review of papers published since its inception. *Alzheimer's & Dementia*, **9**(5), e111–e194.

West, Douglas Brent, *et al.* 2001. *Introduction to graph theory.* Vol. 2. Prentice hall Upper Saddle River.

Wiegmann, Douglas A, Dansereau, Donald F, McCagg, Edward C, Rewey, Kirsten L, & Pitre, Urvashi. 1992. Effects of knowledge map characteristics on information processing. *Contemporary Educational Psychology*, **17**(2), 136–155.

Xia, Mingrui, Wang, Jinhui, He, Yong, *et al.* 2013. BrainNet Viewer : a network visualization tool for human brain connectomics. *PloS one*, **8**(7), e68910.

Yarkoni, Tal, Poldrack, Russell a, Van Essen, David C, & Wager, Tor D. 2010. Cognitive neuroscience 2.0 : building a cumulative science of human brain function. *Trends in cognitive sciences*, **14**(11), 489–96.

Yee, Ka-Ping, Fisher, Danyel, Dhamija, Rachna, & Hearst, Marti. 2001. Animated exploration of dynamic graphs with radial layout. *Page 43 of : IEEE Symposium on Information Visualization (InfoVis 2001).* IEEE.

Yi, Ji Soo, ah Kang, Youn, Stasko, John T, & Jacko, Julie A. 2007. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), 1224–1231.

Yi, Ji Soo, Elmqvist, Niklas, & Lee, Seungyoon. 2010. TimeMatrix : Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human–Computer Interaction*, **26**(11-12), 1031–1051.

Zaman, Loutfouz, Kalra, Ashish, & Stuerzlinger, Wolfgang. 2011. The effect of animation, dual view, difference layers, and relative re-layout in hierarchical diagram differencing. *Pages 183–190 of : Proceedings of Graphics Interface 2011.* Canadian Human-Computer Communications Society.

Zhang, Jiaje, & Norman, Donald A. 1994. Representations in distributed cognitive tasks. *Cognitive science*, **18**(1), 87–122.

Notice bibliographique

Les articles scientifiques rédigés pendant la thèse et en rapport avec le présent manuscrit sont données par catégories et dans l'ordre chronologique :

Journaux internationaux

– IJITM : Marianne Allanic, Pierre-Yves Hervé, Alexandre Durupt, Marc Joliot, Philippe Boutinaud, and Benoît Eynard. PLM as a strategy for the management of heterogeneous information in bio-medical imaging field. In *Int. J. Information Technology and Management*, 2015.

Conférences internationales avec actes

– PLM14 : Marianne Allanic, Thierry Brial, Alexandre Durupt, Marc Joliot, Philippe Boutinaud, and Benoît Eynard. Towards an enhancement of relationships browsing in mature PLM systems. *Product Lifecycle Management for a Global Market*, 2014.

– TMCE2014 : Marianne Allanic, Alexandre Durupt, Marc Joliot, Benoît Eynard, and Philippe Boutinaud. Towards a data model for PLM application in bio-medical imaging. In *Proceedings of Tools and Methods for Competitive Engineering*, 2014.

– PLM13 : Marianne Allanic, Alexandre Durupt, Marc Joliot, Benoît Eynard, and Philippe Boutinaud. Application of PLM for bio-medical imaging in neuroscience. *Product Lifecycle Management for Society*, pages 520–529, 2013.

Annexe A

Étude de cas : jeux de données multidimensionnels

Etude en neuroimagerie fonctionnelle

Analyse de la connectivité fonctionnelle cérébrale

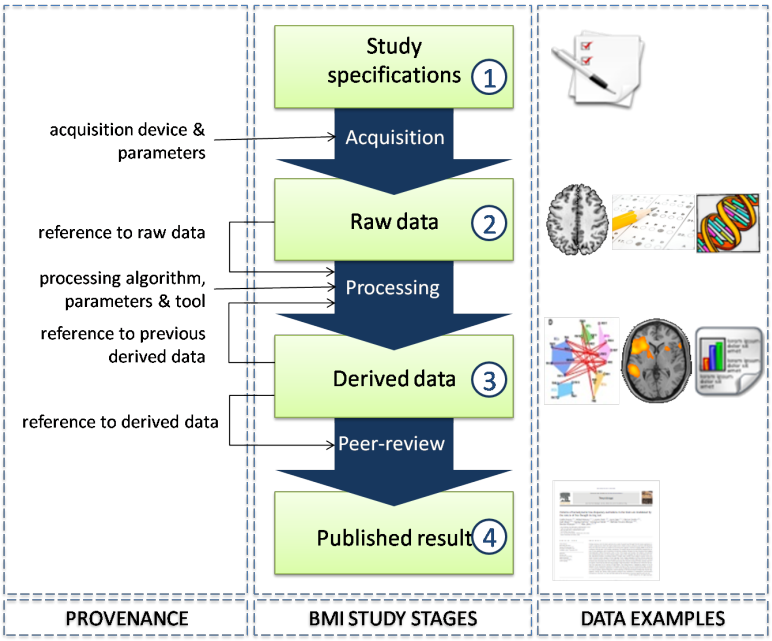
La connectivité fonctionnelle est la dépendance ou l'association statistique entre les éléments d'un *réseau* (Bullmore & Sporns, 2009). Le cerveau est segmenté en régions dont les associations sont mesurées : les régions sont les *nœuds* et les associations sont les *arêtes* du *graphe* qui représente le réseau cérébral.

Des recherches récentes en neurosciences ont commencé à analyser la connectivité des données en utilisant des méthodes de la théorie des graphes et des statistiques (voir le chapitre 3). La visualisation des réseaux peut permettre la découverte de motifs de corrélation structurelle inattendus, en particulier à travers plusieurs jeux de données. Par exemple, comparer des motifs de connectivité fonctionnelle et anatomique pre- et post- ablation de morceaux de tissu du cerveau peut aider les neuroscientifiques à comprendre comment le cerveau se reconnecte pour restaurer ses fonctions. Bien que les méthodes statistiques et de théorie des graphes soient disponibles pour permettre une telle analyse, la visualisation de réseaux en collaboration avec des outils de comparaison de la connectivité peut fournir des apports significatifs pour la découverte de motifs qui n'avaient pas été anticipés. La comparaison visuelle de graphes peut être un outil essentiel pour la compréhension de la connectivité cérébrale.

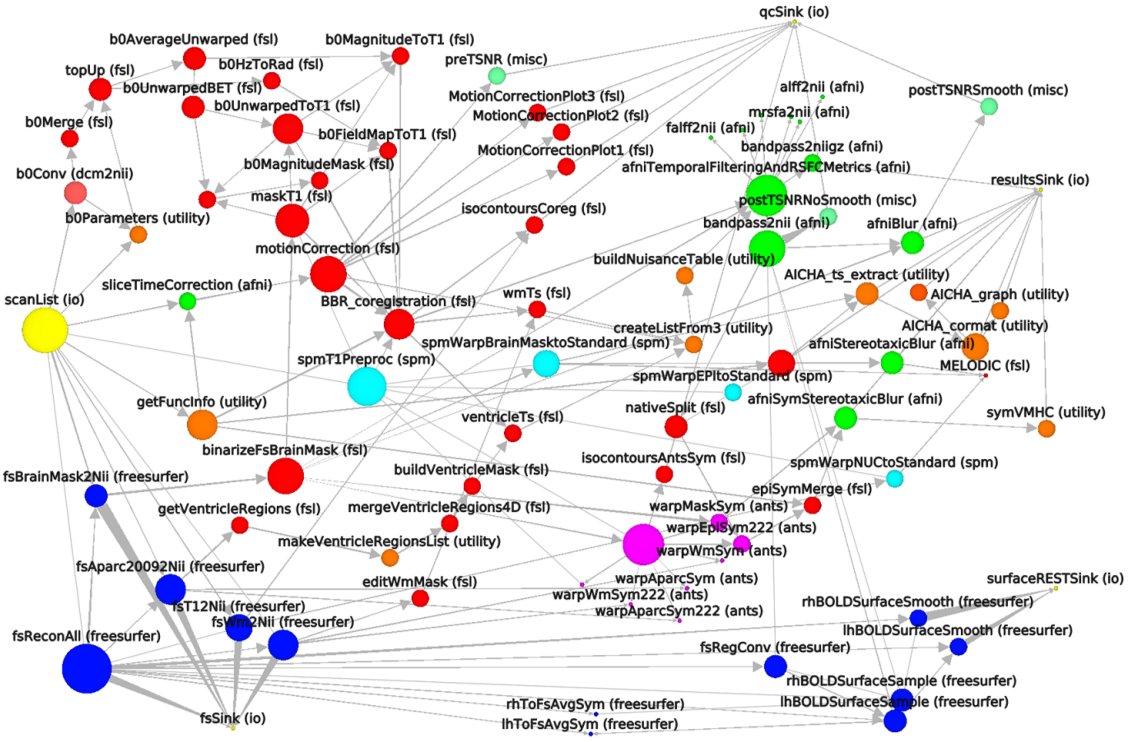
Données de l'étude

D'après l'analyse de la bibliographie (voir chapitre 2), les chercheurs en neuroimagerie ont besoin de partager et de réutiliser leurs données. La provenance des données en neuroimagerie est complexe et elle doit être gardée à toutes les étapes d'une étude pour permettre la réutilisation des données : depuis les spécifications d'une étude aux résultats publiés. Les études longitudinales impliquent une gestion dynamique des données, puisque l'évolution de données dans le temps est analysée : la provenance des données doit être strictement identique entre deux

données acquises à plusieurs années d'intervalle, et les données dérivées doivent être comparées pour étudier l'impact de l'âge sur des sujets.



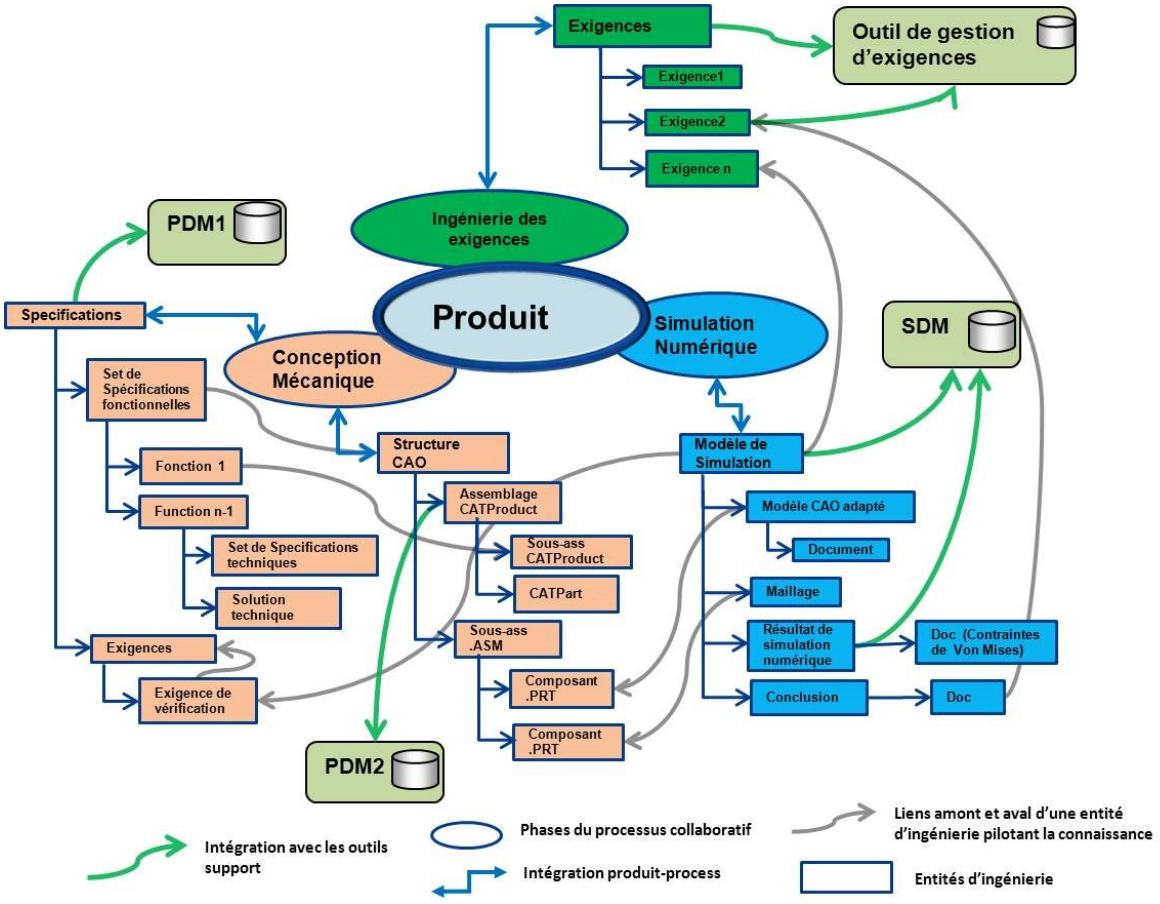
Pour illustrer la complexité des chaînes de traitement en neuroimagerie – c'est-à-dire la provenance des données dérivées –, nous présentons ci-dessous une chaîne de traitement d'images visualisé sous forme de graphe (figure réalisée par Pierre-Yves Hervé, 2015). Les briques de calcul sont reliées entre elles par les flux entrants de données. Les couleurs des nœuds représentent les logiciels d'exécution de chaque brique : FSL=rouge, SPM=bleu, AFNI=vert, ANTS=rose, code isolé=orange, Freesurfer=bleu foncé, l'entrée Nipype=jaune.



La visualisation sous forme de graphe de la chaîne de traitement facilite les tâches de parcours d'un chemin, soit dans le cas présenté ici le suivi de l'enchaînement de plusieurs briques de traitement. Cependant cette représentation visuelle de la chaîne de traitement n'est pas toujours facile à appréhender pour un utilisateur, même avec l'application d'un layout adapté. Pour pouvoir réutiliser des données dérivées, un utilisateur doit être en mesure d'une part de comprendre les liens de dépendance entre les briques de traitement, et d'autre part d'identifier les différences et les points communs entre la chaîne de traitement qui a mené à ces données dérivées et une autre chaîne de traitement ayant la même fonction (production de données d'un certain type).

Développement d'un produit

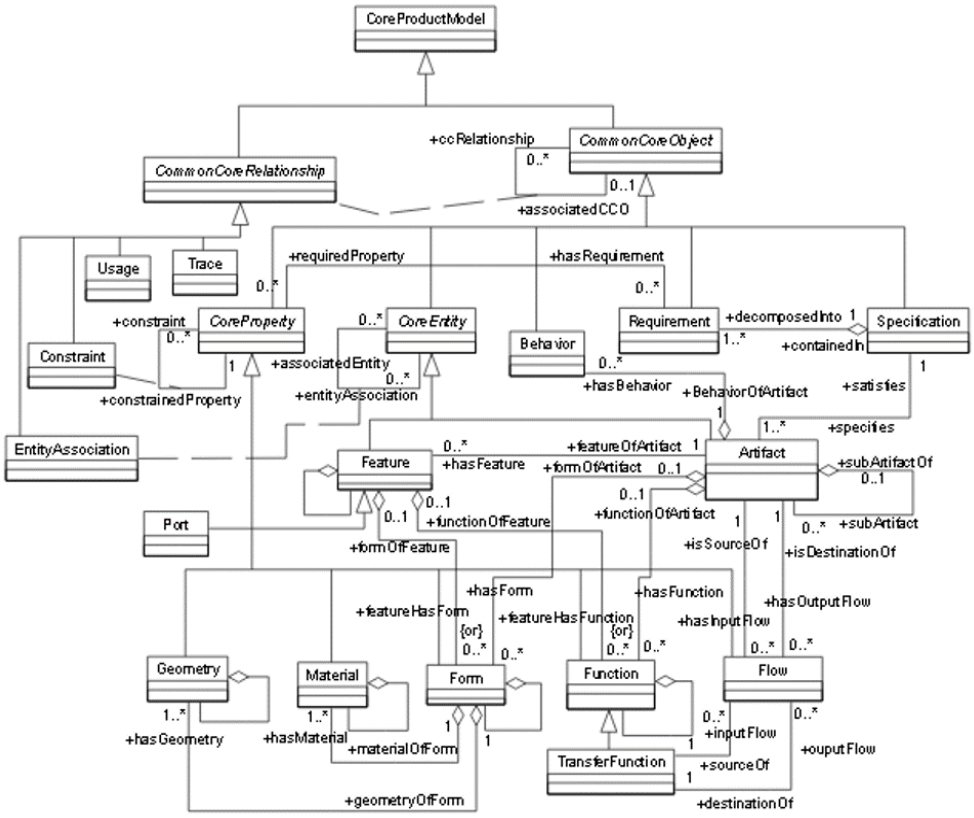
La figure ci-dessous illustre les relations entre les différentes structures de données techniques (Assouroko, 2012) : les exigences, les spécifications, l'arbre de modélisation CAO, le modèle de simulation numérique sont des données nécessaires à la conception d'un produit. L'abondance de relations entre ces données d'une part, et leur évolution dans le temps d'autre part (les versions successives des données sont conservées) les rend difficilement compréhensibles.



Que se passe-t'il quand d'autres données comme celles associées à la fabrication du produit sont ajoutées ? Comment suivre l'évolution des versions du produit ? La caractère multidimen-

sionnel – différents niveaux de description du produit – des données rend ardues non seulement leur exploration mais aussi leur navigation.

Des modèle de données pour la gestion des données produit dans le PLM ont été développés, comme le modèle CPM (Core Product Model) qui est représenté au format UML (Fenves *et al.*, 2008) ci-dessous :



Ce modèle de données permet d'assurer la provenance des données. Il est utile et nécessaire que la dynamique des relations entre les objets du modèle de données du produit puisse être analysée pour comprendre l'évolution des versions d'un produit, et que les structures de données de plusieurs produits puissent être comparées.

La cohérence des données dans la base est souvent difficilement vérifiable (voir l'annexe B pour les limites rencontrées par les interfaces PLM qui empêchent de vérifier que les données de la base sont valides), ce qui ne permet pas de garantir que la provenance des données est complètement gérée, malgré un modèle de données robuste.

Annexe B

Étude de cas : limites des interfaces PLM

Les deux études de cas présentées dans cette annexe ont été réalisées en 2013 et présentées à la conférence PLM14 (Yokohama) en 2014 [1].

Case study 1 : Migration of ACME PLM system

Manufacturing companies are now mostly equipped with PLM systems, and because of information management strategies or costs, they decide to change their PLM system more and more frequently. Therefore, new issues come out such as data consistency in source PLM systems and data import in target PLM systems. In this section, the migration of ACME PLM system is developed to highlight the need for relationships management and visualisation.

Context and data to migrate

In 2011, ACME [2] decided to migrate its Windchill PLM system to Teamcenter PLM system. The CAD software Pro/E remained the same after the migration, so there was no conversion operation. A PLM system handles two distinct types of data : metadata which are stored in a relational database, and data files which are stored in the vault, a securised file system accessible only through PLM. Relationships between data files and metadata are described in the PLM system. In the case of ACME migration, data files are drawings and 3D CAD assemblies.

An Extract Transform Load (ETL) type of implementation was set up for the migration. Indeed, Windchill and Teamcenter do not have the same concepts and data models, so an intermediate migration platform is needed to transform the data. The migration platform is composed of a temporary data files storage and a Graph DataBase (GDB), Neo4J. PLMXQuery tool (Sriti & Boutinaud, 2012) is used to populate the migration platform, which constitutes the

1. Allanic, M., Brial, T., Durupt, A., Joliot, M., Boutinaud, P., & Eynard, B. (2014). Towards an Enhancement of Relationships Browsing in Mature PLM Systems. In *Product Lifecycle Management for a Global Market* (pp. 345-354). Springer Berlin Heidelberg.

2. ACME is a French company designing and manufacturing thermal systems. The name is modified, as the authorisation of the company to use their name is under approval.

Extract phase of ETL. Through XQuery language, the metadata are converted into PLMXML language (SDK & PLMXML, n.d.) and the data files are represented with their dependencies in a ASCII instruction file. Finally, during the load phase of ETL, the target Teamcenter platform is populated thanks to PLMXMLImport Tool for the metadata and IPEM tool for the data files.

A GDB was chosen to store temporarily the metadata, because a PLM system can be seen as a set of unique objects having a set of attributes (key or value) and being linked together by typed relationships. A graph corresponds as well to this definition, as it is composed of nodes and edges having attributes. Moreover, it is easy to populate a GDB through standard formats. A screenshot of an item extracted from Windchill and displayed in Neo4J is shown in figure B.1.

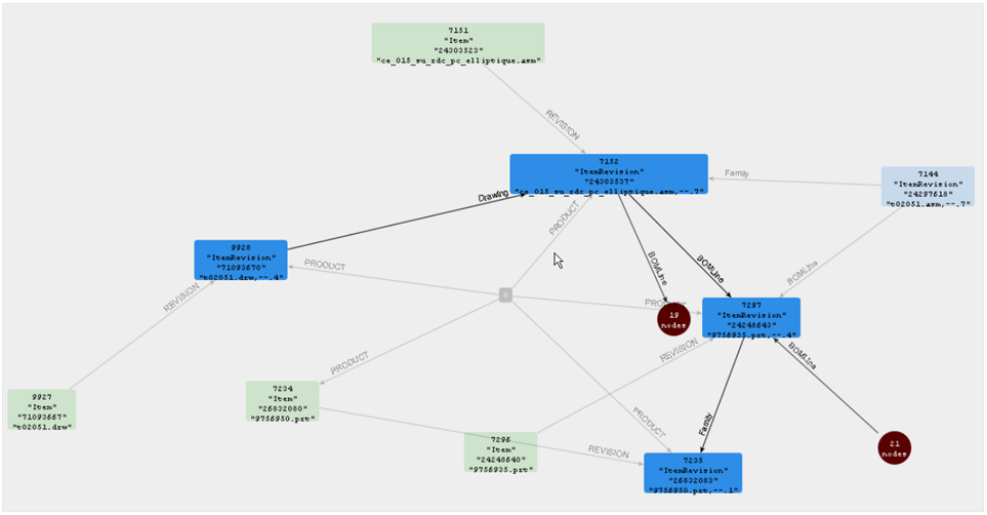


FIGURE B.1 – Screenshot in Neo4J of an item extracted from Windchill with relationships to its revisions, drawing, relatives (family items) and BOM components.

Issues during the migration

CAD files are by nature tricky data : when they are opened by appropriate CAD software, some CAD files require other dependent CAD files to be opened. This means that these dependency relationships are described somewhere in the CAD files, but must also be described in the PLM systems. When a user asks the PLM system to open a CAD file locally, the target CAD file and all dependant CAD files are downloaded from the vault. The relationships dependency must be taken in account during the migration, as data files with dependencies can only be migrated after the dependent files. Thus, the imports have to be scheduled in the right order.

Many errors occurred during the migration, due to wrong referencing between objects : obsolete instances, missing references between CAD files and between templates and drawings, dependency cycles. These issues were a real burden during the migration process which lasted for one year instead of the few months planned initially, because the initial data had to be

"cleaned". An algorithm - called Kdeep - was designed to analyse referencing errors on the GDB. It is able to locate when the errors occurred in time, and to solve automatically these issues. This algorithm was applied on the Neo4J database. Once the Kdeep algorithm had passed through the whole database, there were still some errors due to inconsistency of complex relationships, and they had to be solved by hand.

The main lesson from the migration is the unexpected number of wrong referencing of the relationships between the levels of granularity in PLM systems. The ACME company was not aware that these issues were existing in its PLM system. Obviously, it is difficult to analyse dependencies and identify referencing problems between objects in a PLM system. Besides, the Neo4J GDB used for the migration turned out to be useful to do the job. Indeed, algorithms can be easily computed on graphs to retrieve statistics and apply display filters for a better understanding of the relationships between graph nodes. Some of the errors could not have been solved out without a visual representation of the dependencies.

Case study 2 : Management of GIN research studies

A growing number of domains outside manufacturing industry are implementing PLM systems. Most of these domains handle heterogeneous data with complex relationships, which require to browse, retrieve and visualise information efficiently. Browsing needs of neuroimaging domain are developed in this section.

A PLM database in neuroimaging domain

Neuroimaging domain is multidisciplinary "by its very nature" (Van Horn *et al.*, 2001) : the study of brain require an active interaction between many specialities - physics, medicine, mathematics and engineering among others. Magnetic Resonance Imaging (MRI) is one of the most promising imaging technique to study brain complexity. Structural MRI examines brain anatomy, whereas functional MRI analyses what happens while a subject is performing a given task. A Bio-Medical Imaging (BMI) research study can be represented by four stages that constitute a cycle [3] : study specifications (stage 1), raw data (stage 2), derived data (stage 3) and published results (stage 4). Between stages, it is crucial to keep all the information to be able to understand the context of computing and the history of data, which is a requirement to reproduce derived data result or to reuse them. What a piece of data is, when, where and how it was produced, why and for whom it was performed is called provenance (Simmhan *et al.*, 2005). Due to costs, trends to huge cohorts of subjects and growing complexity of analyses, neuroimaging researchers must collaborate and share data between disciplines and laboratories (Yarkoni *et al.*, 2010), which are similar issues than the one of manufacturing industry.

PLM was proposed and shown to be relevant to manage efficiently neuroimaging heterogeneous data and to enable the conditions of sharing and reuse of data [4]. Since 2010, the GIN [5]

3. see chapter 2
4. see chapters 4 and 7
5. Groupe d'Imagerie Neurofonctionnelle - Neurofunctional Imaging Research Group

has been using the GIN first Brain Imaging Laterality (BIL&GIN1) dataset, which is composed of 300 subjects - balanced by gender and handedness - and which was acquired between 2009 and 2011 (Petit *et al.*, 2012). In 2013 a PLM system – Teamcenter 9 – was installed at the GIN, populated with the BIL&GIN1 dataset.

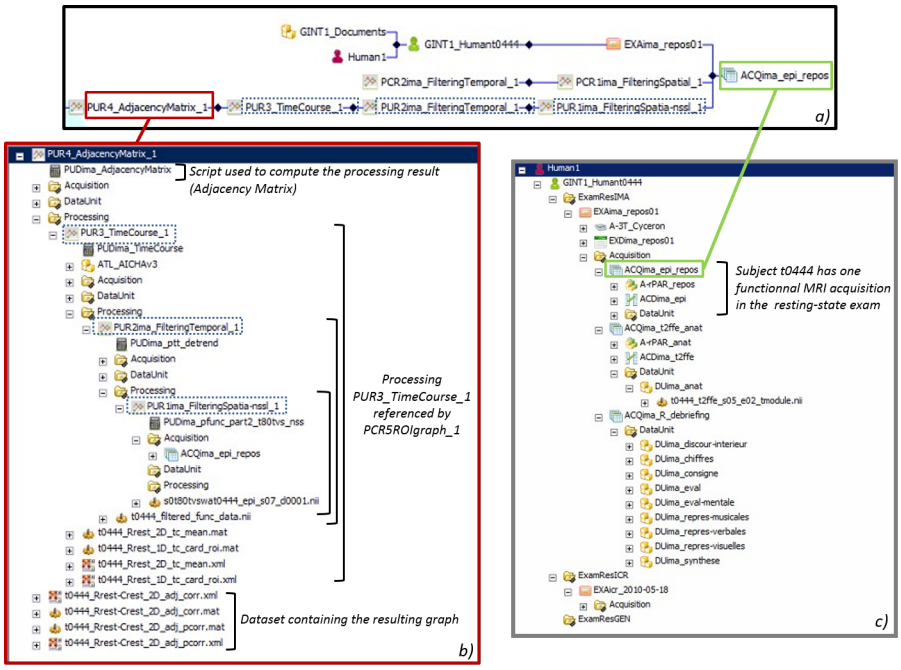


FIGURE B.2 – Screenshots of Teamcenter rich client browsing interfaces. Impact analysis view shows that acquisition ACQima_epi_repos is referenced in a raw data branch and two processing sequence branches(a) Similar information in shown in hierarchy view from top objects PCR5_ROIgraph_1 (b) and GINT1_Humant0444 (c) through descending hierarchy.

Limits of PLM systems browsing

The figure B.2 shows the two main ways of browsing information in Teamcenter – the concepts are similar in other PLM systems – : hierarchy view (descending) and impact analysis view (ascending). It is easy to see where an object has been used on close levels, as well as referencing sequences, with a limited number of relationships. However, users have to switch from one view to another to be able to navigate freely among objects relationships. Indeed, research work require data exploration to build new hypotheses, but the PLM system allows no overall view of the complexity.

The feedback of the GIN researchers on the PLM system was good concerning the capacities of the system to manage neuroimaging data organisation. However, they were very critical about the interface of the software (basic Teamcenter rich client display) : there are too many menus, icons and sub-windows in the environment. Besides, the actions of searching for information is not immediate. What they are looking for in terms of ergonomics is an over-simplified interface with few choices to make for each options, as well as a relationships full management (browsing,

consistency checking, visual information retrieval. In addition, they express the will to handle the system almost without any training. In the end, the GIN researchers are reluctant to use the PLM database, even if they recognise that it is of great benefits for storing and sharing data and associated provenance.

Discussion

The two case studies show some limits of the interfaces of current PLM systems. This section discusses the limits divided in three categories : query, analyse and browsing. Leads for future work are proposed at the end of each subsection.

Information access : query

In neuroimaging there is a trend for cross-domain analyses : imaging results are correlated with demographical, clinical, psychological and genetics data. Therefore, numerous joins between concepts are required for one single query, which make the search process complex. Indeed, to design customized queries, users have to know the exact data model organisation. From user's point of view, querying is textual browsing : the search fields that are filled in make the query engine browsing among data relationships. Without efficient retrieval capacities, a database is just a storage room. Apart from the limitation of users' daily data search, the lack of effective querying in PLM systems has an impact on the understanding of data provenance and therefore data reuse.

PLM data models should be transparent to the users and a proposal is to design a graphical query builder, with which users would be query-autonomous. Another future work would be semantic enrichment of PLM systems, based on ontology, and which handles the management of relationships between objects in a flexible way Assouroko *et al.* (2012). This would improve performances. Besides, managing PLM concept as a graph would facilitate search processes as well as query performances, thanks to graph theory algorithms.

Information visualisation : reading and analyse

The way the information is displayed inside the space of the PLM systems windows is meaningless. Only one information level is displayed at a time, under the shape of a list. As a result, it is difficult to read and understand two information levels, especially because they do not require the same display format. PLM systems interfaces are not ergonomic nor intuitive, and companies spend significant time and money in training. If in the manufacturing industry world companies have to use a PLM system to stay competitive, other PLM application domains would be reluctant to use PLM systems because of their current interfaces, which is emphasised by the case study 2 of the paper. So it is of importance that the PLM community addresses the limits of PLM interfaces. Moreover, reducing users' training time on information systems would be obviously a competitive advantage for companies.

Some future work should address new ways of information display in data management systems, particularly focusing on multi viewpoints display. On figure B.1, different levels of information are displayed in the Neo4J graph. The positive aspect is that it is easy to access to all the relationships between data and concepts, but the positive aspect is that all the edges look the same, and nothing distinguishes one level of information from the other, for instance CAD files view and product view. So multi viewpoints developments should take this remark into account.

Information access and visualisation : browsing

The two case studies developed in the paper show that graphical relationships browsers in current PLM systems are not satisfactory. Ascending and descending data relationships hierarchy can only be browsed in two different windows, which prevent from consulting two product information levels at a time or from checking relationships consistency between information levels throughout product lifecycle. As shown in the case study 1, there exists no way to analyse references consistency in a PLM system, which implies a loss of information that could be important for the competitiveness of a company.

Therefore, a major concern in the upcoming works is to visualise data relationships by graphs in one single sub-window, in order to improve browsing and visualisation of every component of data provenance in PLM systems. Further researches should be conducted regarding the application of graph theory to the analysis of relationships in PLM systems.

Annexe C

Note de synthèse des interviews menées au GIN le 23/10/2012

Rappel des objectifs : comprendre les besoins et attentes des chercheurs du labo en termes de gestion des données pour la conception du modèle de données pour la base PLM.

Interviewés : Marc Joliot, Bernard Mazoyer, Nathalie Tzourio, Laurent Petit, Fabrice Crivello, Gaël Jobard, Pierre-Yves Hervé, Gaëlle Leroux

Ce document est la synthèse de la base « idéale » des chercheurs du GIN. Toutes les fonctionnalités décrites ci-après viennent s'ajouter aux fonctionnalités initialement présentes dans la base BIL&GIN.

Remarques générales

Pour les chercheurs du GIN, leurs intérêts d'utilisation d'une base de données sont de pouvoir :

- accéder et récupérer les données facilement ;
- savoir ce qui a été fait par le passé et pourquoi ;
- lancer les analyses depuis l'interface graphique de la base.

Au niveau de l'interface, il est souhaité :

- de privilégier les listes de valeurs aux champs textuels libres ;
- de ne pas mettre plusieurs informations dans un même champ ;
- qu'elle soit claire et que le modèle de données soit transparent pour l'utilisateur, en particulier concernant l'interrogation de la base.

Par ailleurs, le modèle de données doit rester suffisamment souple pour autoriser de futures évolutions.

Remarques spécifiques

A propos des sujets : Les utilisateurs doivent être en mesure d'ajouter des champs liés à la démographie au fur et à mesure des besoins. Il prendre en compte des liens familiaux, ainsi que conserver les questionnaires répondus par le sujet lui-même et par le(s) membre(s) de la famille considéré(s).

A propos des acquisitions : Le numéro de comité d'étique de l'étude doit figurer dans la base, lié à chaque acquisition, et référençant dans la base le document d'accord de consentement signé par le sujet. L'organisation et la présentation des données liées aux tests de comportements doivent être revues :

- les données brutes et post-traitées doivent figurer toutes les deux dans la base ;
- supprimer le fichier tdat et conserver le fichier edat (qui contient toutes les informations) ;
- présenter les informations du fichier edat sous la forme d'un tableau à double entrée.

A propos des pré-traitements :

- disposer de bibliothèques de traitements ;
- stocker les paramètres de traitement associés aux résultats ;
- renforcer le procédé de contrôle qualité des résultats en associant un statut à chaque run (qui sont des entités indépendantes) et en établissant un vocabulaire commun des critères de rejet d'une donnée.

A propos des analyses :

- lancement des analyses depuis la base : intégration de Matlab et SPM pour la base ;
- prévoir la gestion des analyses de groupe : stocker la question posée lors de la requête dans la base, les sujets impliqués au moment de cette requête, les outils utilisés et les résultats obtenus.

Bases EVA et 3C : La nouvelle base PLM doit contenir dans l'idéal les données des bases EVA et 3C.

Annexe D

Modèle de données BMI-LM

Document rédigé par Marianne Allanic et Arthur Grioche dans le cadre du projet BIOMIST.

Le modèle de donnée est défini de manière à supporter de façon efficace et précise les besoins en gestion des données des chercheurs en BMI[2]. Au travers de la documentation il sera explicité le choix du modèle, les objets qui le compose, les relations entre ceux-ci et enfin les outils permettant l'utilisation des données qu'ils contiennent.

1 COMMENT LIRE LA DOCUMENTATION ET VOCABULAIRE

1.1 LECTURE DE LA DOCUMENTATION MDD

Le document est organisé autour de 4 axes de lecture en se plaçant tour à tour du point de vue des objets du modèle, des types de dataset, des relations et en présentant les outils ajoutés afin de manipuler et voir les fichiers gérés dans le système PLM.

Les **objets** sont présentés à partir de leur utilisation tout au long du cycle de vie d'une étude. Dans ce document, le cycle BMI est séparé en 3 phases : Acquisition, Exploitation et Valorisation. Chaque objet est positionné par rapport au schéma UML global, puis ses caractéristiques sont présentées : description, attributs et relations avec les autres objets.

Les **datasets** existants sont présentés dans le document avec leurs extensions, les outils permettant de les ouvrir et une description de leur utilisation pour le BMI.

Les **relations** sont présentées par ordre alphabétique avec leurs caractéristiques comme pour les objets : description, attributs et relations qu'elles permettent de créer avec leurs cardinalités.

Une description des **outils** de consultation et d'édition des datasets est donnée à la fin du document.

1.2 ELEMENTS DE VOCABULAIRE

Eléments	Définitions
GRM Rules	Règle d'application des relations entre objets. Les relations sont orientées et possèdent une cardinalité.
Primary Object	Objet parent de la relation
Secondary Object	Objet enfant de la relation

[2] BMI : « Bio Medical Imaging », désigne le champ de recherche.

2 MODELISATION ET TYPES DE DONNEES

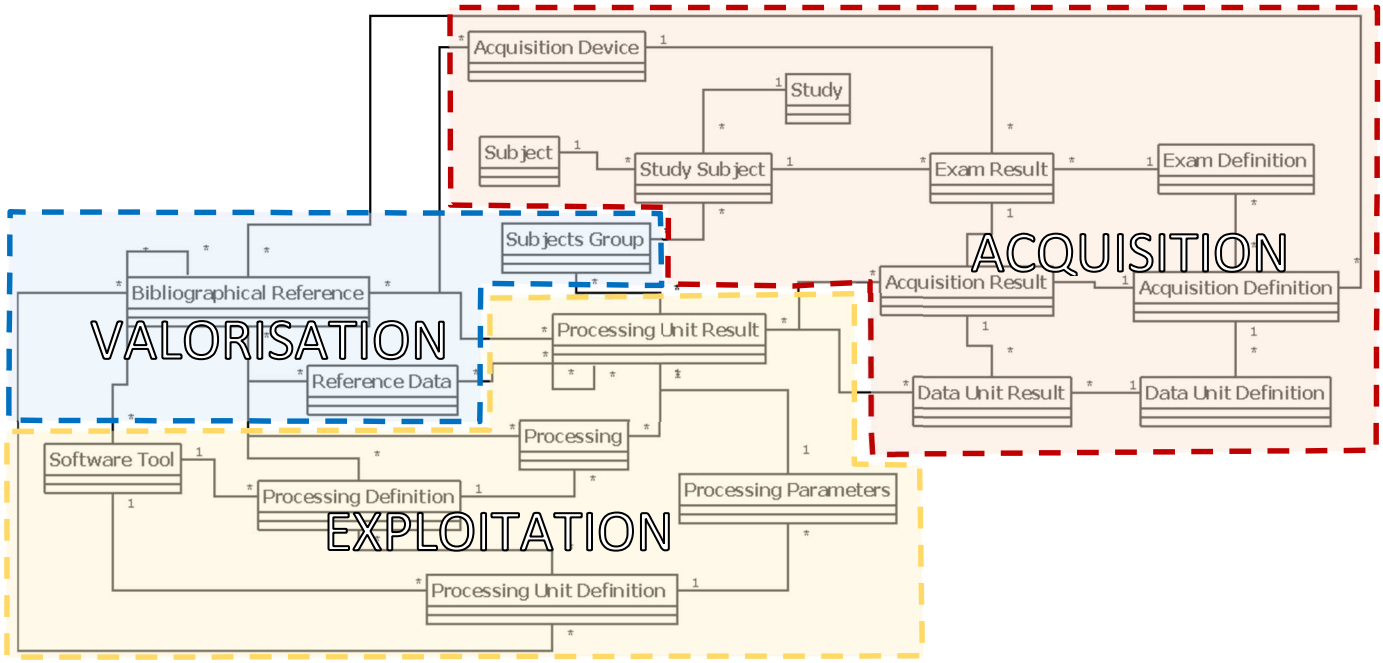


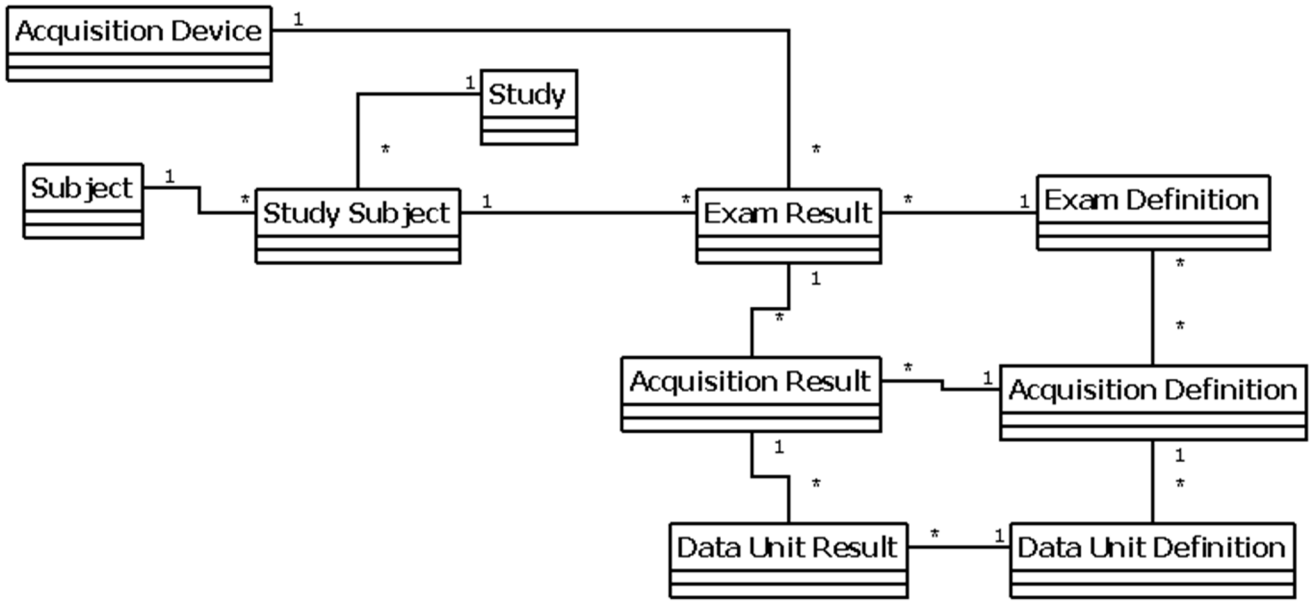
Figure 1: Modélisation UML du modèle de données BMI-LM

Le modèle de donnée BMI-LM présenté sous cette forme offre une vue complète des relations entre objets du modèle. Trois groupements de données d'un point de vue de l'utilisation des données sont mis en relief sur la modélisation UML. Les objets en rouge supportant les données d'acquisition, en jaune les données utilisées pour transcrire les opérations de traitement en phase d'exploitation et en bleu les données de référence.

Dans la suite du document les caractéristiques des objets du modèle de données sont présentées – description, attributs, relations. Les objets sont regroupés selon la distinction ci-dessus.

2.1 ACQUISITION

Les données d'acquisitions sont les données obtenues à la suite d'un ensemble de mesure prise sur les sujets. La structure et les relations de ces objets permettent de gérer la provenance des données pour pouvoir les réutiliser par la suite, tout en restant flexible pour permettre de futures évolutions.



className	parentClassName	isExportable	isUninstantiable	isUninheritable
GIN4_Study	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Study (étude)

Description : étude de recherche

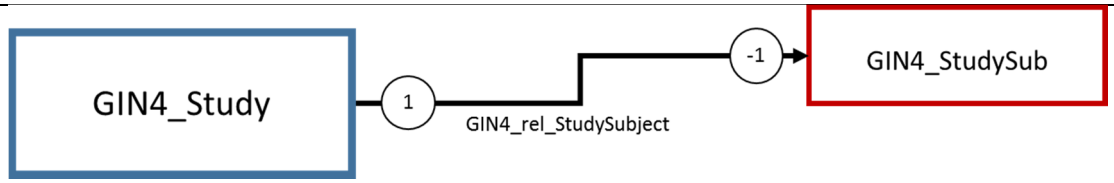
Attributs :

attributeName	description2	maxStringLength	isArray	followOnExport	isNullsAllowed	isUnique
gin4_idnumber	id number	0	FAUX	FAUX	VRAI	FAUX

isPublicRead	isPublicWrite	isCandidateKey	isTransient	exportAsString	noBackpointer	initialValue
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	0

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_Study	1	GIN4_rel_StudySubject	-1	GIN4_StudySub



GIN4_StudySub	GIN4_Item_NRev	FAUX	FAUX	FAUX

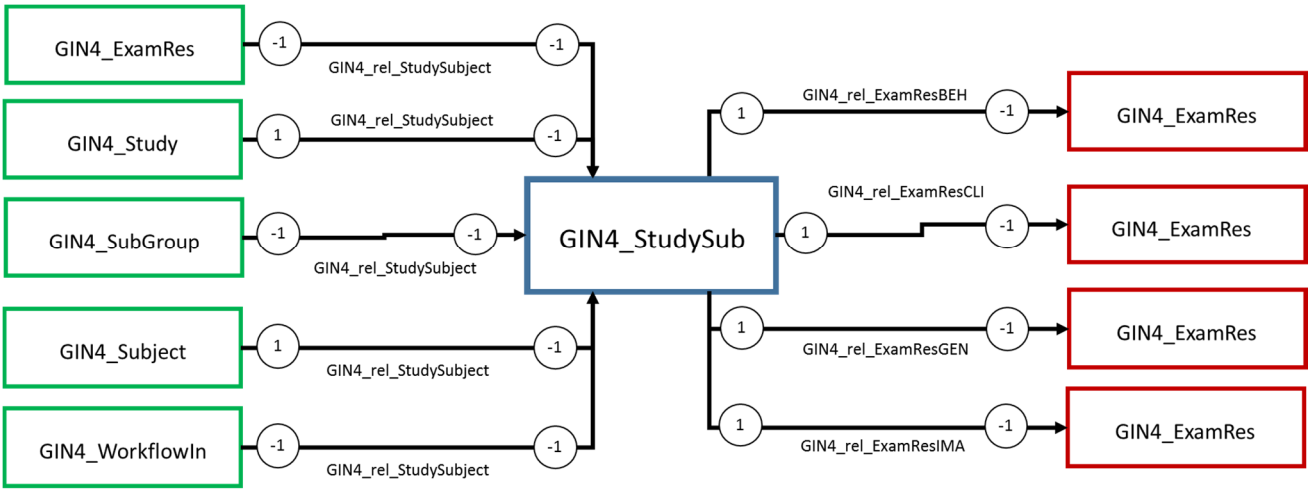
Nom d'affichage : Study Subject (sujet dans l'étude)

Description : sujet dans le contexte d'une étude

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Study	1	GIN4_rel_StudySubject	-1	**GIN4_StudySub**
GIN4_SubGroup	-1	GIN4_rel_StudySubject	-1	**GIN4_StudySub**
GIN4_Subject	1	GIN4_rel_StudySubject	-1	**GIN4_StudySub**
GIN4_ExamRes	-1	GIN4_rel_StudySubject	-1	**GIN4_StudySub**
GIN4_WorkflowIn	-1	GIN4_rel_StudySubject	-1	**GIN4_StudySub**
GIN4_StudySub	1	GIN4_rel_ExamResBEH	-1	GIN4_ExamRes
GIN4_StudySub	1	GIN4_rel_ExamResCLI	-1	GIN4_ExamRes
GIN4_StudySub	1	GIN4_rel_ExamResGEN	-1	GIN4_ExamRes
GIN4_StudySub	1	GIN4_rel_ExamResIMA	-1	GIN4_ExamRes



GIN4_Subject	GIN4_Item_NRev	FAUX	FAUX	FAUX

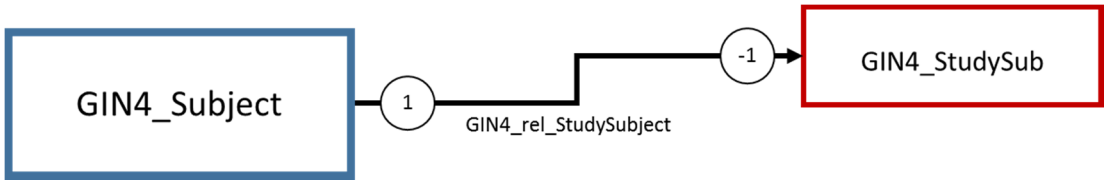
Nom d'affichage : Subject (sujet)

Description : sujet unique dans la base

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Subject	1	GIN4_rel_StudySubject	-1	GIN4_StudySub



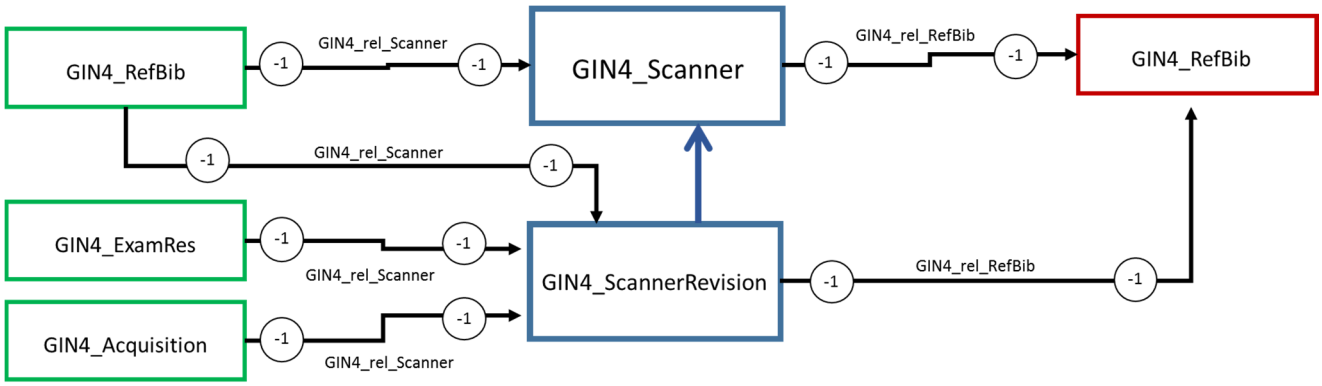
GIN4_Scanner	GIN4_Item_Rev	FAUX	FAUX	FAUX

Nom d'affichage : Acquisition Device (dispositif d'acquisition)
Description : Description d'un dispositif utilisé pendant un examen

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Scanner	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_ScannerRevision	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_RefBib	-1	GIN4_rel_Scanner	-1	**GIN4_Scanner**
GIN4_ExamRes	-1	GIN4_rel_Scanner	-1	**GIN4_ScannerRevision**
GIN4_RefBib	-1	GIN4_rel_Scanner	-1	**GIN4_ScannerRevision**
GIN4_Acquisition	-1	GIN4_rel_Scanner	-1	**GIN4_ScannerRevision**



GIN4_ExamRes	GIN4_Item_NRev	FAUX	FAUX	FAUX

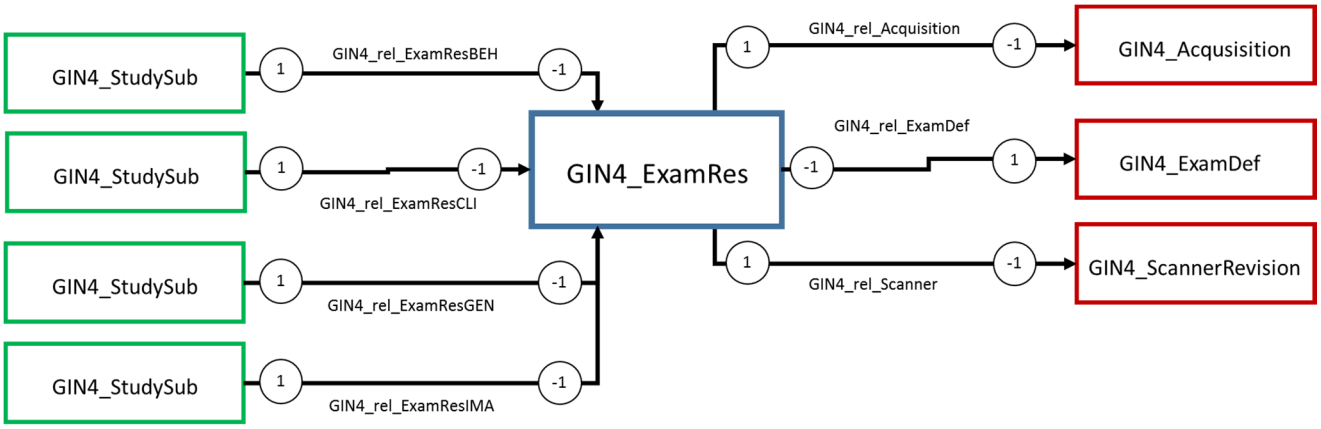
Nom d'affichage : Exam Result (Résultat d'examen)

Description : ligne continue d'acquisitions

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	1	GIN4_rel_Acquisition	-1	GIN4_Acquisition
GIN4_ExamRes	-1	GIN4_rel_ExamDef	1	GIN4_ExamDef
GIN4_ExamRes	-1	GIN4_rel_Scanner	-1	GIN4_ScannerRevision
GIN4_StudySub	1	GIN4_rel_ExamResBEH	-1	**GIN4_ExamRes**
GIN4_StudySub	1	GIN4_rel_ExamResCLI	-1	**GIN4_ExamRes**
GIN4_StudySub	1	GIN4_rel_ExamResGEN	-1	**GIN4_ExamRes**
GIN4_StudySub	1	GIN4_rel_ExamResIMA	-1	**GIN4_ExamRes**



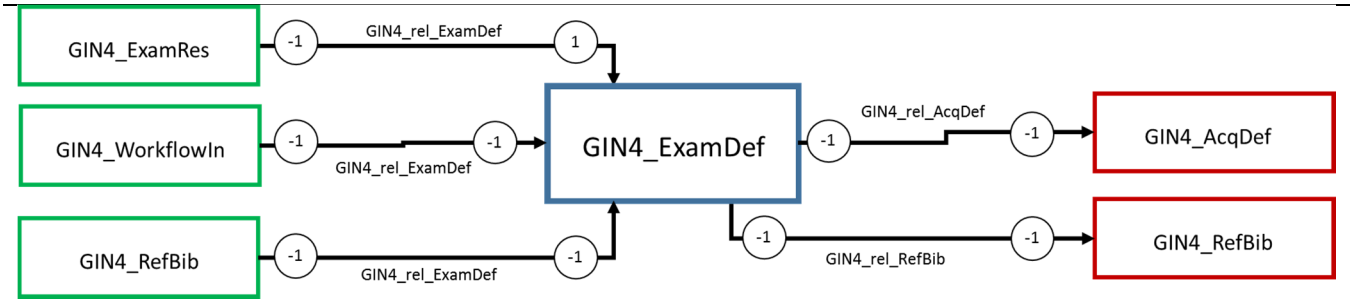
GIN4_ExamDef	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Exam Definition (définition d'un examen)
Description : Description de la chaîne d'acquisitions

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	-1	GIN4_rel_ExamDef	1	**GIN4_ExamDef**
GIN4_WorkflowIn	-1	GIN4_rel_ExamDef	-1	**GIN4_ExamDef**
GIN4_RefBib	-1	GIN4_rel_ExamDef	-1	**GIN4_ExamDef**
GIN4_ExamDef	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_ExamDef	-1	GIN4_rel_AcqDef	-1	GIN4_AcqDef



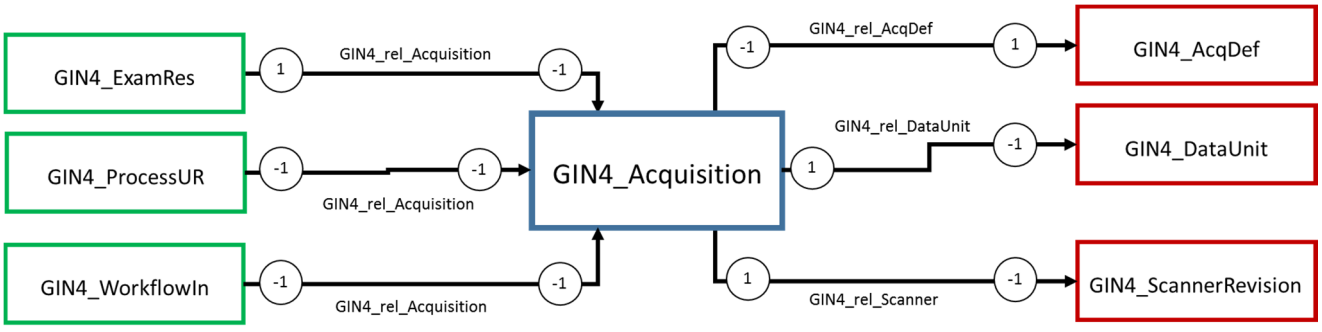
GIN4_Acquisition	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Acquisition Result (résultat d'acquisition)
Description : Période indivisible d'acquisition de données

Attributs : Aucun

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	1	GIN4_rel_Acquisition	-1	**GIN4_Acquisition**
GIN4_ProcessUR	-1	GIN4_rel_Acquisition	-1	**GIN4_Acquisition**
GIN4_WorkflowIn	-1	GIN4_rel_Acquisition	-1	**GIN4_Acquisition**
GIN4_Acquisition	-1	GIN4_rel_AcqDef	1	GIN4_AcqDef
GIN4_Acquisition	1	GIN4_rel_DataUnit	-1	GIN4_DataUnit
GIN4_Acquisition	-1	GIN4_rel_Scanner	-1	GIN4_ScannerRevision



GIN4_AcqDef	GIN4_Item_NRev	FAUX	FAUX	FAUX

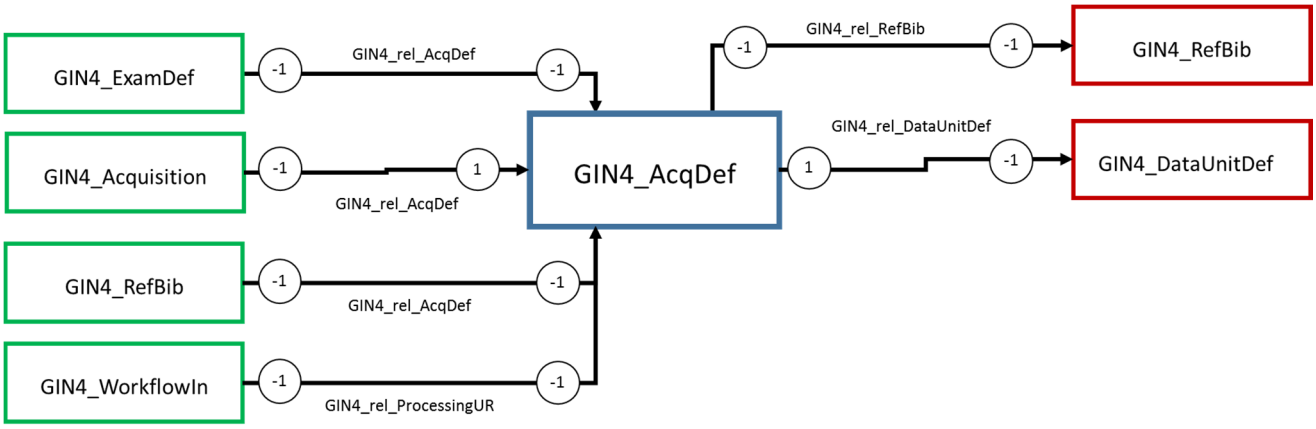
Nom d'affichage : Acquisition Definition (définition d'une acquisition)
Description : Description d'un dispositif utilisé pendant un examen

Attributs : Aucun

Relations:

Primary	Card.	Relation	Card.	Secondary

GIN4_AcqDef	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_AcqDef	1	GIN4_rel_DataUnitDef	-1	GIN4_DataUnitDef
GIN4_RefBib	-1	GIN4_rel_AcqDef	-1	**GIN4_AcqDef**
GIN4_WorkflowIn	-1	GIN4_rel_AcqDef	-1	**GIN4_AcqDef**
GIN4_Acquisition	-1	GIN4_rel_AcqDef	1	**GIN4_AcqDef**
GIN4_ExamDef	-1	GIN4_rel_AcqDef	-1	**GIN4_AcqDef**



GIN4_DataUnit	GIN4_Item_NRev	FAUX	FAUX	FAUX

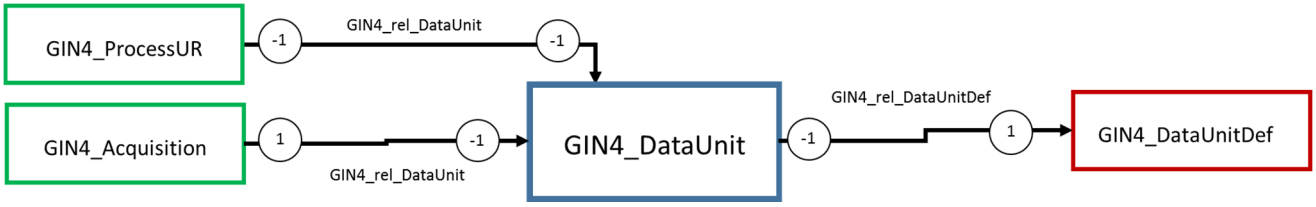
Nom d'affichage : Data Unit Result (unité de données)

Description : Donnée acquise isolée

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessUR	-1	GIN4_rel_DataUnit	-1	**GIN4_DataUnit**
GIN4_Acquisition	1	GIN4_rel_DataUnit	-1	**GIN4_DataUnit**
GIN4_DataUnit	-1	GIN4_rel_DataUnitDef	1	GIN4_DataUnitDef



GIN4_DataUnitDef	GIN4_Item_NRev	FAUX	FAUX	FAUX

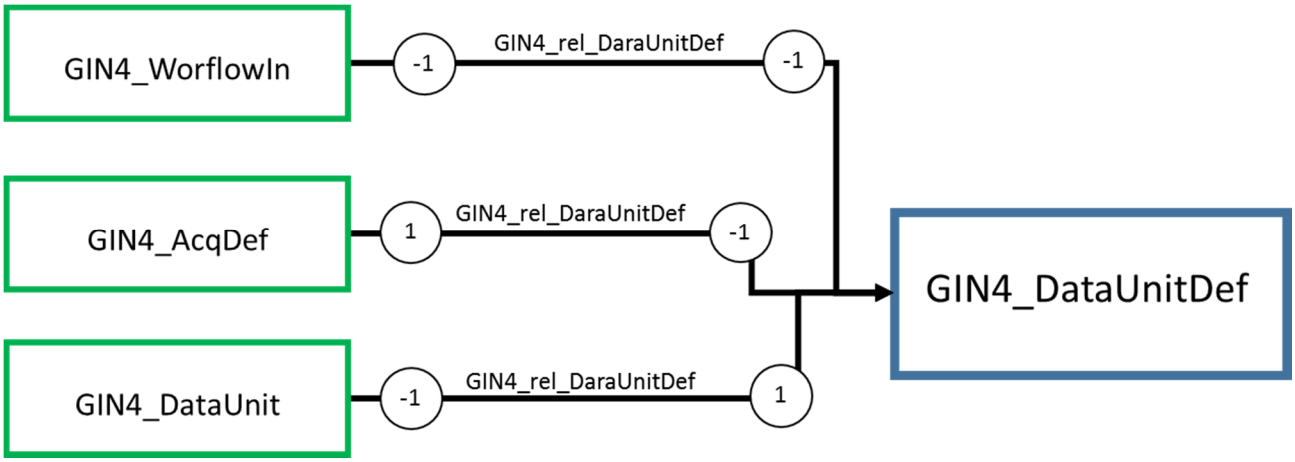
Nom d'affichage : Data Unit Definition (definition d'une unité de données)

Description : description d'une unité de données

Attributs : Aucun

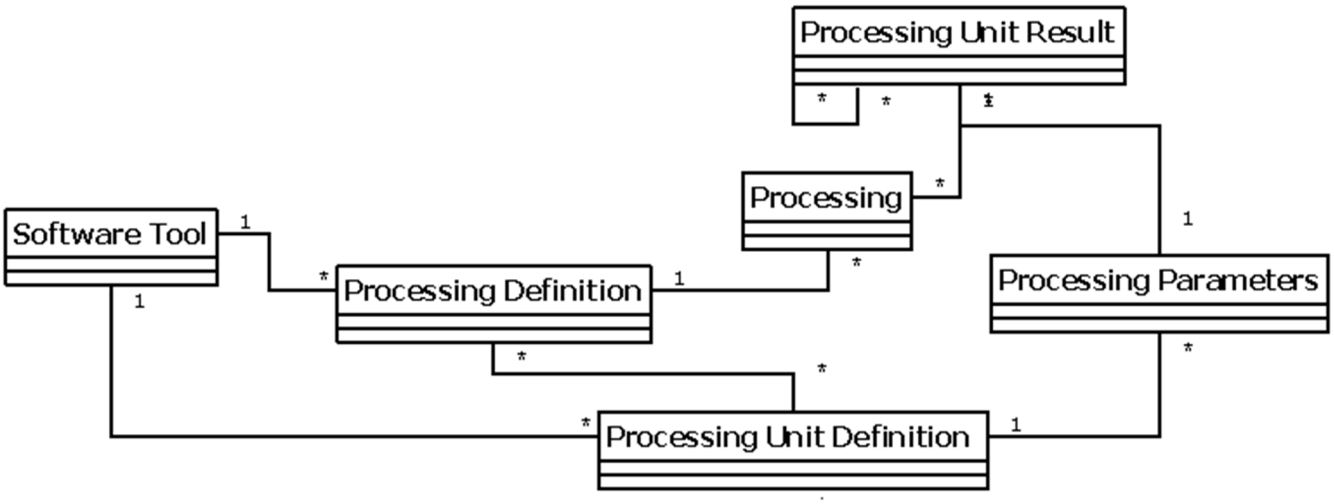
Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_WorkflowIn	-1	GIN4_rel_DataUnitDef	-1	**GIN4_DataUnitDef**
GIN4_AcqDef	1	GIN4_rel_DataUnitDef	-1	**GIN4_DataUnitDef**
GIN4_DataUnit	-1	GIN4_rel_DataUnitDef	1	**GIN4_DataUnitDef**



2.2 EXPLOITATION

Les objets d'exploitation sont utilisés afin de retranscrire les chaines de traitement effectuées avec et sur les données.



className	parentClassName	isExportable	isUninstantiable	isUninheritable
GIN4_ProcessRes	GIN4_Item_NRev	FAUX	FAUX	FAUX

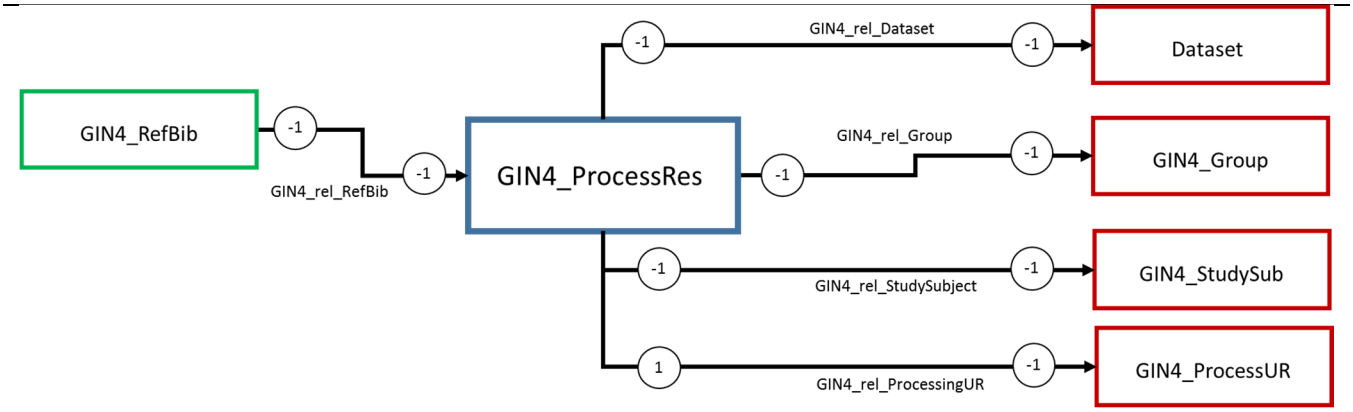
Nom d'affichage : Processing Result (chaîne de traitement)

Description : Chaîne de traitement

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessRes	-1	GIN4_rel_Dataset	-1	Dataset
GIN4_ProcessRes	-1	GIN4_rel_Group	-1	GIN4_Group
GIN4_ProcessRes	1	GIN4_rel_ProcessingUR	-1	GIN4_ProcessUR
GIN4_ProcessRes	-1	GIN4_rel_StudySubject	-1	GIN4_StudySub
GIN4_RefBib	-1	GIN4_rel_ProcessingRes	-1	**GIN4_ProcessRes**



GIN4_ProcessDef	GIN4_Item_NRev	FAUX	FAUX	FAUX

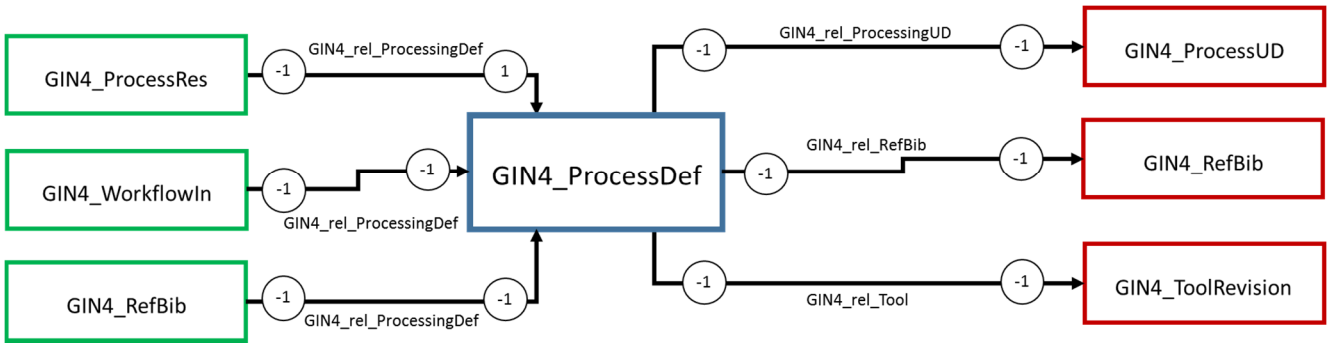
Nom d'affichage : Processing Definition (définition d'une chaîne de traitement)

Description : Description d'une chaîne de traitement

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessDef	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_ProcessDef	-1	GIN4_rel_Tool	-1	GIN4_ToolRevision
GIN4_ProcessDef	-1	GIN4_rel_ProcessingUD	-1	GIN4_ProcessUD
GIN4_ProcessRes	-1	GIN4_rel_ProcessingDef	1	**GIN4_ProcessDef**
GIN4_RefBib	-1	GIN4_rel_ProcessingDef	-1	**GIN4_ProcessDef**
GIN4_WorkflowIn	-1	GIN4_rel_ProcessingDef	-1	**GIN4_ProcessDef**



GIN4_ProcessUR	GIN4_Item_NRev	FAUX	FAUX	FAUX

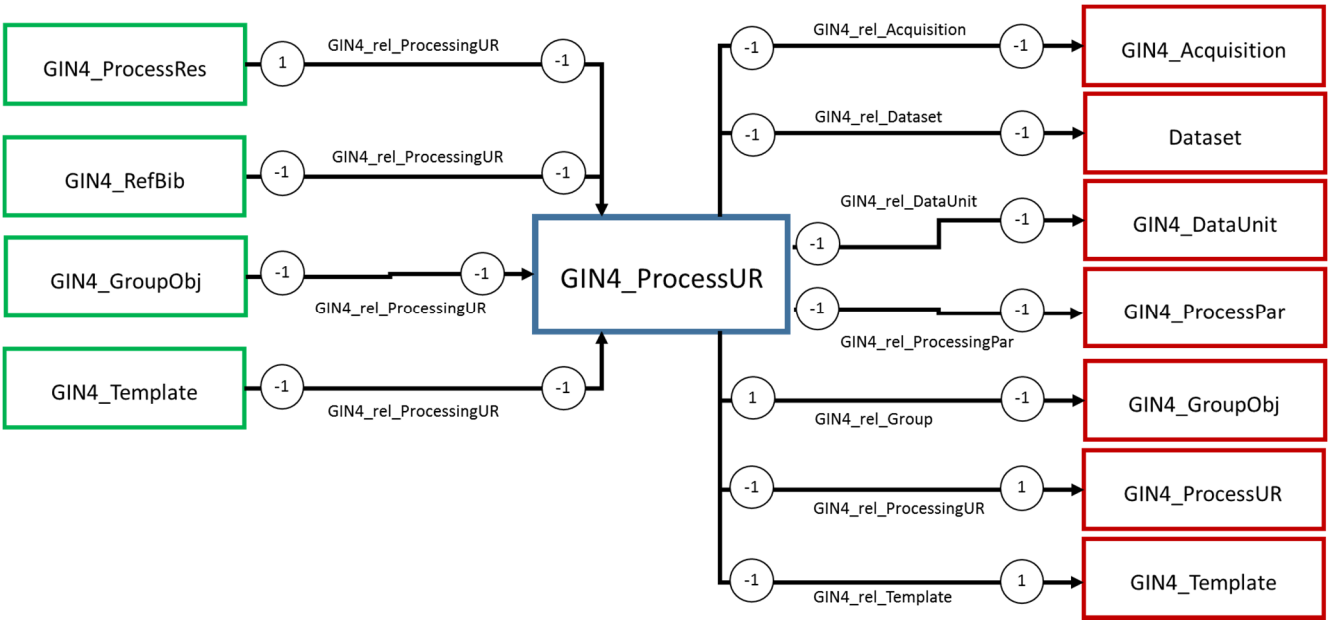
Nom d'affichage : Processing Unit Result (unité de traitement)

Description : Traitement effectué sur des données

Attributs : Aucun

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessUR	-1	GIN4_rel_Acquisition	-1	GIN4_Acquisition
GIN4_ProcessUR	-1	GIN4_rel_Dataset	-1	Dataset
GIN4_ProcessUR	-1	GIN4_rel_DataUnit	-1	GIN4_DataUnit
GIN4_ProcessUR	1	GIN4_rel_Group	-1	GIN4_GroupObj
GIN4_ProcessUR	-1	GIN4_rel_ProcessingPar	1	GIN4_ProcessPar
GIN4_ProcessUR	-1	GIN4_rel_ProcessingUR	-1	GIN4_ProcessUR
GIN4_ProcessUR	-1	GIN4_rel_Template	-1	GIN4_Template
GIN4_ProcessRes	1	GIN4_rel_ProcessingUR	-1	**GIN4_ProcessUR**
GIN4_RefBib	-1	GIN4_rel_ProcessingUR	-1	**GIN4_ProcessUR**
GIN4_GroupObj	-1	GIN4_rel_ProcessingUR	-1	**GIN4_ProcessUR**
GIN4_Template	-1	GIN4_rel_ProcessingUR	-1	**GIN4_ProcessUR**



GIN4_ProcessUD	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Processing Unit Definition (définition d'une chaîne de traitement)

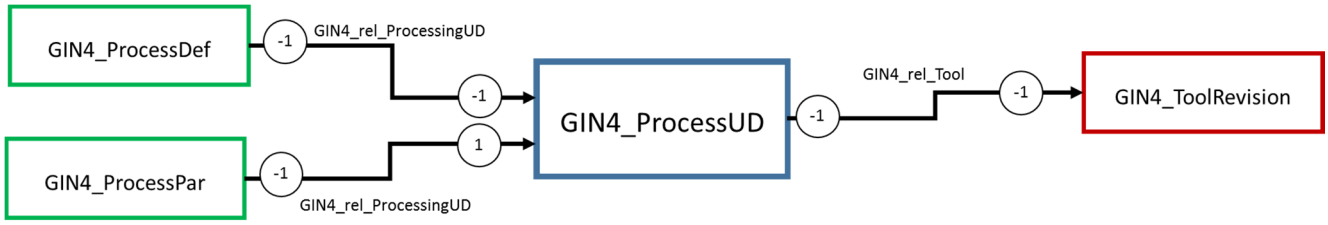
Description : Description d'une chaîne de traitement

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessUD	-1	GIN4_rel_Tool	-1	GIN4_ToolRevision
GIN4_ProcessDef	-1	GIN4_rel_ProcessingUD	-1	**GIN4_ProcessUD**

GIN4_ProcessPar	-1	GIN4_rel_ProcessingUD	1	**GIN4_ProcessUD**



GIN4_ProcessPar	GIN4_Item_NRev	FAUX	FAUX	FAUX

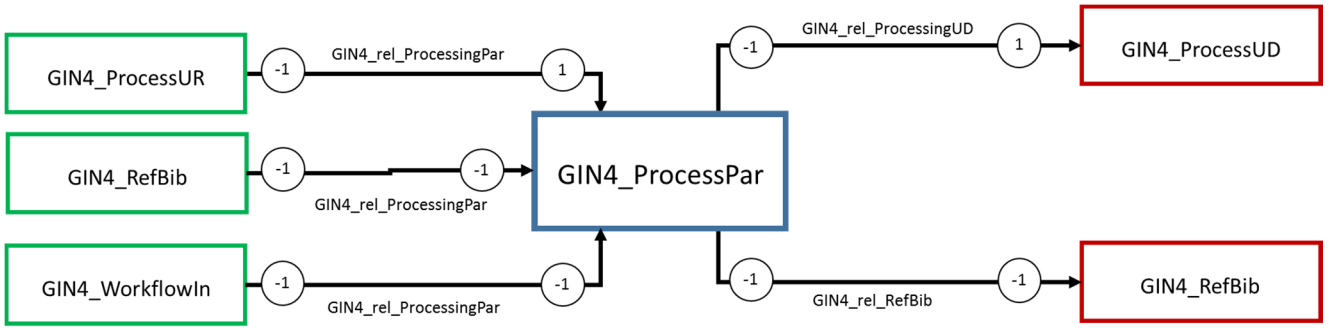
Nom d'affichage : Processing Parameter (Paramètres de traitement)

Description : Jeu de paramètres utilises pour un traitement

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessPar	-1	GIN4_rel_ProcessingUD	1	GIN4_ProcessUD
GIN4_ProcessPar	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_ProcessUR	-1	GIN4_rel_ProcessingPar	1	**GIN4_ProcessPar**
GIN4_RefBib	-1	GIN4_rel_ProcessingPar	-1	**GIN4_ProcessPar**
GIN4_WorkflowIn	-1	GIN4_rel_ProcessingPar	-1	**GIN4_ProcessPar**



GIN4_Tool	GIN4_Item_Rev	FAUX	FAUX	FAUX

Nom d'affichage : Software tool (logiciel)

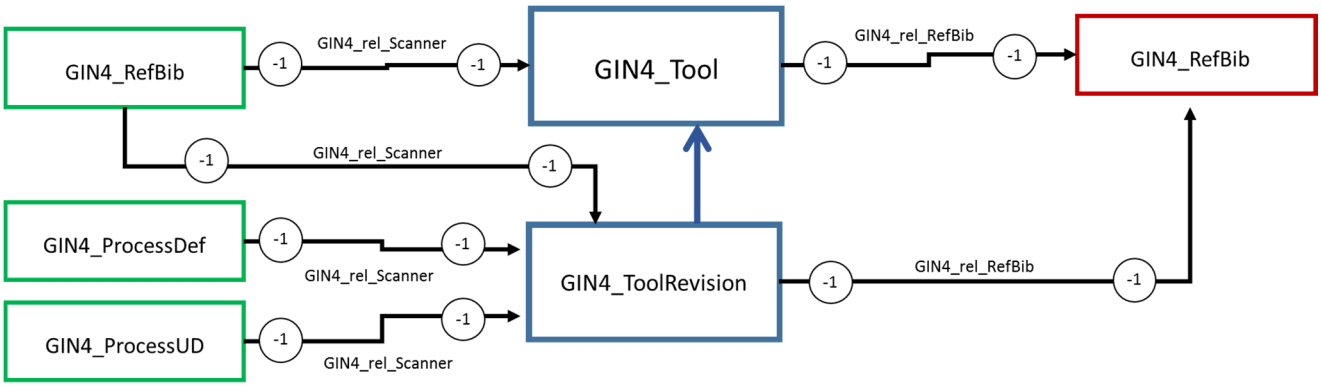
Description : Description d'un logiciel de traitement

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Tool	-1	GIN4_rel_RefBib	-1	GIN4_RefBib

GIN4_ToolRevision	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_ProcessDef	-1	GIN4_rel_Tool	-1	**GIN4_ToolRevision**
GIN4_ProcessUD	-1	GIN4_rel_Tool	-1	**GIN4_ToolRevision**
GIN4_RefBib	-1	GIN4_rel_Tool	-1	**GIN4_Tool**
GIN4_RefBib	-1	GIN4_rel_Tool	-1	**GIN4_ToolRevision**



GIN4_WorkflowIn	GIN4_Item_NRev	FAUX	FAUX	FAUX

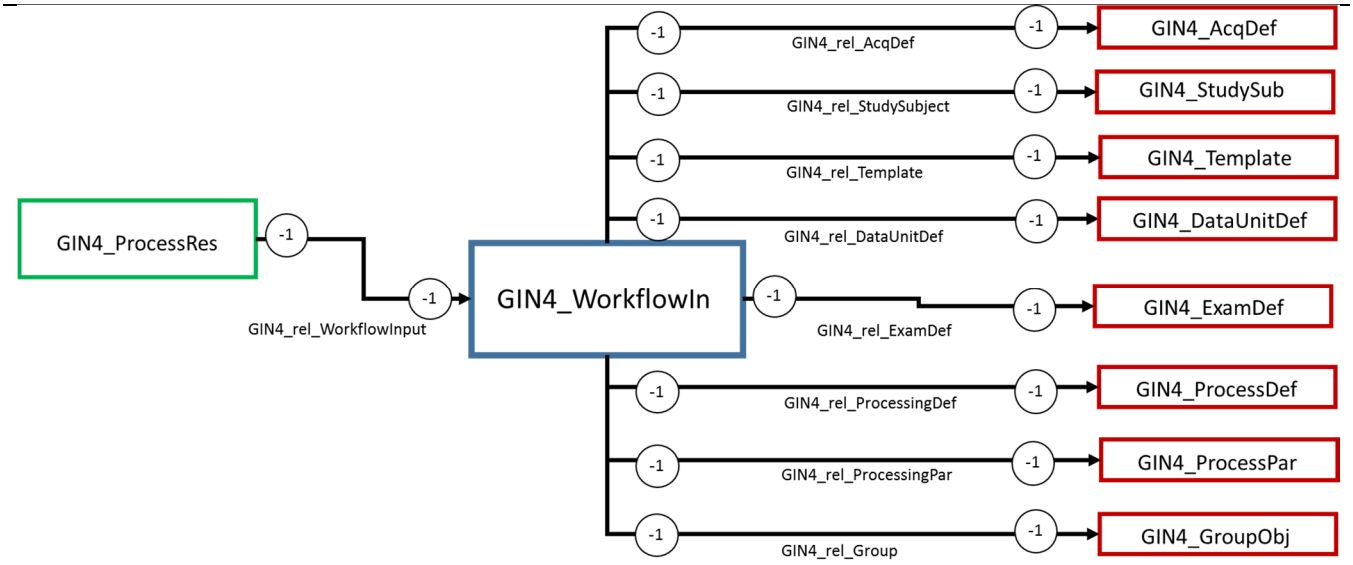
Nom d'affichage : WorkflowIn (entrées du workflow)

Description : Description des entrées d'un workflow (données d'entrée, objets de definition et de reference)

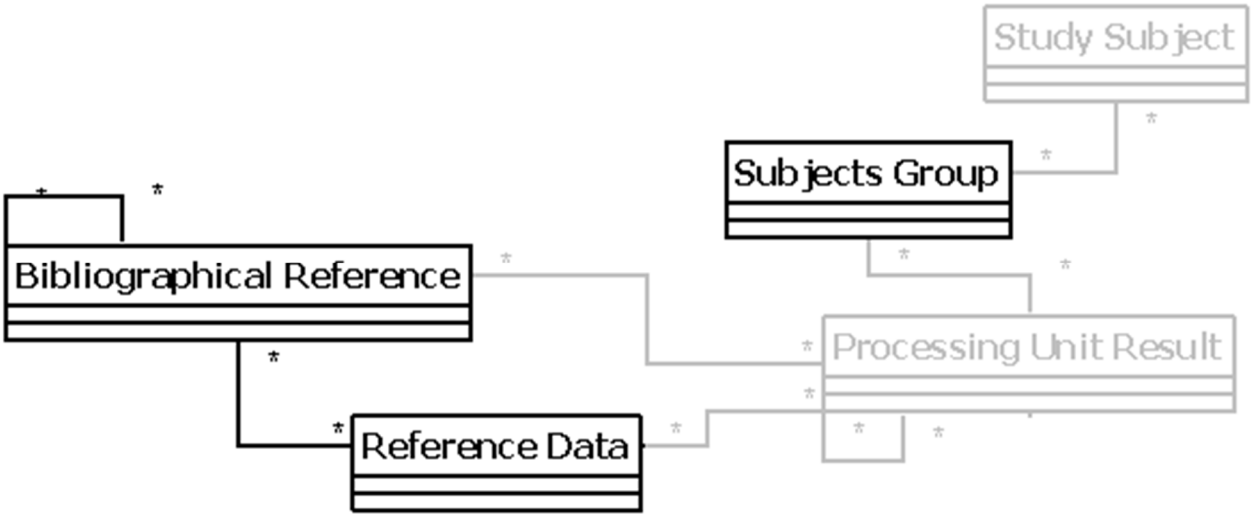
Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_WorkflowIn	-1	GIN4_rel_AcqDef	-1	GIN4_AcqDef
GIN4_WorkflowIn	-1	GIN4_rel_StudySubject	-1	GIN4_StudySub
GIN4_WorkflowIn	-1	GIN4_rel_DataUnitDef	-1	GIN4_DataUnitDef
GIN4_WorkflowIn	-1	GIN4_rel_ExamDef	-1	GIN4_ExamDef
GIN4_WorkflowIn	-1	GIN4_rel_Group	-1	GIN4_GroupObj
GIN4_WorkflowIn	-1	GIN4_rel_ProcessingDef	-1	GIN4_ProcessDef
GIN4_WorkflowIn	-1	GIN4_rel_ProcessingPar	-1	GIN4_ProcessPar
GIN4_WorkflowIn	-1	GIN4_rel_Template	-1	GIN4_Template
GIN4_ProcessRes	-1	GIN4_rel_WorkflowInput	-1	**GIN4_WorkflowIn**


GIN4_ProcessRes
-1
GIN4_rel_WorkflowInput
-1
GIN4_WorkflowIn
-1
GIN4_rel_AcqDef
-1
GIN4_AcqDef
-1
GIN4_rel_StudySubject
-1
GIN4_StudySub
-1
GIN4_rel_Template
-1
GIN4_Template
-1
GIN4_rel_DataUnitDef
-1
GIN4_DataUnitDef
-1
GIN4_rel_ExamDef
-1
GIN4_ExamDef
-1
GIN4_rel_ProcessingDef
-1
GIN4_ProcessDef
-1
GIN4_rel_ProcessingPar
-1
GIN4_ProcessPar
-1
GIN4_rel_Group
-1
GIN4_GroupObj

2.3 VALORISATION



- Les objets description et attributs
- Les relations : GRM Rules

className	parentClassName	isExportable	isUninstantiable	isUninheritable
GIN4_SubGroup	GIN4_GroupObj	FAUX	FAUX	FAUX

Nom d'affichage : Subjects Group (groupe de sujets – dans l'étude)

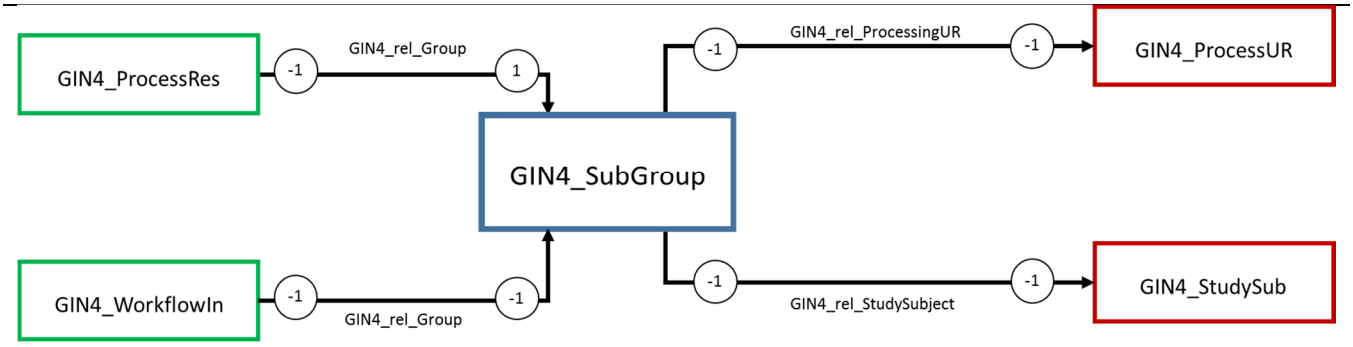
Description : Ensemble de sujets dans l'étude regroupés selon un critère (brut ou dérivé)

Attributs : Aucun

(*)Le group est parent du SubGroup il peut donc être utilisé dans les cas suivants

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Group*	-1	GIN4_rel_ProcessingUR	-1	GIN4_ProcessUR
GIN4_SubGroup	-1	GIN4_rel_StudySubject	-1	GIN4_StudySub
GIN4_ProcessRes	-1	GIN4_rel_Group	-1	**GIN4_Group***
GIN4_WorkflowIn	-1	GIN4_rel_Group	-1	**GIN4_Group***



GIN4_RefBib	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Bibliographical Reference (référence bibliographique)

Description : Article scientifique

Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_Author	author	32	FAUX	FAUX	VRAI	FAUX
isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_Title	title	64	FAUX	FAUX	VRAI	FAUX
isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

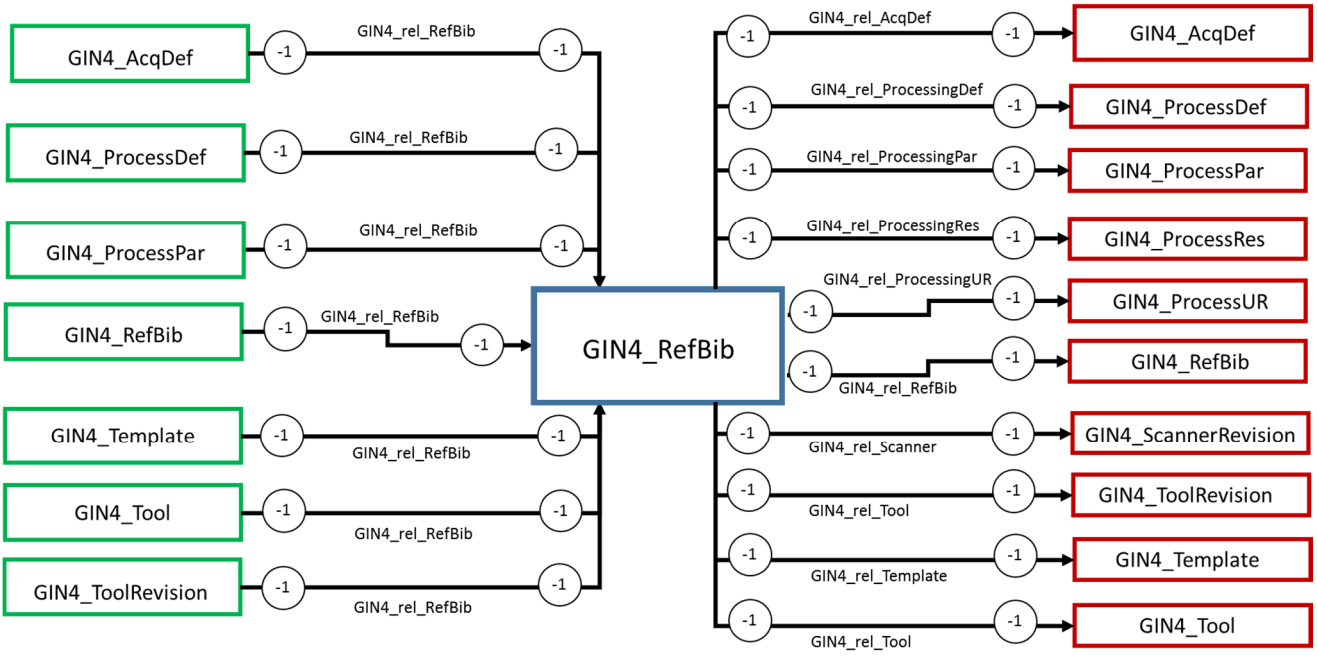
attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_Type	type of reference	32	FAUX	FAUX	VRAI	FAUX
isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_ Comments	comments	256	FAUX	FAUX	VRAI	FAUX
isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_RefBib	-1	GIN4_rel_AcqDef	-1	GIN4_AcqDef
GIN4_RefBib	-1	GIN4_rel_ProcessingDef	-1	GIN4_ProcessDef
GIN4_RefBib	-1	GIN4_rel_ProcessingPar	-1	GIN4_ProcessPar
GIN4_RefBib	-1	GIN4_rel_ProcessingRes	-1	GIN4_ProcessRes

GIN4_RefBib	-1	GIN4_rel_ProcessingUR	-1	GIN4_ProcessUR
GIN4_RefBib	-1	GIN4_rel_RefBib	-1	GIN4_RefBib
GIN4_RefBib	-1	GIN4_rel_Scanner	-1	GIN4_ScannerRevision
GIN4_RefBib	-1	GIN4_rel_Template	-1	GIN4_Template
GIN4_RefBib	-1	GIN4_rel_Tool	-1	GIN4_Tool
GIN4_RefBib	-1	GIN4_rel_Tool	-1	GIN4_ToolRevision
GIN4_AcqDef	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_ProcessDef	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_ProcessPar	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_RefBib	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_Template	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_Tool	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**
GIN4_ToolRevision	-1	GIN4_rel_RefBib	-1	**GIN4_RefBib**



GIN4_Template	GIN4_Item_NRev	FAUX	FAUX	FAUX

Nom d'affichage : Reference data (donnée de référence)

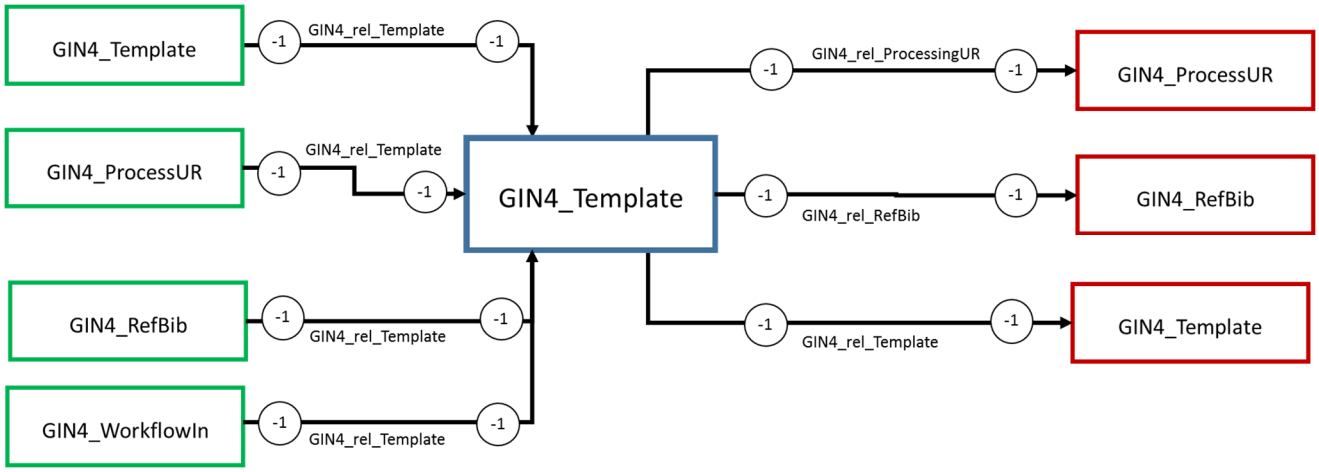
Description : Donnée d'entrée d'un traitement hors du context d'une étude

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Template	-1	GIN4_rel_ProcessingUR	-1	GIN4_ProcessUR
GIN4_Template	-1	GIN4_rel_RefBib	-1	GIN4_RefBib

GIN4_Template	-1	GIN4_rel_Template	-1	**GIN4_Template**
GIN4_ProcessUR	-1	GIN4_rel_Template	-1	**GIN4_Template**
GIN4_RefBib	-1	GIN4_rel_Template	-1	**GIN4_Template**
GIN4_WorkflowIn	-1	GIN4_rel_Template	-1	**GIN4_Template**



3 DATASETS

- Présenter les datasets

Nom	Nom Affichage	Extension	Outils	Descriptif
GIN4_2CSV	CSV	*.csv	MSExcel, TextEditor	
GIN4_2EMRG	MRG	*.emrg	GIN4_FSLVIEW	
GIN4_2GEXF	GEXF	*.gexf	GIN4_GEPHI	
GIN4_2HDR	HDR	*.hdr	GIN4_MICRON	
GIN4_2IMG	IMG	*.img	GIN4_MICRON	
GIN4_2JGEX	JGEX	*.jgex , *.json	TextEditor	
GIN4_2JMP	JMP	*.jmp	MSExcel	
GIN4_2MAT	MAT	*.m, *.mat	GIN4_Matlab	
GIN4_2NII	NII	*.nii	GIN4_MICRON	
GIN4_GZ	GZ	*.gz	GIN4_UNZIP	

4 RELATIONS

Les GRMrules associées à chaque relation sont présentées dans cette partie. Sur les figures, les objets primaires sont à gauche et les objets secondaires sont à droite avec leurs cardinalités.

className	parentClassName	isExportable	isUninstantiable	isUninheritable
GIN4_rel_AcqDef	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relAcquisitionDefinition

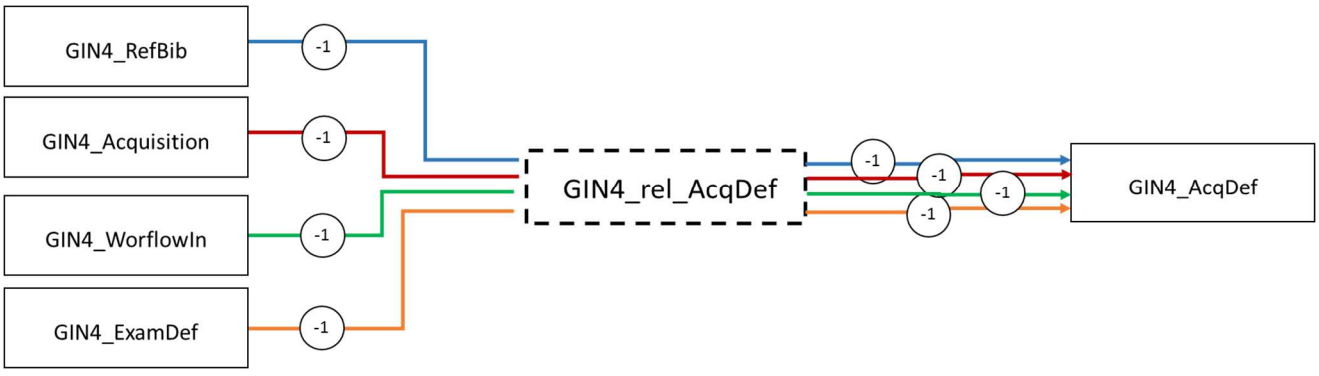
Description :

Attributs :

attributeName	description2	maxStringLength	isArray	followOnExport	isNullsAllowed	isUnique
		POM_string	256	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_RefBib	-1	**GIN4_rel_AcqDef**	-1	GIN4_AcqDef
GIN4_WorkflowIn	-1	**GIN4_rel_AcqDef**	-1	GIN4_AcqDef
GIN4_Acquisition	-1	**GIN4_rel_AcqDef**	1	GIN4_AcqDef
GIN4_ExamDef	-1	**GIN4_rel_AcqDef**	-1	GIN4_AcqDef



GIN4_rel_Acquisition	ImanRelation	FAUX	FAUX	FAUX

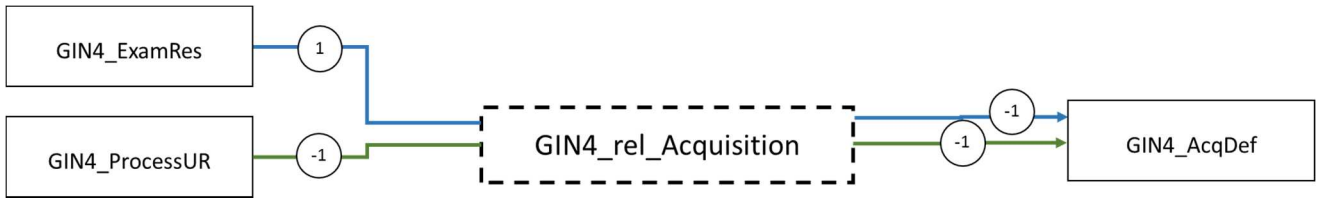
Nom d'affichage : relAcquisitionResult

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	1	**GIN4_rel_Acquisition**	-1	GIN4_Acquisition
GIN4_ProcessUR	-1	**GIN4_rel_Acquisition**	-1	GIN4_Acquisition



GIN4_rel_DataUnit	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relDataUnitResult

Description :

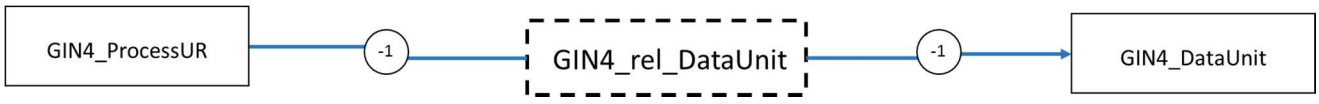
Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_role		POM_string	256	FAUX	FAUX	VRAI

isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessUR	-1	**GIN4_rel_DataUnit**	-1	GIN4_DataUnit



GIN4_rel_DataUnitDef	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relDataUnitDefinition

Description :

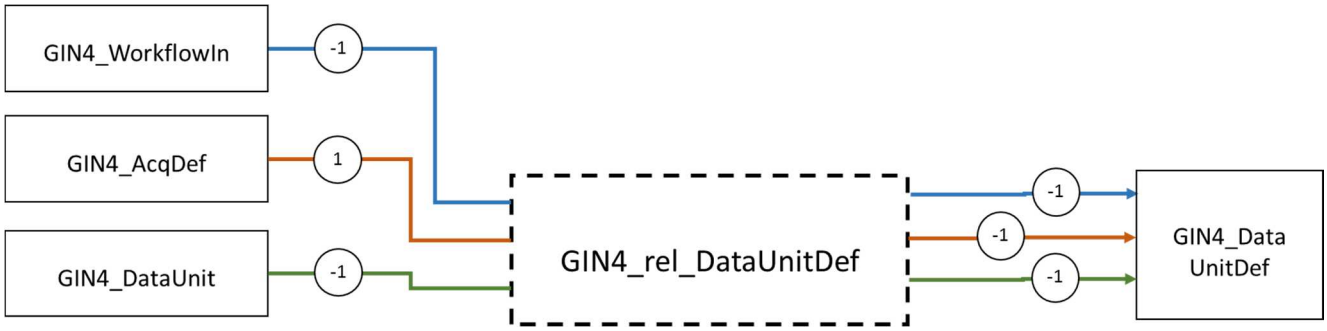
Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
gin4_role		POM_string	256	FAUX	FAUX	VRAI

isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_WorkflowIn	-1	**GIN4_rel_DataUnitDef**	-1	GIN4_DataUnitDef
GIN4_AcqDef	1	**GIN4_rel_DataUnitDef**	-1	GIN4_DataUnitDef
GIN4_DataUnit	-1	**GIN4_rel_DataUnitDef**	1	GIN4_DataUnitDef



GIN4_rel_Dataset	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relDataset

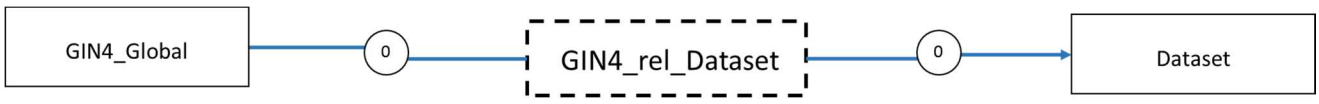
Description :

Attributs :

attributeName	description2	maxStringLength	isArray	followOnExport	isNullsAllowed	isUnique
		POM_string	64	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Global	0	**GIN4_rel_Dataset**	0	Dataset



GIN4_rel_ExamDef	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relExamDefinition

Description :

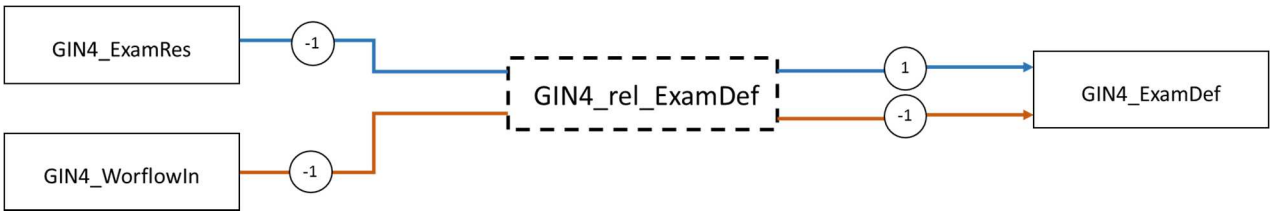
Attributs :

attributeName	description2	maxStringLength	isArray	followOnExport	isNullsAllowed	isUnique

gin4_role		POM_string	256	FAUX	FAUX	VRAI	
	isPublicRead	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	-1	**GIN4_rel_ExamDef**	1	GIN4_ExamDef
GIN4_WorkflowIn	-1	**GIN4_rel_ExamDef**	-1	GIN4_ExamDef



GIN4_rel_ExamRes	ImanRelation	FAUX	FAUX	FAUX

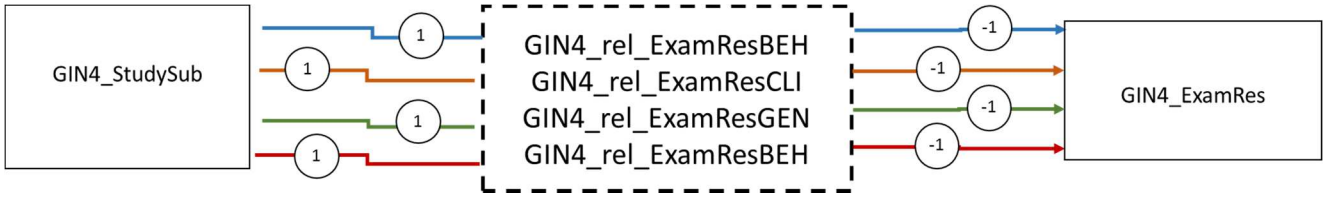
Nom d'affichage : relExamResult

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_StudySub	1	**GIN4_rel_ExamResBEH**	-1	GIN4_ExamRes
GIN4_StudySub	1	**GIN4_rel_ExamResCLI**	-1	GIN4_ExamRes
GIN4_StudySub	1	**GIN4_rel_ExamResGEN**	-1	GIN4_ExamRes
GIN4_StudySub	1	**GIN4_rel_ExamResBEH**	-1	GIN4_ExamRes



GIN4_rel_Group	ImanRelation	FAUX	FAUX	FAUX

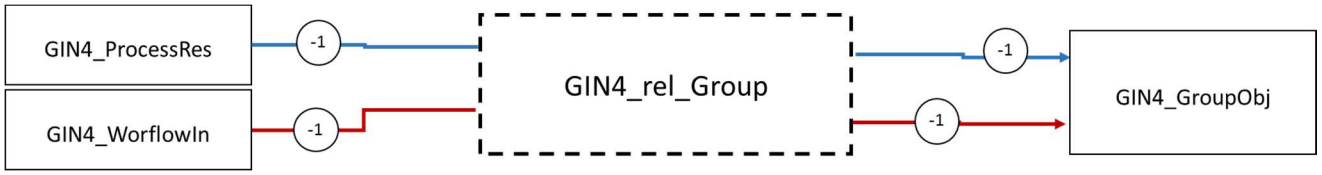
Nom d'affichage : relGroup

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessRes	-1	**GIN4_rel_Group**	-1	GIN4_GroupObj
GIN4_WorkflowIn	-1	**GIN4_rel_Group**	-1	GIN4_GroupObj



GIN4_rel_ProcessingDef	ImanRelation	FAUX	FAUX	FAUX

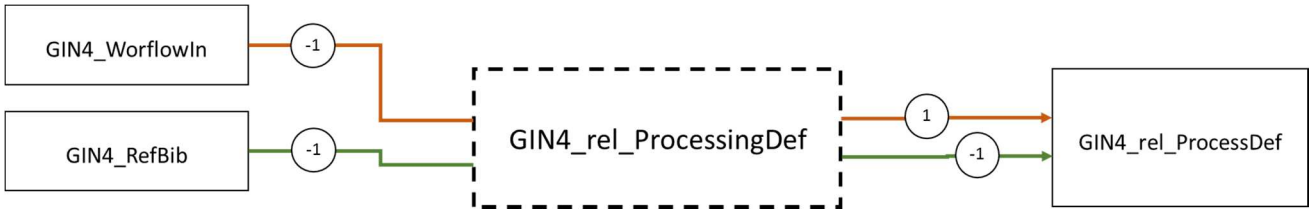
Nom d'affichage : relProcessingDefinition

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_RefBib	-1	**GIN4_rel_ProcessingDef**	-1	GIN4_ProcessDef
GIN4_WorkflowIn	-1	**GIN4_rel_ProcessingDef**	-1	GIN4_ProcessDef



GIN4_rel_ProcessingPar	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relProcessingParameter

Description :

Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
		POM_string	256	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessUR	-1	**GIN4_rel_ProcessingPar**	1	GIN4_ProcessPar
GIN4_RefBib	-1	**GIN4_rel_ProcessingPar**	-1	GIN4_ProcessPar

GIN4_WorkflowIn	-1	**GIN4_rel_ProcessingPar**	-1	GIN4_ProcessPar

GIN4_rel_ProcessingRes	ImanRelation	FAUX	FAUX	FAUX

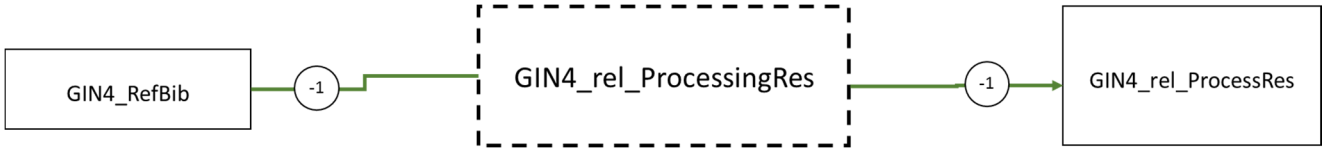
Nom d'affichage : relProcessingResult

Description:

Attributs: Aucuns

Relations:

Primary	Card.	Relation	Card.	Secondary
GIN4_RefBib	-1	**GIN4_rel_ProcessingRes**	-1	GIN4_ProcessRes



GIN4_rel_ProcessingUD	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relProcessingUnitDefinition

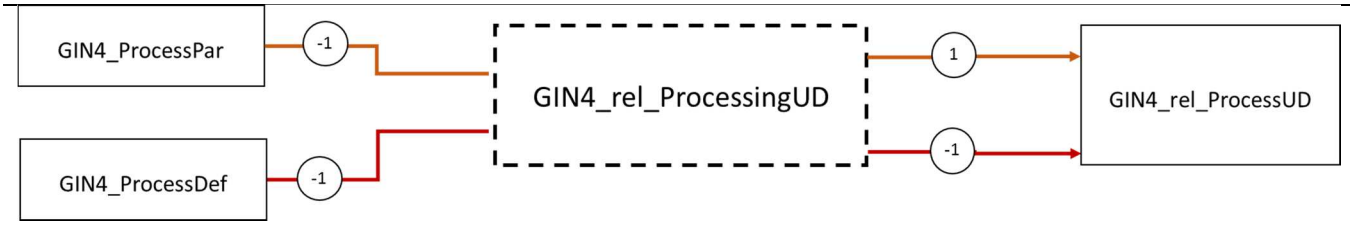
Description :

Attributs :

attributeName	description2	maxStringLength	isArray	followOnExport	isNullsAllowed	isUnique
		POM_string	256	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessPar	-1	**GIN4_rel_ProcessingUD**	1	GIN4_ProcessUD
GIN4_ProcessDef	-1	**GIN4_rel_ProcessingUD**	-1	GIN4_ProcessUD



GIN4_rel_ProcessingUR	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relProcessingUnitResult

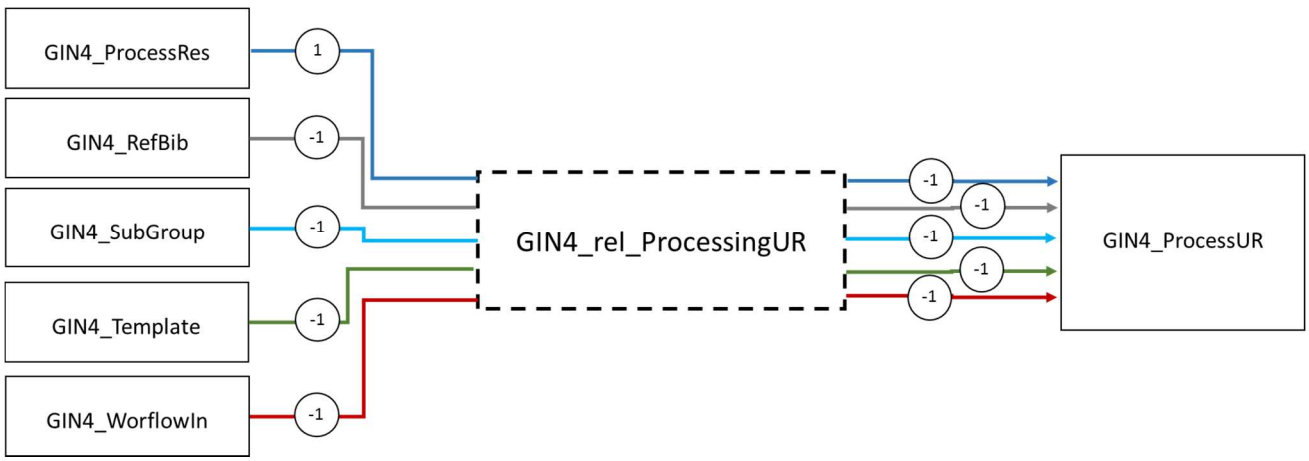
Description :

Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
		POM_string	256	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessRes	1	**GIN4_rel_ProcessingUR**	-1	GIN4_ProcessUR
GIN4_RefBib	-1	**GIN4_rel_ProcessingUR**	-1	GIN4_ProcessUR
GIN4_SubGroup	-1	**GIN4_rel_ProcessingUR**	-1	GIN4_ProcessUR
GIN4_Template	-1	**GIN4_rel_ProcessingUR**	-1	GIN4_ProcessUR
GIN4_WorkflowIn	-1	**GIN4_rel_ProcessingUR**	-1	GIN4_ProcessUR



GIN4_rel_RefBib	ImanRelation	FAUX	FAUX	FAUX

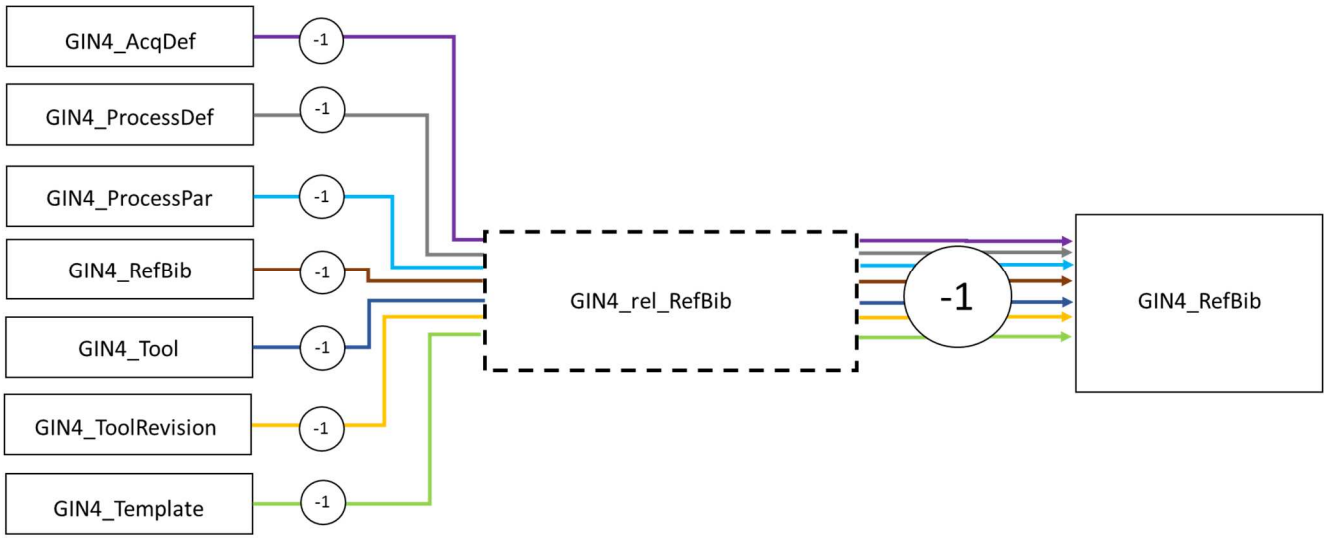
Nom d'affichage : relBibliographicalReference

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_AcqDef	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_ProcessDef	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_ProcessPar	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_RefBib	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_Template	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_Tool	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib
GIN4_ Tool Revision	-1	**GIN4_rel_RefBib**	-1	GIN4_RefBib



GIN4_rel_Scanner	ImanRelation	FAUX	FAUX	FAUX

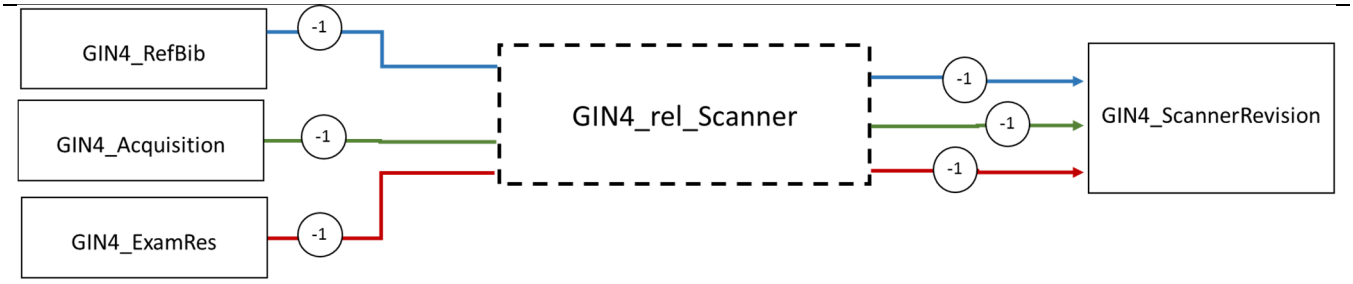
Nom d'affichage : relAcquisitionDevice

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ExamRes	-1	**GIN4_rel_Scanner**	-1	GIN4_ScannerRevision
GIN4_RefBib	-1	**GIN4_rel_Scanner**	-1	GIN4_ScannerRevision
GIN4_Acquisition	-1	**GIN4_rel_Scanner**	-1	GIN4_ScannerRevision



GIN4_rel_StudySubject	ImanRelation	FAUX	FAUX	FAUX

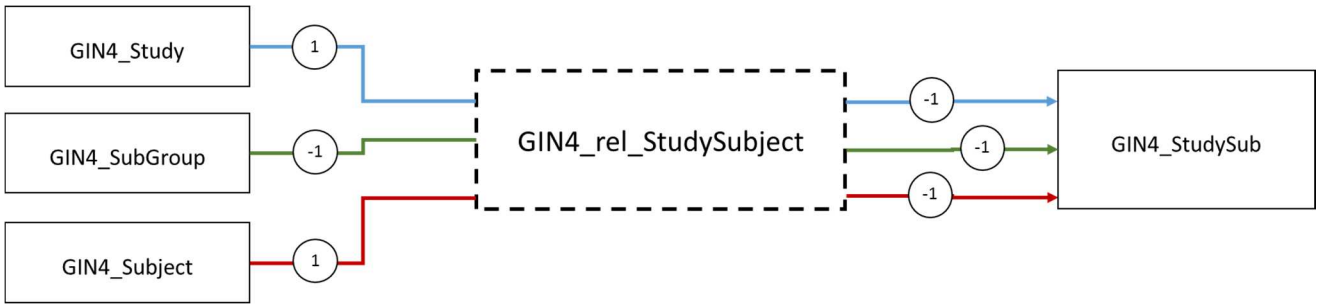
Nom d'affichage : relStudySubject

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_Study	1	**GIN4_rel_StudySubject**	-1	GIN4_StudySub
GIN4_SubGroup	-1	**GIN4_rel_StudySubject**	-1	GIN4_StudySub
GIN4_Subject	1	**GIN4_rel_StudySubject**	-1	GIN4_StudySub



GIN4_rel_Template	ImanRelation	FAUX	FAUX	FAUX

Nom d'affichage : relReferenceData

Description :

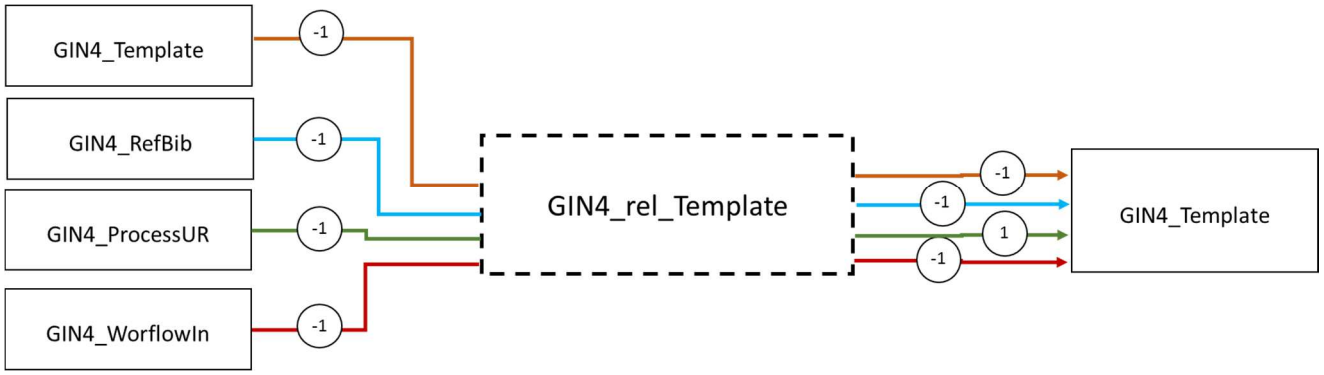
Attributs :

attributeName	**description2**	**maxStringLength**	**isArray**	**followOnExport**	**isNullsAllowed**	**isUnique**
		POM_string	256	FAUX	FAUX	VRAI
gin4_role	**isPublicRead**	**isPublicWrite**	**isCandidateKey**	**isTransient**	**exportAsString**	**noBackpointer**	**initialValue**
	VRAI	VRAI	FAUX	FAUX	FAUX	FAUX	

Relations :

Primary	Card.	Relation	Card.	Secondary

GIN4_RefBib	-1	**GIN4_rel_Template**	-1	GIN4_Template
GIN4_Template	-1	**GIN4_rel_Template**	-1	GIN4_Template
GIN4_WorkflowIn	-1	**GIN4_rel_Template**	-1	GIN4_Template
GIN4_ProcessUR	-1	**GIN4_rel_Template**	-1	GIN4_Template



GIN4_rel_Tool	ImanRelation	FAUX	FAUX	FAUX

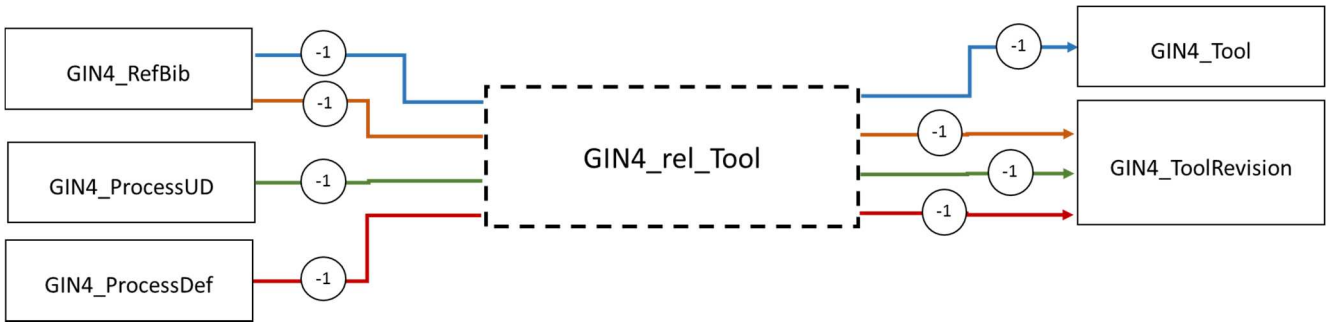
Nom d'affichage : relAcquisitionDevice

Description :

Attributs : Aucun

Relations :

Primary	Card.	Relation	Card.	Secondary
GIN4_ProcessDef	-1	**GIN4_rel_Tool**	-1	GIN4_ToolRevision
GIN4_ProcessUD	-1	**GIN4_rel_Tool**	-1	GIN4_ToolRevision
GIN4_RefBib	-1	**GIN4_rel_Tool**	-1	GIN4_Tool
GIN4_RefBib	-1	**GIN4_rel_Tool**	-1	GIN4_ToolRevision



Annexe E

Classification pour la neuroimagerie

La classification est amenée à évoluer régulièrement, et l'ensemble des classes peut difficilement être donné dans une annexe synthétique. Nous présentons dans ce document les branches de la classification, et la correspondance de certaines classes avec des ontologies existantes. La classification a été développée en collaboration avec Pierre-Yves Hervé, post-doctorant au GIN, de même que la rédaction de ce document.

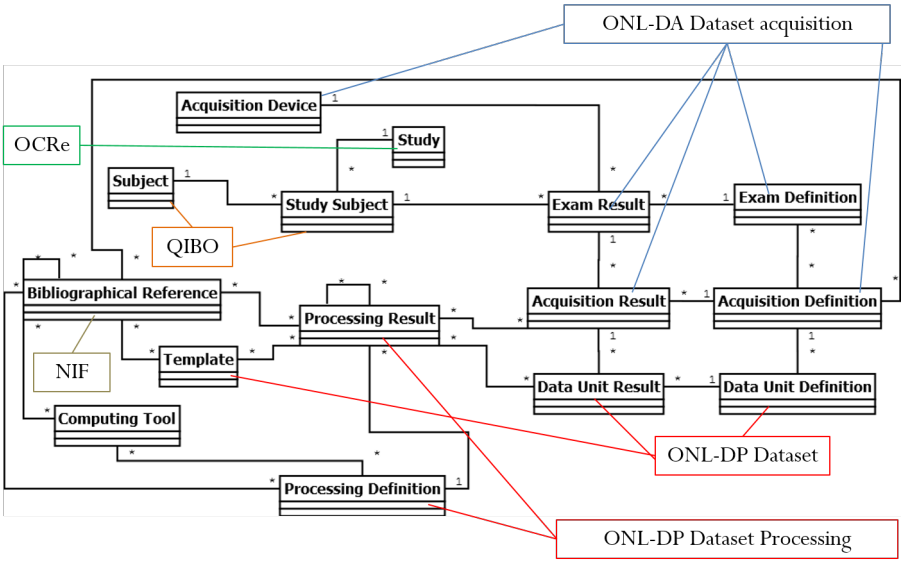


Figure : schéma UML du modèle de données BMI-LM sur lequel sont indiquées les correspondances entre les objets du modèle et les ontologies utilisées pour définir la classification des données pour la neuroimagerie.

Principes généraux

Un arbre de classification et un dictionnaire d'attributs

La classification TeamCenter est avant tout un système de classification hiérarchique. L'exemple type des classifications hiérarchiques est la classification phylogénétique du vivant : une branche pour les végétaux, une pour les animaux, une pour les champignons, et de multiples ramifications. Une nouvelle espèce hérite de caractéristiques de ses ancêtres, et présente de nouvelles caractéristiques.

Dans le cas de TeamCenter, la classification sert à définir des catégories d'items, et à spécifier leurs caractéristiques au moyen d'attributs, eux-mêmes sélectionnés dans un grand catalogue extensible.

On a là aussi une notion d'héritage, car les nouvelles classes héritent des attributs des précédentes. Les nouvelles classes peuvent aussi obtenir de nouveaux attributs, mais ne peuvent jamais en perdre.

Contrairement à la classification phylogénétique du vivant, qui porte sur des espèces strictement séparées, une classification TeamCenter, à visée descriptive, permet l'appartenance à plusieurs classes. Ceci permet d'utiliser un système 'factoriel' : si nous avons un facteur à 2 niveaux (plumes /pas plumes) et un autre facteur à 3 niveaux (2 pattes /3 pattes/4 pattes), il n'y a pas besoin de créer 6 classes pour représenter tous les combinaisons possibles, il suffit de classer l'item de façon appropriée pour chacun des deux facteurs. Ceci nous épargne de dupliquer des branches de la classification.

Dans TeamCenter un groupe (utilisée ici de façon interchangeable avec une branche) permet de rassembler des classes d'un même domaine, à un niveau très général (ex : les animaux, les ustensiles de cuisine), elles ont tellement générales qu'elles ne peuvent contenir d'attribut. Les classes abstraites permettent de spécifier les attributs communs à toutes les classes situées en aval, mais ne peuvent pas contenir d'items. Les oiseaux auront une forme de bec, une couleur de plumage, un poids, une taille, mais on choisira de placer les items plus précisément dans les rapaces, les passereaux, les gallinacées...jusqu'à arriver à des classes concrètes comme par exemple une espèce bien définie telle que la buse variable. Notre buse étant cependant variable, on peut vouloir utiliser des subdivisions de cette classe concrète, ce que TeamCenter permet aussi.

Rapports avec le modèle de données (Business Model TC)

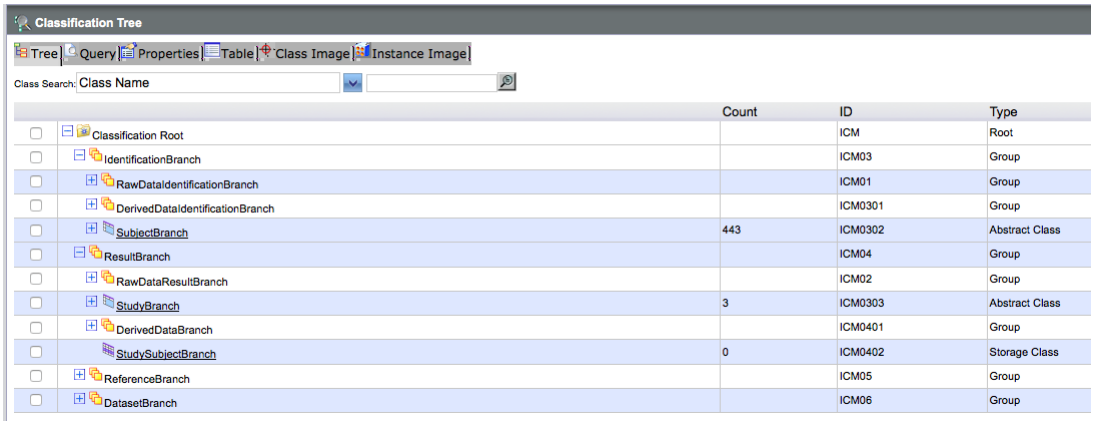
La classification occupe une grande place dans la gestion des données : en règle générale, les attributs de classification servent à stocker toutes les données propres à l'item (exception pour StudySub, qui passe par un formulaire). Donc les classes servent en quelque sorte de tables de données. Ceci permet de faire évoluer le 'MDD au sens large' très rapidement.

A chaque type d'item spécifié dans le Modèle De Données (MDD), nous avons associé un groupe de classification, ou branche. Ceci permet de savoir très rapidement quelle partie de la classification se rapporte à quel type d'objet. Le nom du groupe reprend en effet le nom du

BusinessItem et y ajoute le suffixe Branch.

Les principales branches de la classification BIOMIST Toutes ces branches (une quinzaine) ne partent cependant pas de la racine de la classification.

Un trait d'organisation extrêmement fort du MDD de BIOMIST (voir Thèse de Marianne Allanic), est la séparation entre des items abstraits de définition et des items concrets qui contiennent les entités définies par ces items : d'un coté la définition abstraite du canari, et de l'autre le canari, le vrai. Nous avons donc une Branche de descriptions, parallèlement à une Branche de résultats. On voit ci-dessous ces groupes en jaune.



	Count	ID	Type
Classification Root		ICM	Root
IdentificationBranch		ICM03	Group
RawDataIdentificationBranch		ICM01	Group
DerivedDataIdentificationBranch		ICM0301	Group
SubjectBranch	443	ICM0302	Abstract Class
ResultBranch		ICM04	Group
RawDataResultBranch		ICM02	Group
StudyBranch	3	ICM0303	Abstract Class
DerivedDataBranch		ICM0401	Group
StudySubjectBranch	0	ICM0402	Storage Class
ReferenceBranch		ICM05	Group
DatasetBranch		ICM06	Group

Ceci s'applique aux acquisitions de données brutes (RawData) :

Nous avons donc une branche raw data sous la branche de définitions, avec des classes :

- d'Examens d'Imagerie : définit quoi faire pendant une heure avec un scanner IRM, ou avec une caméra à positons, un imageur ultra-sons, Computed Tomography ou microscopie...
- d'Acquisitions d'imagerie : définit un scan d'une des modalités précédemment évoquées
- d'images médicales : définit une image dérivant d'un scan (ex : module/phase en IRM)

et de même sous la branche résultats, les DerivedData (données dérivées) :

- des résultats d'examen (des logs expérimentaux, des questionnaires de débriefing, des scans, des images)
- des scans : une image T1
- des images numériques : une image de contraste de phase au format DICOM

De même, nous avons une branche Processing pour définition et résultats nous avons des classes pour les définitions des traitements d'images et les résultats des traitements d'image. La définition d'une procédure de filtrage, et le résultat concret de l'application du filtre à une image.

Une troisième branche fondamentale concerne les données de référence, qui sont souvent associées aux définitions. Les références bibliographiques qui ont décrit les méthodes d'acquisition ou de traitement employées sont en fait les sources de nos items de définition. De même, nous utilisons des standards, tels que des atlas d'anatomies, des systèmes de coordonnées tridimensionnels pour la cartographie cérébrale, qui sont exprimés sous la forme d'images.

Une quatrième branche fondamentale est celle des datasets, c'est notamment le cas pour les fichiers images, qui contiennent les résultats de traitements numériques. Ceci permet une

grande précision dans l'annotation des résultats, notamment pour des procédures qui génèrent plusieurs fichiers de nature différente.

Le MDD spécifie que tout item de résultat est lié à une définition. Un item de définition est par conséquent lié à de nombreux résultats. Cependant, une limite de la classification TeamCenter est qu'elle ne permet pas d'établir des liens pourtant systématiques (donc prévisibles de façon automatique) entre les classes de définitions et les classes de résultats. Pour cela, il faut utiliser une ontologie, qui est un système de classification beaucoup plus flexible, capable de gérer des liens à la fois verticaux et horizontaux.

Rapports avec les ontologies métier

Pour échanger des données entre différents sites et différents systèmes de base de données, il est préférable de parler un langage commun, ou au moins d'utiliser un vocabulaire commun. C'est pourquoi nous avons choisi d'utiliser des classes issues des ontologies déjà publiées et mise à disposition de la communauté via l'annuaire de ressources http://bioportal.bioontology.org/.

Une ontologie peut être explorée au moyen du logiciel protégé [1].

Nous venons de préciser qu'une classification TeamCenter ne peut représenter une ontologie, cependant, nous pouvons récupérer tout ou partie des classes définies, de façon à utiliser le même lexique. Cela fournit aussi un premier moyen de brancher une ontologie, forcément plus riche en relations, et contenant aussi des définitions textuelles des classes, sur notre arbre de classification. Nous pouvons intégrer plusieurs ontologies, en fonction du domaine concerné : psychologie, imagerie, clinique, conduite d'études scientifiques, bibliographie, etc.

Nous avons utilisé cette approche pour les Acquisitions et les traitements d'image IRM, qui s'appuie largement sur la classification OntoVIP, qui regroupe OntoNeuroLog Dataset Acquisition et OntoNeuroLog Dataset Processing (ONL-DA/DP). Ces ontologies proviennent de Rennes (B. Gibaud) et sont utilisées actuellement dans le cadre du projet SHANOIR. Ces classes ont parfois du être complétées pour répondre aux besoins du projet.

Branchements OntoVIP/QIBO

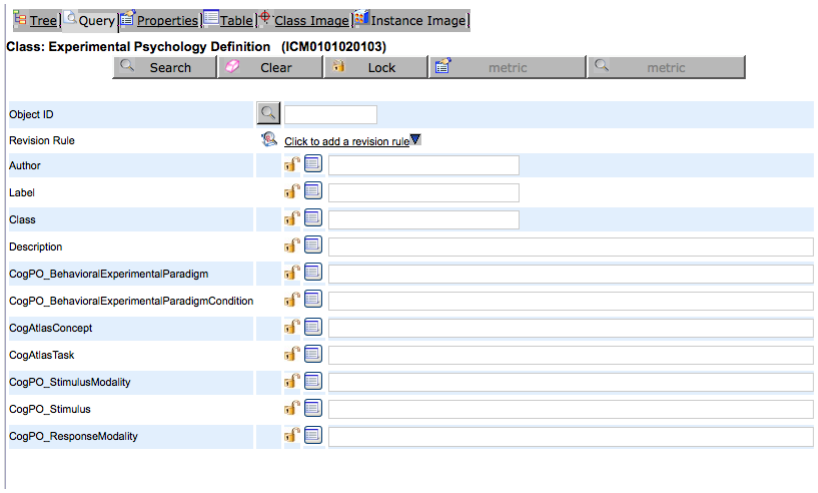
Il s'agit ici d'intégration de classes.

1. Protégé est un éditeur d'ontologies développé par l'Université de Stanford.

BIOMIST (classe parente)	**OntoVIP (classe parente)**
descriptionBranch»ProcessingUnitDefBranch ICM030109	dataset-processing
datasetBranch»imageProcessingResBranch ICM040103 resultsBranch»MR-performed-acquisition ICM010101030301	dataset
descriptionBranch»MR-examination-protocol1 ICM01010103	planned-acquisition-MR-protocol
resultsBranch»MR-performed-protocol ICM0101010303	performed-acquisition-MR-protocol
descriptionBranch»MR-sequence ICM010101030101	MR-sequence
	QIBO
SubjectBranch ICM0302	Organism

Annotation via les attributs de classe

Ce cas de figure concerne les attributs des classes de définition pour la psychologie. On l'utilise pour annoter un test psychologique avec les fonctions qu'il évalue, via des entités du Cognitive Atlas et CognitiveParadigmOntology.



Annexe F

Étude du Graphe Dynamique Multidimensionnel GMD-4-6

Nous donnons cet exemple simple de GDM pour illustrer les mécanismes de l'analyse dynamique multidimensionnelle et montrer différentes façons de présenter des données de graphes : format tabulaire, représentation node-link et format JGEX.

Données du GMD-4-6

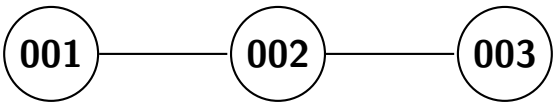
Le jeu de données est composé de deux graphes. Le premier graphe est composé de quatre nœuds et de six arêtes dont l'existence est fonction de deux dimensions, *sujet* et *acquisition time* ; le graphe présente au total six états. Le deuxième graphe est composé de trois nœuds et de deux arêtes ; il s'agit du graphe des conditions de la dimension *sujet*.

Données sous forme tabulaire

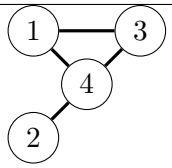
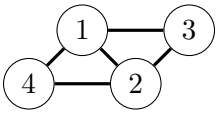
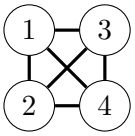
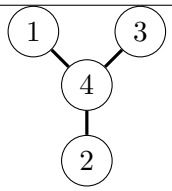
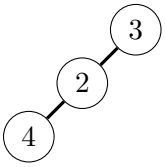
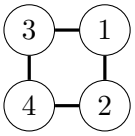
2 dimensions	sujet	001 : age=25, handedness=left
		002 : age=23, handedness=right
		003 : age=34, handedness=right
	acquisition time	t1
		t2

Représentation node-link des données

Graphe 1 - représentation node-link des trois conditions de la dimension "sujets" [TODO : rajouter une direction aux flèches et plusieurs ordres possibles / Ce graphe-là est aussi dynamique, sur les arêtes.]



Graphe 2 - représentation node-link des six états du graphe

	subject 001	subject 002	subject 003
time t1			
time t2			

Données au format JGEX

```
{"meta" :
        {"creator":"mallanic",
         "keywords":null,
         "lastmodifieddate":"2015-05-28",
         "description":null},
 "attributes" :
        [{"scope":"edge", "impact":"Regions",
          "id":"aSubjects", "label":"Subjects",
          "type":"string", "structure":"simple"},
         {"scope":"edge", "impact":"Regions",
          "id":"aTime", "label":"Time",
          "type":"string", "structure":"simple"},
         {"scope":"node", "impact":"Subjects",
          "id":"aAge", "label":"Age",
          "type":"integer", "structure":"simple"},
         {"scope":"edge", "impact":"Regions",
          "id":"aweight", "label":"WeightDyn",
          "type":"double", "structure":"simple"},
         {"scope":"node", "impact":"Subjects",
          "id":"aHandedness", "label":"Handedness",
          "type":"string", "structure":"simple"}],
 "graphs":
   [{"id":"Regions", "label":"Regions",
         "mode":"undirected", "type":"exemple",
         "nodes":
```

```
        [{"id":"r2","label":"2","type":"region","attvalues":[]},
         {"id":"r3","label":"3","type":"region","attvalues":[]},
         {"id":"r4","label":"4","type":"region","attvalues":[]},
         {"id":"r1","label":"1","type":"region","attvalues":[]}
        ],
        "edges":
        [{"weight":1.0,"id":"e02","label":"e02","source":"r1",
          "target":"r3","type":"undirected","attvalues":
                [{"attr":"aweight","value":{"ranges":
                      [{"condition":"s003","value":{"ranges":
                            [{"condition":"t1","value":1.0},
                             {"condition":"t2","value":1.0}
                            ], "dim":"aTime"}}
                      ], "dim":"aSubjects"}, "currentValue":
                         null}]},
         {"weight":1.0,"id":"e01","label":"e01","source":"r1",
          "target":"r2","type":"undirected","attvalues":
             [{"attr":"aweight","value":{"ranges":
               [{"condition":"s002","value":{"ranges":
                   [{"condition":"t1","value":1.0}
                   ],"dim":"aTime"}},
                {"condition":"s003","value":{"ranges":
                      [{"condition":"t1","value":1.0},
                       {"condition":"t2","value":1.0}
                      ],"dim":"aTime"}},
                {"condition":"s001","value":{"ranges":
                   [{"condition":"t1","value":1.0}
                   ],"dim":"aTime"}},
              ], "dim":"aSubjects"},"currentValue":null}]},
         {"weight":1.0,"id":"e04","label":"e04","source":"r2",
          "target":"r3","type":"undirected","attvalues":
             [{"attr":"aweight","value":{"ranges":
               [{"condition":"s002","value":{"ranges":
                   [{"condition":"t1","value":1.0},
                    {"condition":"t2","value":1.0}
                   ],"dim":"aTime"}},
                {"condition":"s003","value":{"ranges":
                   [{"condition":"t1","value":1.0}
                   ],"dim":"aTime"}}
              ],"dim":"aSubjects"},"currentValue":null}]},
         {"weight":1.0,"id":"e03","label":"e03","source":"r1",
```

```
"target":"r4","type":"undirected","attvalues":
[{"attr":"aweight","value":{"ranges":
[{"condition":"s001","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}},
{"condition":"s002","value":{"ranges":
[{"condition":"t1","value":1.0}
],"dim":"aTime"}},
{"condition":"s003","value":{"ranges":
[{"condition":"t1","value":1.0}
],"dim":"aTime"}}
],"dim":"aSubjects"},"currentValue":null}]},
{"weight":1.0,"id":"e06","label":"e06","source":"r3",
"target":"r4","type":"undirected","attvalues":
[{"attr":"aweight","value":{"ranges":
[{"condition":"s001","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}},
{"condition":"s003","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}}
],"dim":"aSubjects"},"currentValue":null}]},
{"weight":1.0,"id":"e05","label":"e05","source":"r2",
"target":"r4","type":"undirected","attvalues":
[{"attr":"aweight","value":{"ranges":
[{"condition":"s001","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}},
{"condition":"s002","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}},
{"condition":"s003","value":{"ranges":
[{"condition":"t1","value":1.0},
{"condition":"t2","value":1.0}
],"dim":"aTime"}}
],"dim":"aSubjects"},"currentValue":null}]}
```

```
        ]},
    {"id":"Subjects","label":"Subjects",
     "mode":"undirected","type":"condition",
     "nodes":
     [{"id":"s003","label":"subject 003","type":"region",
       "attvalues":
         [{"attr":"aAge","value":34,"currentValue":null},
         {"attr":"aHandedness","value":"right","currentValue":
            null}]},
          {"id":"s002","label":"subject 002","type":"region",
           "attvalues":
             [{"attr":"aAge","value":23,"currentValue":null},
              {"attr":"aHandedness","value":"right","currentValue
                 ":null}]},
          {"id":"s001","label":"subject 001","type":"region",
           "attvalues":
             [{"attr":"aAge","value":25,"currentValue":null},
             {"attr":"aHandedness","value":"left","currentValue":
                null}]}
         ],
     "edges":
     [{"weight":1.0,"id":"es01","label":"es01","source":"s001",
     "target":"s002","type":"undirected","attvalues":[]},
          {"weight":1.0,"id":"es02","label":"es02","source":"s002
             ",
           "target":"s003","type":"undirected","attvalues":[]}
         ]}
   ]
}
```

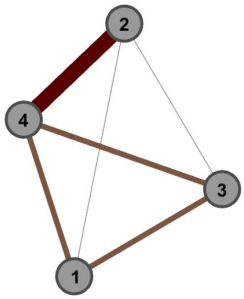

Etude du GMD-4-6

Graphes de synthèse

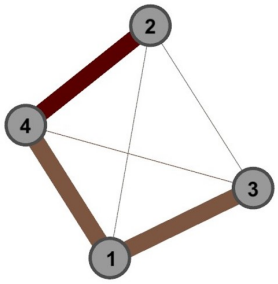
Le tableau ci-dessous donne la valeur du poids de synthèse des arêtes sur l'ensemble des dimensions du GMD (*Global*), sur la dimension *sujets* pour l'acquisition *t1* (*Sujets-t1*), et sur la dimension *acquisitions* pour le sujet *001* (*Acquisitions-sujet001*).

	1-2	1-3	1-4	2-3	2-4	3-4
Global	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	1	$\frac{2}{3}$
Sujets-t1	$\frac{2}{3}$	1	1	$\frac{2}{3}$	1	$\frac{2}{3}$
Acquisitions-sujet001	0	$\frac{1}{2}$	1	0	1	1

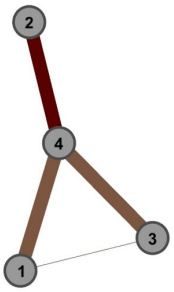
Les trois graphes ci-dessous présentent de façon visuelle le poids des arêtes de synthèse (épaisseur et couleur des arêtes) :



a- Global



b- Sujets-t1

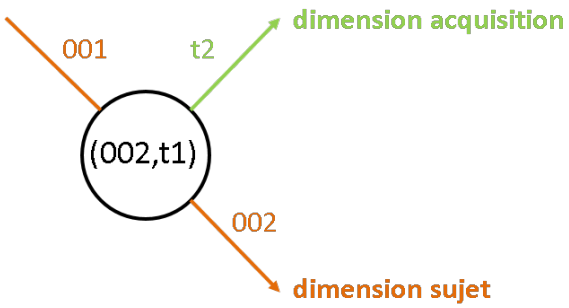


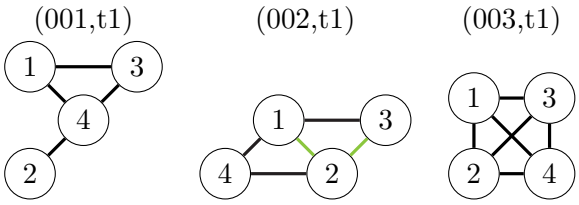
c- Acquisitions-sujet001

Comparaisons *en contexte*

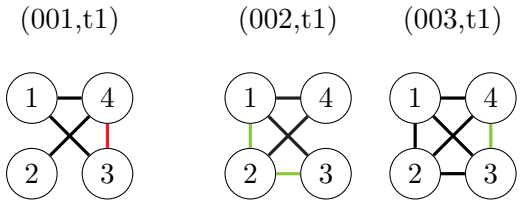
Naissance et mort

Dans un premier temps, les événements de naissance et de mort du GMD sont analysés *en contexte*. L'état $(0002, t1)$ est pris comme exemple ; la situation de l'état en contexte est illustrée dans la figure ci-dessous :

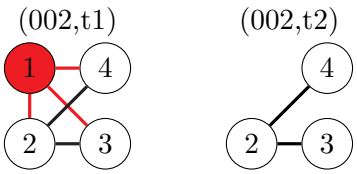




a- selon la dimension *sujet*



b- selon la dimension *sujet*, layout fixe



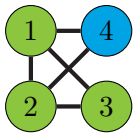
c- selon la dimension *acquisition*

d- selon le contexte e- synthèse du contexte

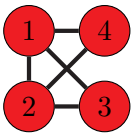
Évolution d'un attribut

L'évolution du degré des nœuds est étudié *en contexte*. Les valeurs des degrés des nœuds pour chaque état sont données dans le tableau ci-dessous :

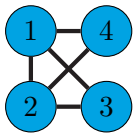
nœud / état	**(001,t1)**	**(001, t2)**	**(002,t1)**	**(002, t2)**	**(003,t1)**	**(003, t2)**
1	2	1	3	0	3	2
2	1	1	3	2	3	2
3	2	1	2	1	3	2
4	3	3	2	1	3	2



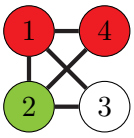
a- selon la dimension *sujet*



b- selon la dimension *acquisition*



c- selon le contexte



d- synthèse du contexte

La figure ci-dessus présente les résultats de la comparaison pour l'état (002,t1) :

a) Selon la dimension *sujet* : les nœuds 1, 2 et 3 gagnent des connexions dans le contexte de la dimension sujet, tandis que le nœud 4

b) Selon la dimension *acquisition* : les quatre nœuds perdent des connexions entre les états (002,t1) et (002,t2).

c) Selon le contexte (dimensions *sujet* et *acquisition*) : les quatre nœuds présentent une inflexion dans le contexte, ils gagnent et perdent chacun des connexions.

d) Synthèse du contexte : la somme signée des changements est nulle sur le nœud 3 ; tandis que les nœuds 1 et 4 perdent globalement des connexions sur le contexte, le nœud 2 en gagne.

Annexe G

Primer JGEX

JGEX 1.0 draft Primer

Marianne Allanic, Nicolas Boulic, Philippe Boutinaud

January, 2015

Abstract JGEX Primer is a non-normative document and aims at providing a quick understanding of the Json Graph Exchange (JGEX) format. Basic features of the JGEX format are explained through simple examples. The specification is in JSON format and can be found at biomist.fr.

Introduction

Json Graph Exchange (JGEX) format has been designed to support the exchange of dynamic multi-dimensionnal data between programs and applications.

JGEX format is an extension of JSON format, and its schema can be found at biomist.fr.

This primer explains the main features of the JGEX format and should be used as a handbook to understand the specification of JGEX. The primer itself is not a specification, but provides examples and other material. The intended audience of the document is application developers and every user that need to use JGEX to exchange data. The document assumes that the reader is familiar with the JSON syntax.

First, the basic features of JGEX are described, such as graphs and their elements, definition of attributes and metadata.

Second, the dynamic features of JGEX are clarified.

Third, some common conventions among JGEX users are presented.

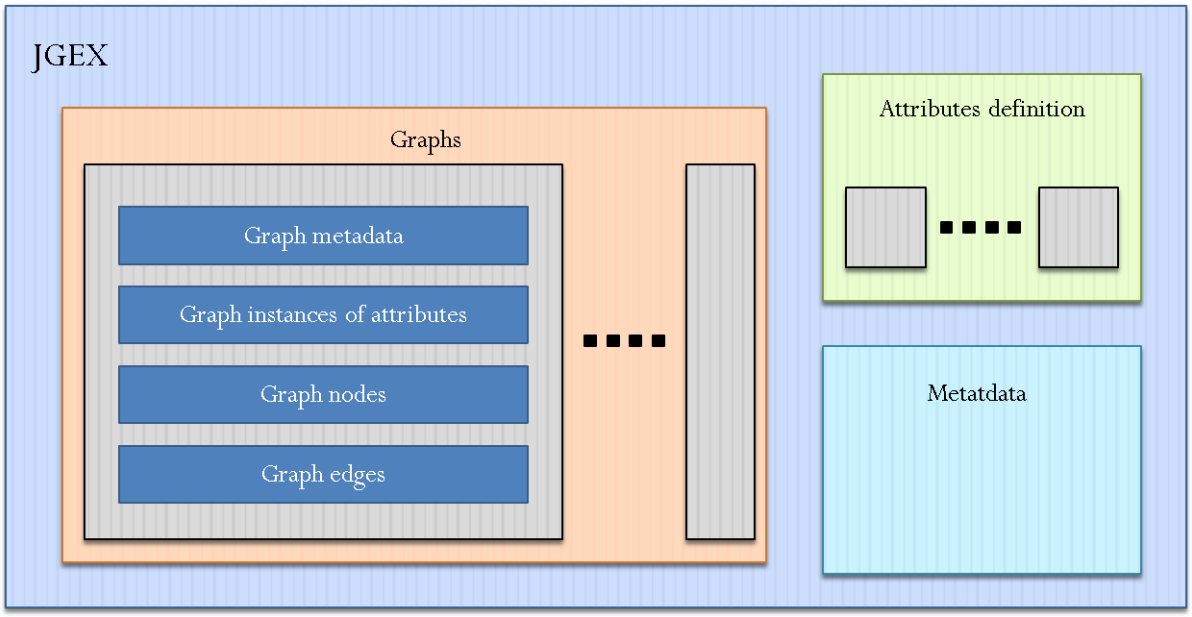


FIGURE G.1 – Overall structure of a JGEX file

```
{
    "lastmodifieddate": "2015-01-01",
    "creator": "BIOMIST project",
    "keywords": "graph, exchange",
    "description": "file containing two graphs"
}
```

FIGURE G.2 – JGEX : metadata of a file

Basic features

Structure of a JGEX file

JGEX is composed of some metadata, a list of graphs and a list of definitions of attributes. The overall structure of a JGEX file is represented in figure G.1.

JGEX file metadata

Metadata in JGEX are overall information about the file. *Meta* object can only be located on the root of the JSON file structure.

Meta object has four properties :

- *Creator* : (required) string value defining author of the file.
- *Lastmodifieddate* : (required) string value indicating last modification date.
- *Description* : string describing the file.
- *Keywords* : string of words separated by commas.

```
{
  "id": "aheight",
  "impact": "all",
  "scope": "node",
  "label": "height",
  "type": "double",
  "unit": "m",
  "structure": "simple",
  "script": "",
  "condGraph": "",
  "default": 0.0
}
```

FIGURE G.3 – JGEX : definition of an attribute

Definition of attributes

Attributes array contains the definition of every attribute applied on a graph element. It can only be located on the root of the JSON file structure. *Attribute* object is the definition of an attribute that can be applied on a graph element. It has ten properties :

- *Id* : (required) unique string among every attributes Ids.
- *Label* : (required) string defining display name of the attribute.
- *Scope* : (required) string defining the extend of the attribute on graph elements. Allowed values : {global, graph, node, edge}.
- *Impact* : (required) string defining the reach of application of the attribute : one specific graph (through its Id) or every graph ("all").
- *Type* : (required) string indicating value type of the attribute.
- *Structure* : (required) string indicating value structure of the attribute. Allowed values : {simple, array, map}.
- *Unit* : string indicating the unit of the attribute values.
- *CondGraph* : string id of the graph that contains condition nodes of the dimension (this assumes that the attribute is a dimension of a graph, see section G for explanations on dynamics features).
- *Default* : default value to assign to instances of the attribute.
- *Script* : string with javascript code.

Content of a graph

Graph properties

Graphs array contains the graphs stored in the file. It can only be located on the root of the JSON file structure. *Graph* object has seven properties :

- *Id* : (required) unique string among every graphs Ids.
- *Label* : (required) string defining display name of the graph.

```
"graphs":
        [{"id":"graphId",
          "label":"graphLabel",
          "mode":"undirected",
          "type":"graphType",
          "nodes": [],
          "edges": [],
          "attvalues":[]
        }]
```

FIGURE G.4 – JGEX : graph

```
"nodes":
        [{"id":"nodeId",
          "label":"nodeLabel",
          "type":"nodeType",
          "attvalues":[]
        }]
```

FIGURE G.5 – JGEX : node

- *Mode* : (required) string defining the mode of the edges of the graph directed, undirected.
- *Type* : (required) string indicating value type of the graph.
- *Nodes* : (required) array of nodes of the graph.
- *Edges* : array of edges of the graph.
- *Attvalues* : array of instances of attributes of the graph.

Node objects contained in *nodes* array have four properties :

- *Id* : (required) unique string among every nodes Ids in the graph.
- *Label* : (required) string defining display name of the node.
- *Type* : (required) string indicating value type of the node.
- *Attvalues* : array of instances of attributes of the node.

Edge objects contained in *edges* array have seven properties :

- *Id* : (required) unique string among every nodes Ids in the graph.
- *Label* : (required) string defining display name of the edge.
- *Type* : (required) string indicating value type of the edge.
- *Source* : (required) string id of the source node of the edge.
- *Target* : (required) string id of the target node of the edge.
  When graph mode is set to *undirected*, source and target are interpreted indiscriminately.
- *Weight* : (required) double value of the weight of an edge (cannot be dynamic). The value is 1.0 by default.
- *Attvalues* : array of instances of attributes of the edge.

```
"edges":
        [{"id":"edgeId",
          "label":"edgeLabel",
          "source":"sourceNodeId",
          "target":"targetNodeId",
          "weight":1.0,
          "type":"edgeType",
          "attvalues":[]
        }]
```

FIGURE G.6 – JGEX : edge

```
"attvalues":
        [{"attr":"aweight",
          "value":0.25,
          "currentValue":null
        }]
```

FIGURE G.7 – JGEX : instance of attribute

Elements data

Graph, *node* and *edge* can be associated with instances of attributes through *attvalues* array. *attvalue* object has three properties :

- *Attr* : (required)string id of the attribute definition.
- *Value* : (required) value of the instance (type to specify).
- *CurrentValue* : temporary value of the instance (type to define). This property is used when *value* is dynamic.

Authorized data types The data types allowed in JGEX are standards : boolean, integer, float, double, object, string, date.

Dynamics features

One of the most interesting feature of JGEX is the possibility to make nodes and edges attributes vary along many *dimensions*, whereas other graph format only allow static graphs or at most temporal dynamics.

In JGEX, any defined attribute can be a dimension, and any attribute can vary along a dimension, which allows infinite possibilities. A dimension is composed of *conditions*. The *state* of a graph is the intersection of the dimensions, defined by a set of conditions. A *configuration* of the graph is the extraction of the values associated to each element at a given state.

```
"attvalues":
        [{"attr":"aweight",
          "value":
            {"dim":"conditionId",
             "ranges":
                   [{"condition":"conditionId1","value":0.64},
                    {"condition":"conditionId2","value":0.27}]
            },
          "currentValue":0.27
        }]
```

FIGURE G.8 – JGEX : configured value in instance of attribute

Dynamics of elements data

Configured Value

To indicate that an instance of attribute of an element (*graph*, *node* or *edge*) is dynamic, a *configured value* is put in *value* field of *attvalue*. This *configured value* has two properties :

- *Dim* : (required) string Id of the attribute chosen for dimension.
- *Ranges* : (required) array of ranges, which represents the impact of a condition for a given dimension.
  - *Condition* : (required) value of the condition (type to specify). Two ranges of an instance cannot have the same *condition*.
  - *Value* : (required) value of the instance (type to specify).

The property *current value* of an instance of attribute may be used to store the last dynamic value read.

Interval and Boundary

In JGEX, dimensions can be continuous or discrete, and the conditions can be of many types : integer, double, string or an interval. The type of a dimension is given at attribute definition level.

An interval is defined by two properties :

- *Start* : (required) start boundary of the interval.
- *End* : (required) end boundary of the interval.

Interval object has two properties :

- *Value* : (required) value of the instance (type to specify).
- *Type* : (required) string characterizing the boundary. Allowed values : {opened,closed,infinite}.

Conditions graph

It is possible in JGEX to associate some information to the conditions of a dimension. During the definition of an attribute, the field *condGraph* the id of the graph where the information is stored. This graph is called *condition graph* and its nodes represent the conditions.

```
"attvalues":
        [{"attr":"aweight",
          "value":
            {"dim":"conditionId",
             "ranges":
                   [{"condition":
                       {"start":
                         {"value":"intervalStartValue",
                          "type":"infinite"}},
                        "end":
                         {"value":"intervalEndValue"
                          "type":"closed"}},
                       },
                     "value":0.64}
                   ]
                },
          "currentValue":0.27
        }]
```

FIGURE G.9 – JGEX : configured value in instance of attribute, with a condition as an interval

```
"attributes":
        [{"id":"aweight",
          "label":"WeightDyn",
          "type":"double",
          "structure":"simple",
          "scope":"edge",
          "impact":"impactedGraphId"
        }]
```

FIGURE G.10 – JGEX : dynamic weight attribute

Common conventions

Dynamic weight

Weight property of *edges* cannot be dynamic. Therefore, to set a dynamic weight on edges, a specific edge-scoped attribute must be defined. By convention, the dynamic weight attribute is defined as in figure G.10.

Node position

It is of interest to keep nodes position. A common way to store coordinates is with a cartesian triplet, which we use as well in JGEX. A position can be dynamic, as any value stored in an instance of attribute. By convention, the 3D (respectively 2D, when "z" is forced to 0) position attribute is defined as in figure G.11. An example of instance of attribute is given in figure G.12.

```
"attributes":
        [{"id":"aposition3D",
          "label":"position3D",
          "type":"double",
          "structure":"array",
          "scope":"node",
          "impact":"impactedGraphId",
        }]
```

FIGURE G.11 – JGEX : 3D position attribute

```
"attvalues":
        [{"attr":"aposition3D",
          "value":
                {"x":40.806071,
                 "y":14.509275,
                 "z":4.0910623},
          "currentValue":null
        }]
```

FIGURE G.12 – JGEX : 3D position instance of attribute (static)

Acknowledgement

JGEX was developed within the ANR (Agence Nationale de la Recherche) funded project *BIOMIST* (no ANR-13-CORD-0007) for thematic axis n°2 of the *Contint 2013 Call for Proposal* : from content to knowledge and big data.

Annexe H

Schéma JSON de définition du format JGEX

JSON Graph EXchange

Schéma de définition

Marianne Allanic, Nicolas Boulic

2014

```
1
2  {
3    "description": "A representation of a swodata exchange",
4    "type": "object",
5    "$schema": "http://json-schema.org/draft-04/schema#",
6    "definitions": {
7          "interval" : {
8        "description": "undefined list of graph type",
9        "type": "object",
10       "properties": {
11         "start": {
12           "$ref": "#/definitions/boundary"
13         },
14         "end": {
15           "$ref": "#/definitions/boundary"
16         }
17       }
18     },
```

```
        "condition" : {
      "description": "undefined list of graph type",
      "type": "object",
      "oneOf": [
        {
          "$ref": "#/definitions/interval"
        },
        {
          "type": "string"
        }
      ]
    },
    "graphModes": {
      "description": "list of graph mode",
      "enum": [
        "directed",
        "undirected"
      ]
    },
    "graphTypes": {
      "description": "undefined list of graph type",
      "type": "string"
    },
    "attributTypes": {
      "description": "list of type for a attribute",
      "enum": [
        "boolean",
        "integer",
        "float",
        "object",
        "string",
        "date"
      ]
    },
    "allowedDefaultObjects": {
      "description": "list of allowed object",
      "type": [
        "array",
        "boolean",
        "integer",
        "null",
```

```
        "number",
        "object",
        "string"
      ]
    },
    "allowedObjects": {
      "description": "list of allowed object",
      "oneOf": [
        {
          "$ref": "#/definitions/allowedDefaultObjects"
        },
        {
          "$ref": "#/definitions/configuredValue"
        }
      ]
    },
    "meta": {
      "type": "object",
      "properties": {
        "creator": {
          "type": "string"
        },
        "description": {
          "type": "string"
        },
        "keywords": {
          "type": "string"
        },
        "lastmodifieddate": {
          "type": "string"
        }
      },
      "required": [
        "creator",
        "lastmodifieddate"
      ]
    },
    "boundary": {
      "description": "A boundary of range",
      "type": "object",
      "properties": {
```

```
        "value": {
          "type": [
            "boolean",
            "integer",
            "null",
            "number",
            "object",
            "string"
          ]
        },
        "type": {
          "enum": [
            "opened",
            "closed",
            "infinite"
          ]
        }
      },
      "required": [
        "type"
      ]
    },
    "attribute": {
      "description": "A attribute of object data",
      "type": "object",
      "properties": {
        "scope": {
          "enum": [
            "all",
            "graph",
            "node",
            "edge"
          ]
        },
        "impact": {
          "type": "string",
          "format": "uri"
        },
        "id": {
          "type": "string"
        },
```

```
        "label": {
          "type": "string"
        },
        "unit": {
          "type": "string",
          "default": "null"
        },
        "structure": {
          "enum": [
            "simple",
            "array",
            "map"
          ],
          "default": "simple"
        },
        "script": {
          "type": "string"
        },
        "type": {
          "$ref": "#/definitions/attributTypes"
        },
        "default": {
          "$ref": "#/definitions/allowedDefaultObjects"
        },
        "order": {
          "type": "array"
        }
      },
      "required": [
        "scope",
        "id",
        "label",
        "type"
      ]
    },
    "range": {
      "description": "A range of configured value",
      "type": "object",
      "properties": {
        "condition": {
          "$ref": "#/definitions/condition"
```

```
        },
        "value": {
          "$ref": "#/definitions/allowedObjects"
        }
      },
      "required": [
        "type",
        "value"
      ]
    },
    "configuredValue": {
      "description": "A configured value of attribute",
      "type": "object",
      "properties": {
        "dim": {
          "type": "string"
        },
        "ranges": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/range"
          }
        },
        "currentValue": {
          "type": [
            "array",
            "boolean",
            "integer",
            "null",
            "number",
            "string"
          ]
        }
      },
      "required": [
        "dim",
        "ranges"
      ]
    },
    "attvalue": {
      "description": "A value of attribute",
```

```
      "type": "object",
      "properties": {
        "attr": {
          "type": "string"
        },
        "value": {
          "$ref": "#/definitions/allowedObjects"
        },
        "currentValue": {
          "type": [
            "array",
            "boolean",
            "integer",
            "null",
            "number",
            "string"
          ]
        }
      },
      "required": [
        "attr",
        "value"
      ]
    },
    "node": {
      "type": "object",
      "properties": {
        "attvalues": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/attvalue"
          }
        },
        "id": {
          "type": "string",
          "format": "uri"
        },
        "label": {
          "type": "string"
        },
        "type": {
```

```
          "type": "string"
        }
      },
      "required": [
        "id",
        "label",
        "type"
      ]
    },
    "edge": {
      "type": "object",
      "properties": {
        "attvalues": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/attvalue"
          }
        },
        "id": {
          "type": "string",
          "format": "uri"
        },
        "label": {
          "type": "string"
        },
        "source": {
          "type": "string",
          "format": "uri"
        },
        "target": {
          "type": "string",
          "format": "uri"
        },
        "weight": {
          "type": "number"
        },
        "type": {
          "type": "string"
        }
      },
      "required": [
```

```
        "id",
        "label",
        "source",
        "target",
        "type"
      ]
    },
    "graph": {
      "type": "object",
      "properties": {
        "id": {
          "type": "string",
          "format": "uri"
        },
        "label": {
          "type": "string"
        },
        "mode": {
          "$ref": "#/definitions/graphModes"
        },
        "type": {
          "$ref": "#/definitions/graphTypes"
        },
        "attvalues": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/attvalue"
          }
        },
        "edges": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/edge"
          }
        },
        "nodes": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/node"
          }
        }
```

```
      },
      "required": [
        "id",
        "mode"
      ]
    }
  },
  "properties": {
    "meta": {
      "$ref": "#/definitions/meta"
    },
    "attributes": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/attribute"
      }
    },
    "graphs": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/graph"
      }
    }
  },
  "required": [
    "attributes",
    "graphs"
  ]
}
```

Annexe I

SwoViewer : présentation des fonctionnalités de l'interface web

SwoViewer est un logiciel d'exploration de graphes développé par CADESIS et qui est utilisé dans le projet BIOMIST. Proposé sous forme de client web, l'interface de SwoViewer est implémenté en javascript.

Import-export de données

Pour le moment les formats d'import autorisés sont JGEX et TULIP.

L'export de GMD se fait uniquement en JGEX, qui est le seul format à proposer une représentation de données multidimensionnelles et dynamiques.

Navigation dans les données

Options d'affichage

Les options classiques d'affichage des données sont disponibles :

- Positionnement de la caméra : zoom, translation, rotation, plan précis.
- Changements d'échelle.
- Affichage ou non des arêtes, qui peuvent être droites ou courbes.

Filtres

Filtres classiques Ils peuvent être appliqués à la fois sur les nœuds et sur les arêtes du graphe :

- Visibilité : définit les critères d'affichage des éléments du graphe.
- Taille : définit la taille des éléments (diamètres pour une forme standard de nœuds, épaisseur pour les arêtes).
- Couleur : définit la couleur des éléments.

Filtres de mapping Les filtres de mapping permettre de faire correspondre directement un attribut avec une des variables visuelles proposées :

- Visibilité : prend en entrée un attribut boolean seulement.
- Taille : prend en entrée un attribut entier ou réel positif.
- Couleur : prend en entrée un attribut hexadécimal ou RBG.
- Position : ne s'applique que sur les nœuds, permet de sélectionner une position pré-calculée pour les nœuds d'un graphe. Prend en entrée un vecteur de position à trois coordonnées ou trois attributs qui vont correspondre aux trois coordonnées dans l'espace.
- Poids : ne s'applique que sur les arêtes, permet de sélectionner l'attribut qui représente le poids des arêtes d'un graphe.
- Style : prend en entrée un graphe de style (voir paragraphe "Styles d'affichage").

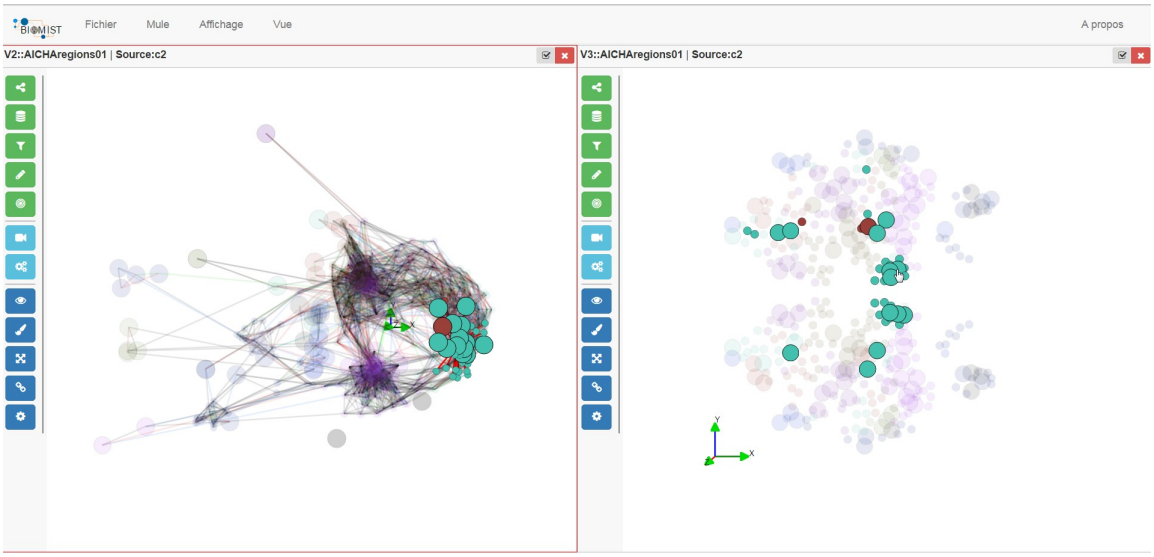
Navigation dynamique L'interface permet une navigation dynamique facilitée par un menu de choix des dimensions et des conditions à afficher dans chaque vue, qui est géré comme l'application d'un filtre. Une option de pré-calul permet de rendre fluide le passage d'une configuration de graphe à une autre.

Multi-vues

L'interface SwoViewer permet la création et l'affichage simultané de plusieurs vues. Lors de la création d'une nouvelle vue, celle-ci peut être :

- Importée depuis un nouveau graphe ;
- Dupliquée depuis la vue en cours en gardant les IDs des éléments identiques ;
- Dupliquée depuis la vue en cours avec création de nouveaux IDs uniques pour les éléments du graphe.

Le choix de conserver ou non les IDs d'origine d'une vue dupliquée a un impact sur la connexion des vues entre elles : deux vues dont les éléments ont les mêmes IDs vont réagir simultanément lors du survol et de la sélection d'un nœud dans une seule des deux vues.



Styles d'affichage

L'utilisateur a la possibilité d'appliquer un *graphe de style* à une vue. Le graphe de style permet de définir les tailles, les couleurs et surtout les formes des éléments du graphe en fonction de leur type.

Annexe J

Identification des éléments constants d'un GMD

Algorithm 1 Comparaison de deux groupes Gp_1 et Gp_2 avec les paramètres *perSimi* et *perSize*

perSimi : percentage of minimal similarity
perSize : percentage of maximal size gap
$resComp \leftarrow false$
if $size(Gp_1) > 4$ && $size(Gp_2) > 4$ **then**
  if $size(Gp_1) \leq size(Gp_2)$ **then**
    if $\frac{size(Gp_2)-size(Gp_1)}{size(Gp_2)} \leq perSize$ **then**
      $resComp \leftarrow compare(Gp_1, Gp_2, perSimi)$
    end if
  else
    if $\frac{size(Gp_1)-size(Gp_2)}{size(Gp_1)} \leq perSize$ **then**
      $resComp \leftarrow compare(Gp_2, Gp_1, perSimi)$
    end if
  end if
end if
return *resComp*

METHOD compare(Gp_1,Gp_2,*perSimi*)
$resComp \leftarrow false$
$count \leftarrow 0$
for $node \in Gp_1$ **do**
  if $node \in Gp_2$ **then**
    $count \leftarrow count + 1$
  end if
end for
$similarity \leftarrow count * 100/size(Gp_1)$
if $similarity \geq perSimi$ **then**
  $resComp \leftarrow true$
end if
return *resComp*

Algorithm 2 Identification des groupes communs et les motifs en contexte avec les paramètres *perSimi* et *perSize*

perSimi : percentage of minimal similarity
perSize : percentage of maximal size gap
ctxtStes : list of states of the context
cSte : current state
patternsList : list of patterns, initialized to empty for each group of the context
for $iCste \in ctxtStes$ **do**
  for $group(cSte) \in groups(cSte)$ **do**
    for $group(iCste) \in groups(iCste)$ **do**
      if $compareGroups(cSte, iCste, perSimi, perSize)$ **then**
        $id(iCste) \leftarrow id(cSte)$
        $pattern(cSte) \leftarrow patternsList(cSte)$
        $pattern(cSte) \leftarrow comparePatterns(motif(cSte), iCste)$
        $patternsList(cSte)$ *append* $pattern(cSte)$
      end if
    end for
  end for
end for
return *patternsList*

METHOD comparePatterns(pattern,group)

patternNeo
for $node \in pattern$ **do**
  if $node \in group$ **then**
    patternNeo append node
  end if
end for
return *patternNeo*

La méthode compareGroups(group1,group2,perSimi,perSize) correspond à l'algorithme 1 présenté dans cette annexe.

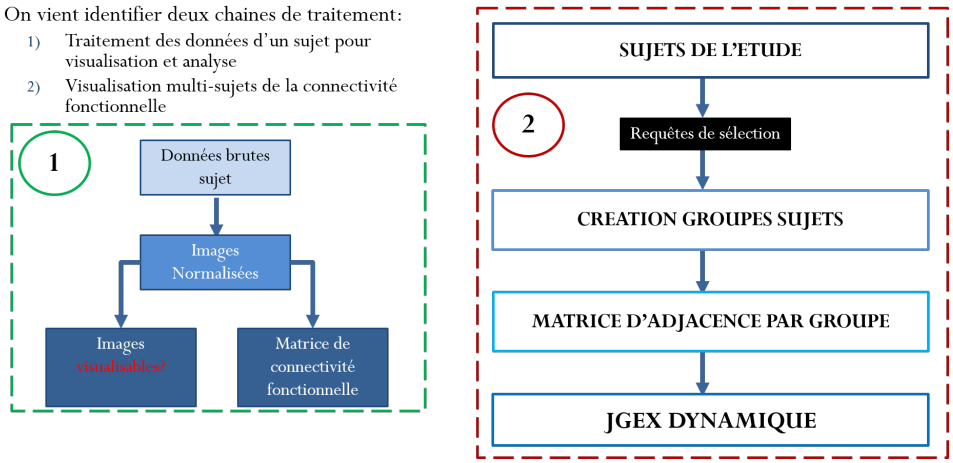
Algorithm 3 Identification des nœuds constants pour des graphes de connectivité fonctionnelle

input graph G, $iPerFixedNodes$, $iPerNodesAct$, $iThresholdAct$, $iThresholdInact$, $iThresholdCc$, $iThresholdSdCc$
$fixedToGo \leftarrow iPerFixedNodes$
$actFixed \leftarrow 0$
$inactFixed \leftarrow 0$
$averMax \leftarrow max(nodeDegreeSum \in G)$
$i \leftarrow 0$
repeat
    $i \leftarrow i + 1$
    if $actFixed < iPerNodesAct$ **then**
        for every node $v \in G$ **do**
            $average \leftarrow nodeDegreeSum(v)$
            $stanDevAver \leftarrow standardDeviationAverage(v)$
            $cc \leftarrow nodeChangeCentralitySum(v)$
            $stanDevCc \leftarrow standardDeviationChangeCentralitySum(v)$
            $thresMax \leftarrow abs((100 - (iThresholdAct * (i - 1)) * \frac{averMax}{100})$
            $thresMin \leftarrow abs((100 - (iThresholdAct * i)) * \frac{averMax}{100})$
            if $(average > thresMin)$ and $(average \leq thresMax)$ **then**
                if $cc \leq iThresholdCc$ **then**
                    $fixed(n) \leftarrow true$
                    $fixedToGo \leftarrow fixedToGo - 1$
                    $actFixed \leftarrow actFixed - 1$
                else
                    if $(stanDevAver \leq (thresMax - thresMin))$ and $(stanDevCc \leq iThresholdSdCc)$ **then**
                        $fixed(n) \leftarrow true$
                        $fixedToGo \leftarrow fixedToGo - 1$
                        $actFixed \leftarrow actFixed - 1$
                    end if
                end if
            end if
        end for
    end if
    if $inactFixed < (iPerFixedNodes - iPerNodesAct)$ **then**
        for every node $v \in G$ **do**
            $average \leftarrow nodeDegreeSum(v)$
            $stanDevAver \leftarrow standardDeviationAverage(v)$
            $thresMax \leftarrow abs(\frac{iThresholdInact*(i-1)*(size(G)-1)}{100})$
            $thresMin \leftarrow abs(\frac{iThresholdInact*i*(size(G)-1)}{100})$
            if $(average \geq thresMin)$ and $(average < thresMax)$ **then**
                if $(stanDevAver \leq ((thresMax - thresMin) * i))$ **then**
                    $fixed(n) \leftarrow true$
                    $fixedToGo \leftarrow fixedToGo - 1$
                    $inactFixed \leftarrow inactFixed - 1$
                end if
            end if
        end for
    end if
until $(fixedToGo > 0)$ or $(i = 1000)$

Annexe K

Use case du projet BIOMIST

Le document présenté ci-dessous a été rédigé par Marianne Allanic et Arthur Grioche dans le cadre du projet BIOMIST.

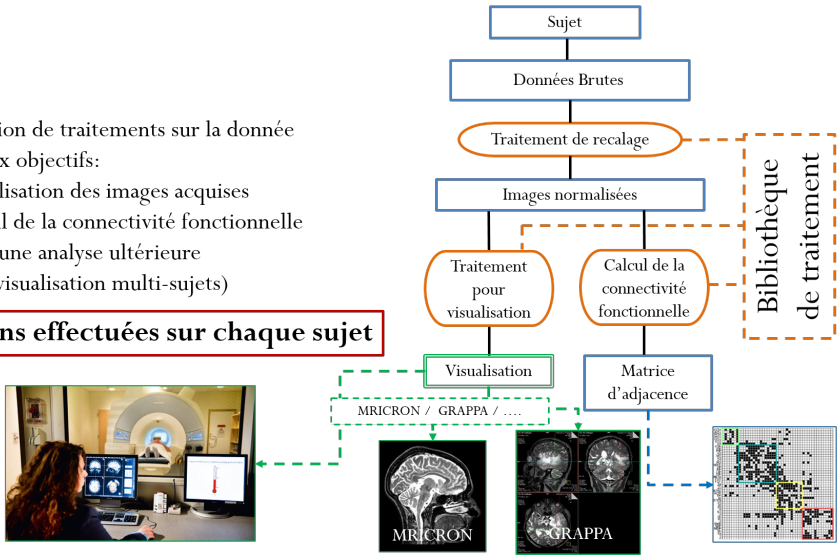


Cas d'usage : deux grandes étapes

Application de traitements sur la donnée avec deux objectifs:

- Visualisation des images acquises
- Calcul de la connectivité fonctionnelle pour une analyse ultérieure (e.g. visualisation multi-sujets)

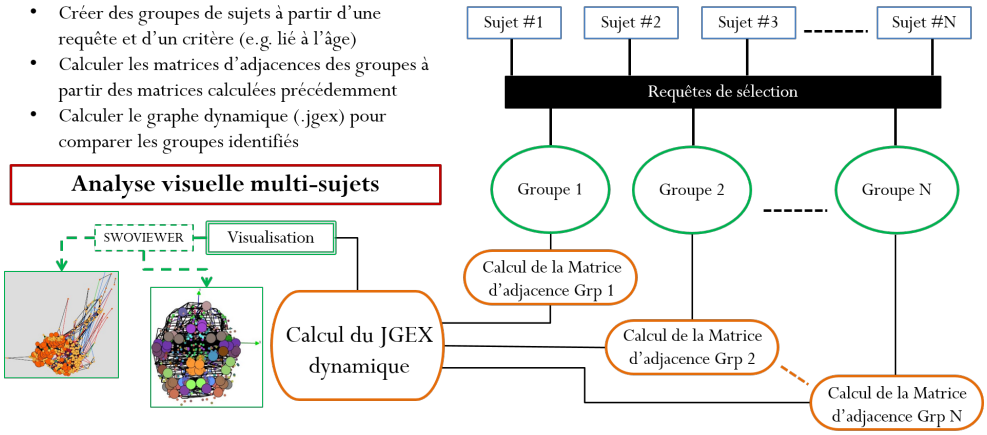
Opérations effectuées sur chaque sujet



Étape 1 – Traitement des images

- Créer des groupes de sujets à partir d'une requête et d'un critère (e.g. lié à l'âge)
- Calculer les matrices d'adjacences des groupes à partir des matrices calculées précédemment
- Calculer le graphe dynamique (.jgex) pour comparer les groupes identifiés

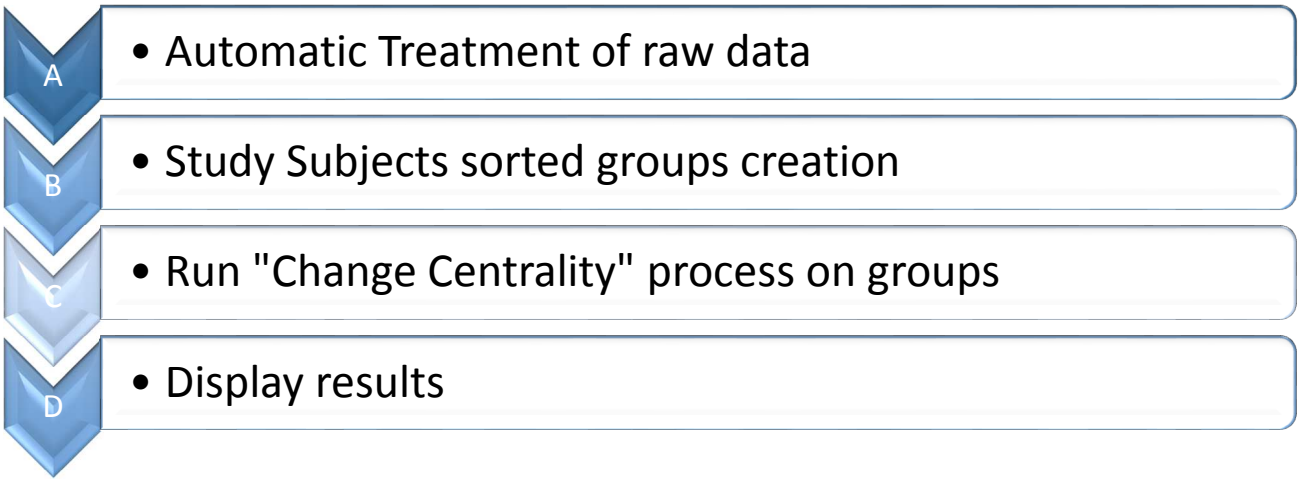
Analyse visuelle multi-sujets



Étape 2 – Visualisation dynamique de la connectivité fonctionnelle

Demonstration UseCase:
Running Change Centrality Process

This Use Case demonstrate Biomist use for BIM application. It is focus on a treatment chain reflecting Use Case done by BIM scientists in their work.
First table explains all main steps and then each point is developed one by one to facilitate understanding. The graph hereafter offers a view of the whole process.



Glossary:

- Study subjects: subjects into a study
- Groups: Contains subjects, it is a selection of study subjects
- Class: Define subject groups created from a previous group of subject.

How to read the document, in each part there is:

- A table that gives the running way,
- Tables below that give parameters used during the process and
- Scheme that explicit the process if possible.

Global Use Case

Use Case ID:	Global view		
Use Case Name:	Process running (specific test case – link with graph)		
Created By:	Grioche A.	**Last Updated By:**	Grioche A.
Date Created:	07/04/2015	**Last Revision Date:**	10/08/2015
Actors:	Scientist user		
Description:	Browse the complete action path possibility for a user that operates actions on data acquisition. The user select raw data, process them and analyze a sample of subjects into graphic interface.		
Trigger:	Use this case as demonstration		
Preconditions:	Subject acquisitions done and accessible for user		
Postconditions:	It is possible to visualize on dynamic graph acquisitions for groups of study subjects regarding their centrality		
Normal Flow:	A. Automatic treatment of raw data a. Select study subjects and create a group as SubGroup object b. Calculate functional connectivity data into adjacency matrix (get CSV and JSON static for each study subject) c. Calculate JSON dynamic for the group d. Calculate Change Centrality between subjects in the group (.csv) B. Study Subjects sorted classes creation a. Open previous study subject group into JMP b. Order and sort study subjects regarding parameters c. Extract .csv with ordered classes C. Run “Change Centrality” process on classes a. Calculate average for each class b. Calculate Change Centrality between classes created D. Display results a. Launch Graphs calculation b. Display graphs into the interface (web Biomist client)		
Alternative Flows:	To be defined		
Assumptions:	- System and device are operative - ToolBox created: Converter CSV, Average group Calculation, Change Centrality Calculator		
Notes and Issues:			

A. Automatic Treatment

Use Case ID:	A_AutoTreatment		
Use Case Name:	Automatic treatment of raw data		
Created By:	Grioche A.	**Last Updated By:**	Allanic M.
Date Created:	29/04/2015	**Last Revision Date:**	29/07/2015
Actors:	Scientist user		
Description:			
Trigger:	Use this case as demonstration		
Preconditions:			
Postconditions:			
Normal Flow:	a. Select study subjects and create a group as SubGroup object ▪ Xquery select study subject regarding parameters form Tab1.1 ▪ Create a new SubGroup object named GPS_DemoGroup01 and link Subjects selected ▪ Create DemoGroup01.xml as a file with all data for each subject b. Calculate connectivity of intrinsic functional data into adjacency matrix and graphs (JSON static) for each study subject ▪ Use 'BlackBox' of PYH - Select all raw data for subjects - Convert data into CSV - Return Data treated to TC ▪ Classify data under each subjects into TC c. Calculate JSON dynamic for the group ▪ Create a new WorkFlowInput named WFI_JSONDynDemo - Link GPS_DemoGroup01 - Link PCD_JSONDynDemo (Look at Tab1.2.1) - Link PCP_JSONDynDemo (Look at Tab1.2.2) - Link EXDef, ACDef and DUDef(Look at Tab1.2.3) ▪ Create a new Processing named PCR_JSONDynDemo - Add WorkFlowInput as Target ▪ Launch processing d. Calculate Change Centrality between subjects in the group (.csv) ▪ Create a new WorkFlowInput named WFI_CCInterSubjectDemo - Link GPS_DemoGroup01 - Link PCD_ CCInterSubjectDemo (Look at Tab1.3.1) - Link PCP_ CCInterSubjectDemo (Look at Tab1.3.2) - Link EXDef, ACDef and DUDef (Look at Tab1.3.3) ▪ Create a new Processing named PCR_ CCInterSubjectDemo - Add WorkFlowInput as Target ▪ Launch processing		
Alternative Flows:	To be defined if Query Builder interface		
Notes and Issues:			

❖ Table

Tab1.2.1: PCD_JSONDynDemo (Processing Defintion)			
Item_id	=	PCDima_nipype_VoxelBasedMorphometry	Get global process definition
Obtain file and load: VoxelBasedMorphometry.py			

Tab1.2.2: PCP_JSONDynDemo (Processing Parameters)			
Item_id	=	PCPima_T1list	Get parameters for process
Item_id	=	PCPima_SegSmooth	
Item_id	=	PCPima_subjectlist	
Item_id	=	PCPima_T1_Segment	

Tab1.2.3: EXDef, ACDef and DUDef (Defintion results)				
Exam Defintion				
Item_id	=	'EXDef'	'role'	Get definition to recover dataset
Item_id	=	'EXDef'	role'	
Acquisition Defintion				
Item_id	=	'AcqDef'	'role'	Get definition to recover dataset
Item_id	=	'AcqDef'	role'	
Data Unit Defintion				
Item_id	=	'DUDef'	'role'	Get definition to recover dataset
Item_id	=	'DUDef'	role'	

Tab1.3.1: PCD_ CCInterSubjectDemo (Processing Defintion)			
Item_id	=	PCDima_nipype_VoxelBasedMorphometry	Get global process definition
Obtain file and load: VoxelBasedMorphometry.py			

Tab1.3.2: PCP_ CCInterSubjectDemo (Processing Parameters)			
Item_id	=	PCPima_T1list	Get parameters for process
Item_id	=	PCPima_SegSmooth	
Item_id	=	PCPima_subjectlist	
Item_id	=	PCPima_T1_Segment	

Tab1.3.3: EXDef, ACDef and DUDef (Defintion results)				
Exam Defintion				
Item_id	=	'EXDef'	'role'	Get definition to recover dataset
Item_id	=	'EXDef'	role'	

Acquisition Defintion				
Item_id	=	'AcqDef'	'role'	Get definition to recover dataset
Item_id	=	'AcqDef'	role'	
Data Unit Defintion				
Item_id	=	'DUDef'	'role'	Get definition to recover dataset
Item_id	=	'DUDef'	role'	

❖ Scheme

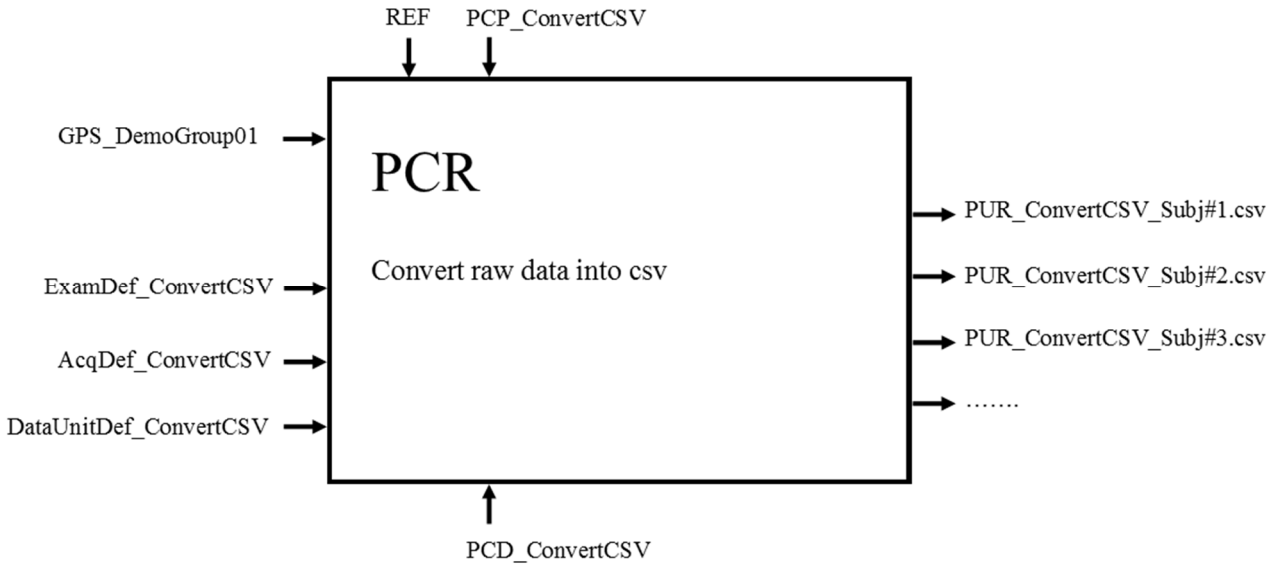


Figure 1: Scheme Convert into CSV

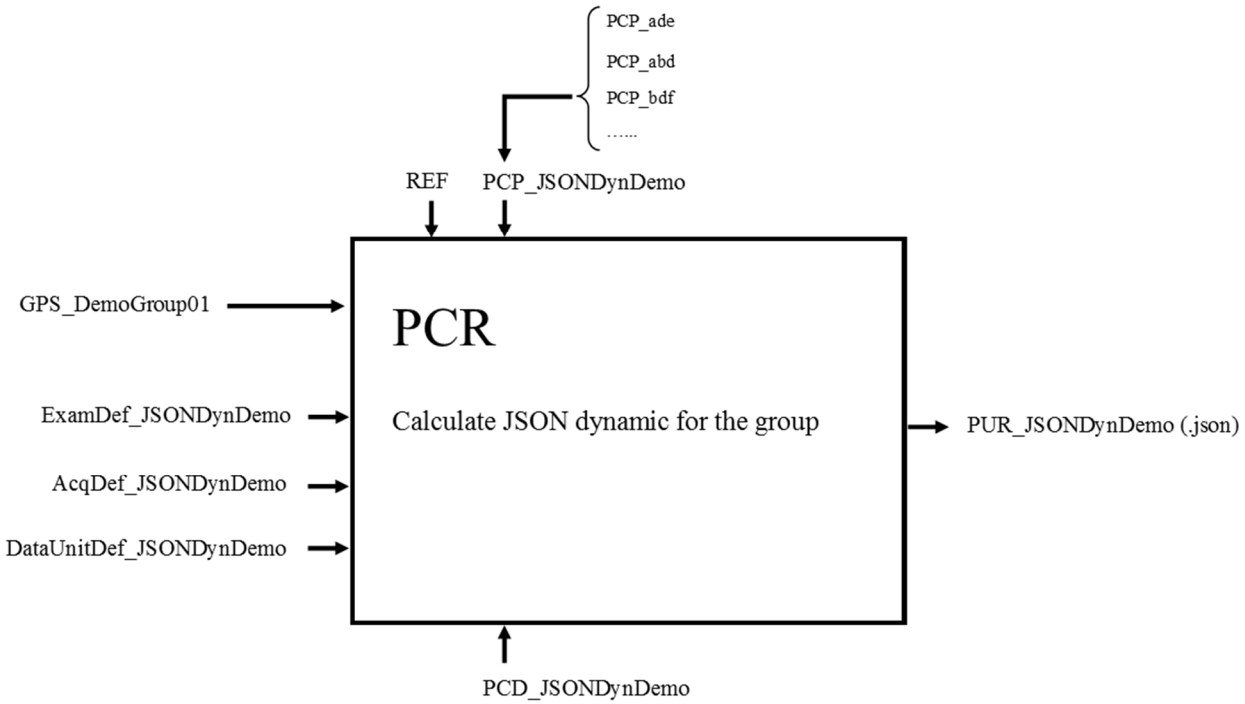


Figure 2 : Change Centrality inter-subjects

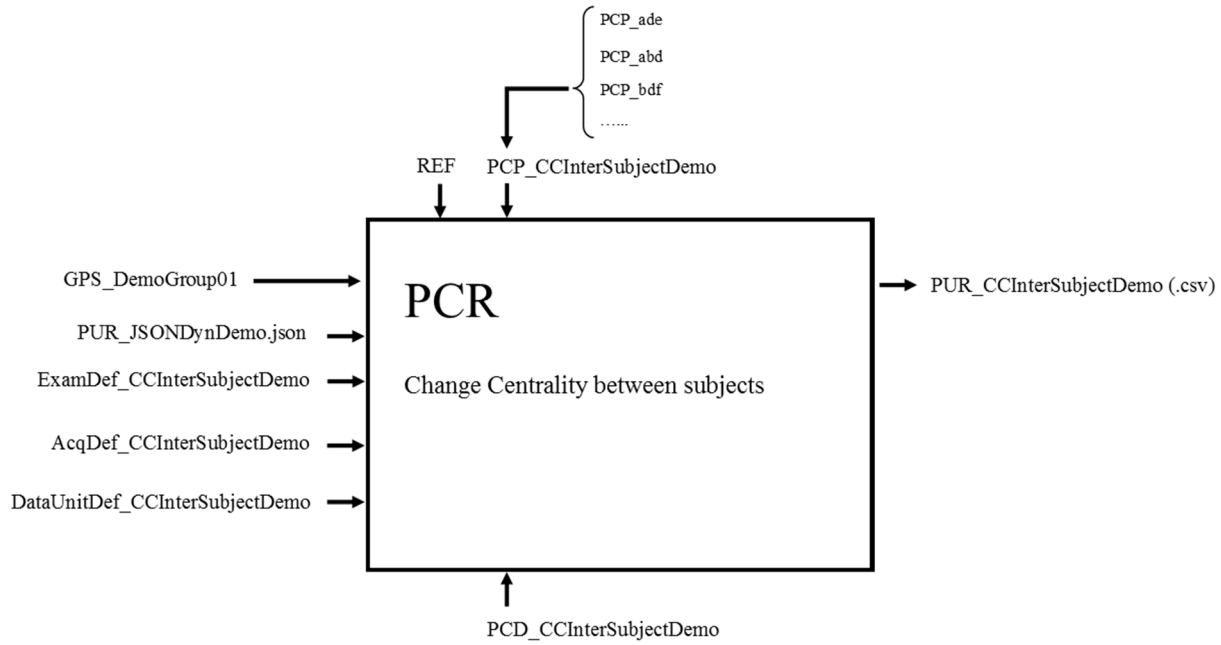


Figure 3 : Scheme Change Centrality inter-subjects

B. Group Creation

Use Case ID:	B_ Group Creation		
Use Case Name:	Study Subjects sorted groups creation		
Created By:	Grioche A.	**Last Updated By:**	Grioche A.
Date Created:	29/04/2015	**Last Revision Date:**	18/05/2015
Actors:	Scientist user		
Description:	Create subjects groups and manipulate data to generate a		
Trigger:	Use this case as demonstration		
Preconditions:	Prepare PUR object with data results		
Postconditions:			
Normal Flow:	a. Open into JMP data from subject of GPS_DemoGroup01 ▪ Do a query into the query builder and open results b. Order and sort study subjects regarding parameters ▪ Manipulate data: --define operation -- ▪ Apply a sorting operation: --define operation -- ▪ Save File as '…. .csv' c. Extract .csv with ordering groups		
Alternative urgency Flows:	a. Open into Excel data form the subject group GPS_DemoGroup01 b. Sort data with Excel regarding parameters into tab2.* c. In TC prepare for following tasks: ▪ The query to get GPS_ DemoGroup01 and display into Excel ▪ Use the query to create a new group into TC ▪ Create 3 groups of subjects with the criteria of Tab2.* GPS_DemoSubGroup01_A/GPS_DemoSubGroup01_B/GPS_DemoSubGroup01_C		
Notes and Issues:			

C. Change Centrality Processing

Use Case ID:	C_ ChangeCentralityProcessing		
Use Case Name:	Run "Change Centrality" process on groups		
Created By:	Grioche A.	**Last Updated By:**	Allanic M.
Date Created:	29/04/2015	**Last Revision Date:**	27/07/2015
Actors:	Scientist user		
Description:	Calculate the change centrality for the groups created previously		
Trigger:	Use this case as demonstration		
Preconditions:	Step A & B		
Postconditions:			

Normal Flow:	a. Calculate average for each group ▪ Create a new WorkFlowInput named WFI_AverageGrpDemo - Link : GPS_DemoSubGroup01_A, GPS_DemoSubGroup01_B, GPS_DemoSubGroup01_C - Link PCD_ AverageGrpDemo (Look at Tab3.1) - Link PCP_ AverageGrpDemo (Look at Tab3.2) - Link EXDef, ACDef and DUDef(Look at Tab3.3) ▪ Create a new Workflow process named PCR_ AverageGrpDemo - WFI_AverageGrpDemo as Target ▪ Launch processing : O/ averagegrp01_A.csv , averagegrp01_B.csv , averagegrp01_C.csv b. Calculate Change Centrality between groups created ▪ Create a new WorkFlowInput named WFI_CCGroupsDemo - Link GPS_DemoSubGroup01_A, GPS_DemoSubGroup01_B, GPS_DemoSubGroup01_C - Link PCD_ CCGroupsDemo (Look at Tab4.1) - Link PCP_ CCGroupsDemo (Look at Tab4.2) - Link EXDef, ACDef and DUDef(Look at Tab4.3) ▪ Create a new Workflow processing named PCR_ CCGroupsDemo - Add WFI_CCGroupsDemo as Target ▪ Launch processing: O/ ccGroupAB.csv, ccGroupBC.csv, ccGroupCA.csv
Notes and Issues:	

❖ Table

Tab3.1: PCD_ AverageGrpDemo (Processing Defintion)			
Item_id	=	PCDima_nipype_VoxelBasedMorphometry	Get global process definition
Obtain file and load: VoxelBasedMorphometry.py			

Tab3.2: PCP_ AverageGrpDemo (Processing Parameters)			
Item_id	=	PCPima_T1list	Get parameters for process
Item_id	=	PCPima_SegSmooth	
Item_id	=	PCPima_subjectlist	
Item_id	=	PCPima_T1_Segment	

Tab3.3: EXDef, ACDef and DUDef (Defintion results)				
Exam Defintion				
Item_id	=	'EXDef'	'role'	Get definition to recover dataset
Item_id	=	'EXDef'	role'	
Acquisition Defintion				
Item_id	=	'AcqDef'	'role'	Get definition to recover dataset
Item_id	=	'AcqDef'	role'	
Data Unit Defintion				
Item_id	=	'DUDef'	'role'	

Item_id	=	'DUDef'	role'	Get definition to recover dataset

Tab4.3: EXDef, ACDef and DUDef (Defintion results)				
Exam Defintion				
Item_id	=	'EXDef'	'role'	Get definition to recover dataset
Item_id	=	'EXDef'	role'	
Acquisition Defintion				
Item_id	=	'AcqDef'	'role'	Get definition to recover dataset
Item_id	=	'AcqDef'	role'	
Data Unit Defintion				
Item_id	=	'DUDef'	'role'	Get definition to recover dataset
Item_id	=	'DUDef'	role'	

❖ Scheme

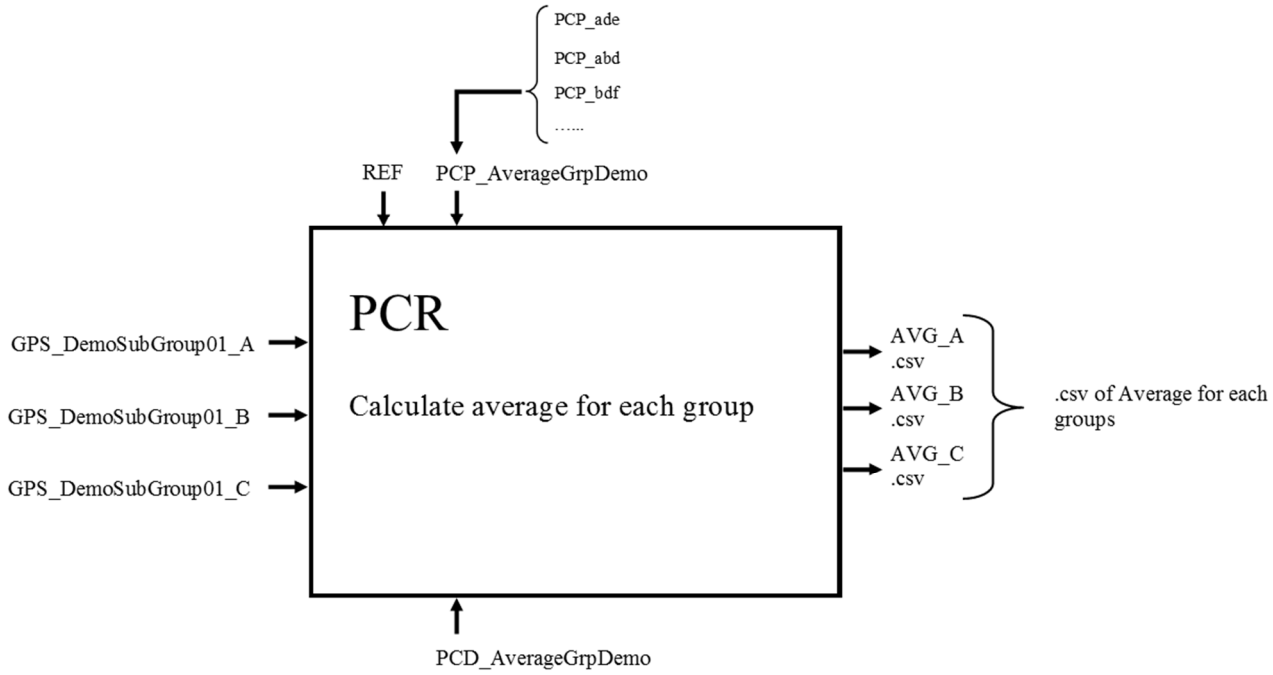


Figure 4: Average by group processing graph

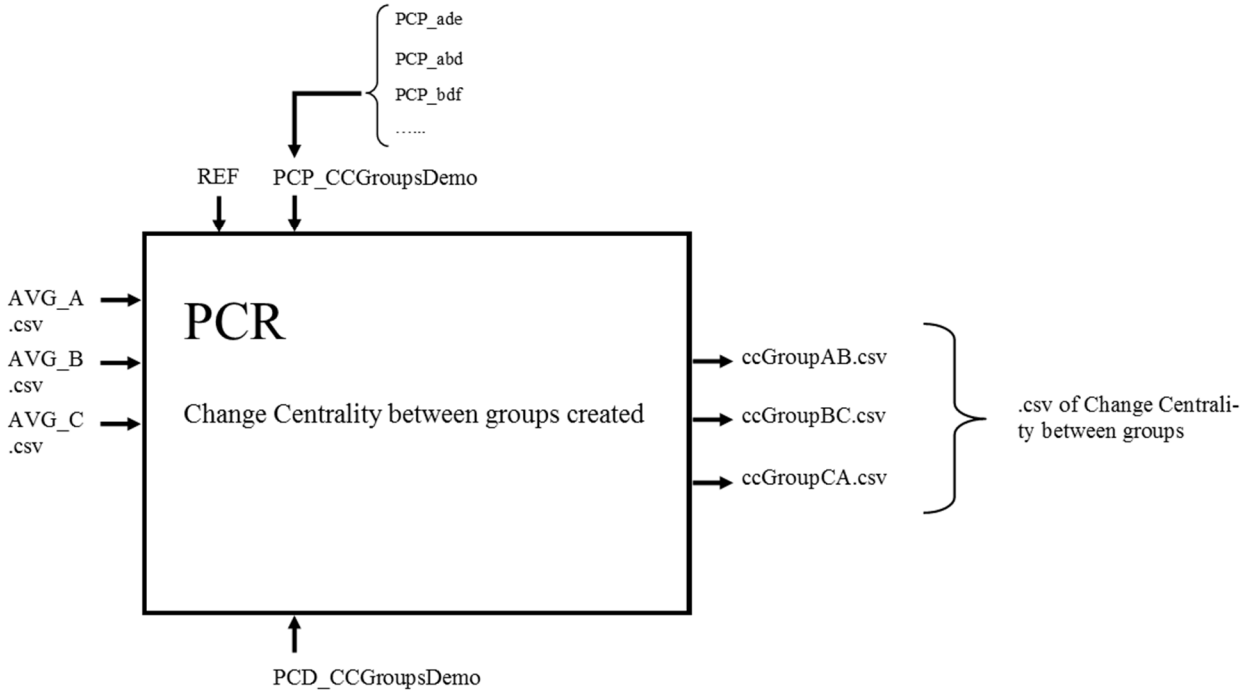


Figure 5 : Change Centrality between groups processing graph

D. Display Results

Use Case ID:	C_GraphCalculation		
Use Case Name:	Display result into Graph		
Created By:	Grioche A.	**Last Updated By:**	Grioche A.
Date Created:	29/04/2015	**Last Revision Date:**	29/04/2015
Actors:	Scientist user		
Description:	Give a space representation of the result to the user.		
Trigger:	Use this case as demonstration		
Preconditions:			
Postconditions:			
Normal Flow:	a. Launch Graphs calculation ▪ Visual comparison ▪ Static layout ▪ Clustering analysis ▪ Fixed layout b. Display graphs into the interface (web Biomist client)		
Notes and Issues:			

Annexe L

OCL : exploration suivant l'âge des sujets

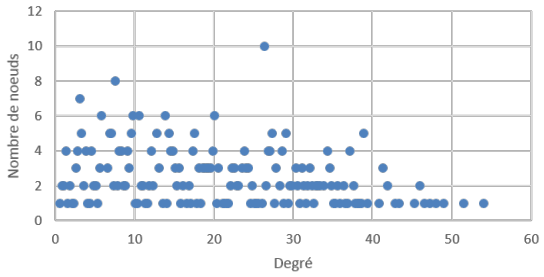
L'impact de l'âge des sujets sur la connectivité fonctionnelle cérébrale est analysé à travers quatre classes d'étude. La répartition des effectifs (218 sujets au total) dans chaque classe est donnée dans le tableau ci-dessous :

Classe		**20.5**	**25.5**	**30.5**	**35.5**
Effectif		93	82	27	16
Homme	Gauche	30.1%	25.6%	7.4%	18.8%
	Droite	15.1%	23.2%	37.0%	56.2%
Femme	Gauche	30.1%	24.4%	14.8%	6.2%
	Droite	24.7%	26.8%	40.7%	18.8%

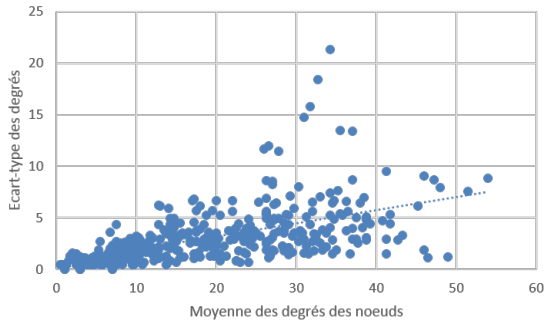
Dans un premier temps les paramètres de préparation des données sont déterminés de façon empirique. Puis, dans un second temps, des copies d'écran de l'exploration OCL sont présentées pour le jeu de paramètres retenu.

Détermination empirique des paramètres

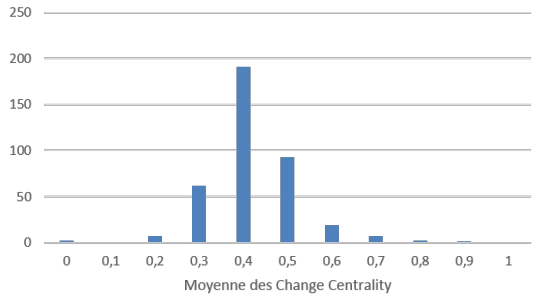
La moyenne et l'écart-type des degrés des nœuds ainsi que de la mesure du change centrality sont calculés sur les quatre classes d'âge. Les figures ci-dessous montrent : le nombre de nœuds par degré (a) et l'écart-type sur la moyenne en fonction du degré (b) ; la répartition de la mesure de change centrality des nœuds par tranche de 0.1 (c) et en fonction de l'écart-type sur la moyenne (d). Nous en déduisons un jeu de paramètres à tester : $iThesholdAct$=5.0, $iThresholdInac$=0.5, $iThresholdCc$=0.4, $iThresholdSdCc$=0.25.



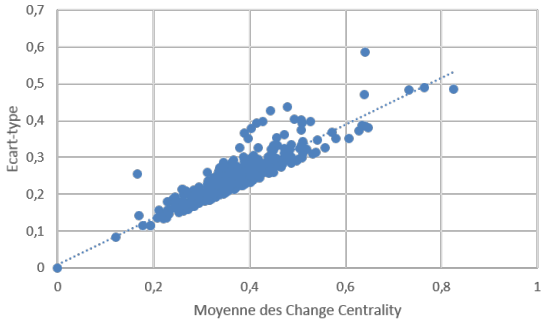
(a)



(b)



(c)



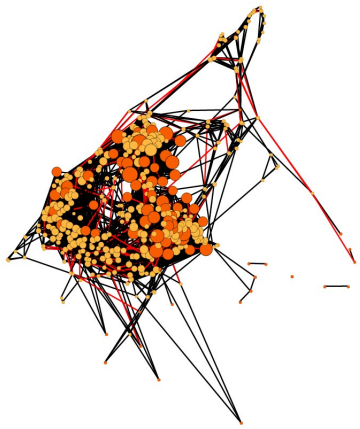
(d)

La préparation des données est lancée sur $iPerFixedNodes$ variant de 20% à 80%, pour $iPerNodesAct$=60%. Nous observons immédiatement qu'un pourcentage de nœuds fixes trop élevé nuit à l'identification de changement topologiques globaux.

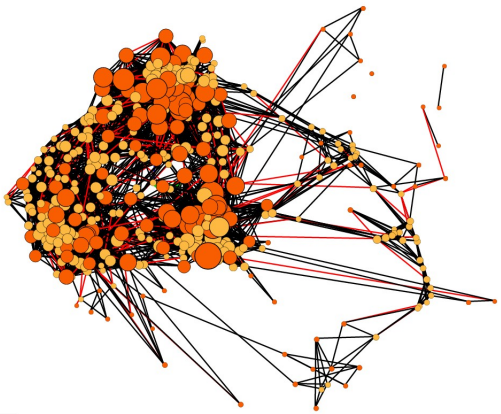
Dans un second nous réduisons donc notre analyse à la variation de $iPerFixedNodes$ entre 10% et 50%, à laquelle nous ajoutons la variation de $iPerNodesAct$ entre 50% et 70%. Le tableau ci-dessous présente le nombre de nœuds constants (actifs et inactifs) obtenus, ce qui permet d'évaluer au passage la pertinence de la détermination précédente de $iThesholdAct$, $iThresholdInac$, $iThresholdCc$ et $iThresholdSdCc$.

Les figures qui nous ont permis d'établir ce raisonnement sont données ci-après dans le document.

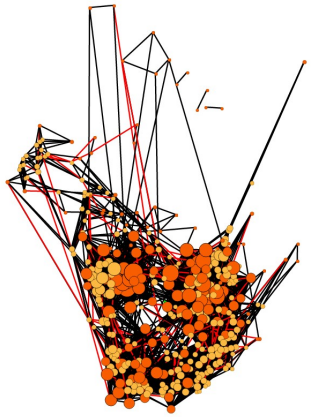
Nœuds constants				**Nœuds actifs**					**Nœuds inactifs**				
%th	th	re	e(%)	%th	th	re	e(%)	iter	%th	th	re	e(%)	iter
10	38.4	41	6.8	70	26.9	38	41.4	7	30	11.5	3	-74.0	7
10	38.4	45	17.2	60	23.0	24	4.2	6	40	15.4	21	36.7	14
10	38.4	45	17.2	50	19.2	24	25.0	6	50	19.2	21	9.4	14
20	76.8	77	0.3	70	53.8	56	4.2	8	30	23.0	21	-8.9	7
20	76.8	77	0.3	60	46.1	56	21.5	8	40	30.7	21	-31.6	8
20	76.8	77	0.3	50	38.4	38	-1.0	7	50	38.4	39	1.6	10
30	115.2	132	14.6	70	80.6	93	15.3	10	30	34.6	39	12.8	7
30	115.2	117	1.6	60	69.1	70	1.3	9	40	46.1	47	2.0	8
30	115.2	115	-0.2	50	57.6	56	-2.8	8	50	57.6	59	2.4	10
40	153.6	165	7.4	70	107.5	118	9.7	11	30	46.1	47	2.0	6
40	153.6	152	-1.0	60	92.2	93	0.9	10	40	61.4	59	-4.0	8
40	153.6	154	0.3	50	76.8	70	-8.9	9	50	76.8	84	9.4	10
50	192	197	2.6	70	134.4	138	2.7	12	30	57.6	59	2.4	6
50	192	202	5.2	60	115.2	118	2.4	11	40	76.8	84	9.4	8
50	192	189	-1.6	50	96.0	93	-3.1	10	50	96.0	96	0.0	9



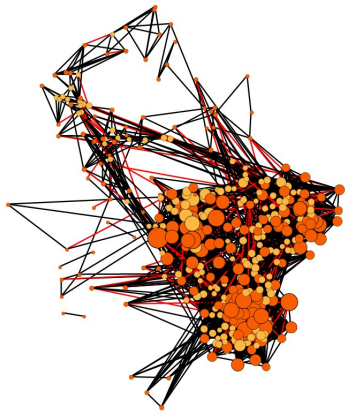
20%



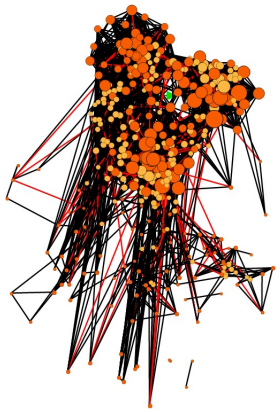
30%



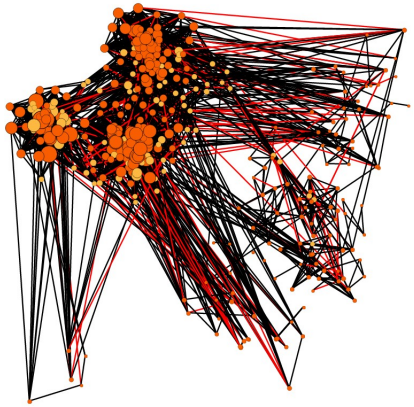
40%



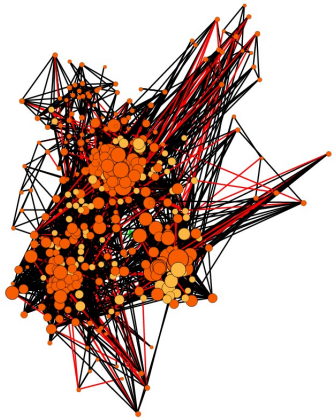
50%



60%

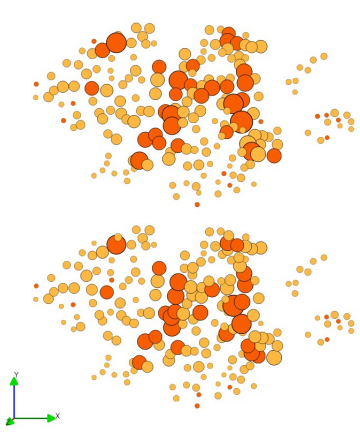


70%

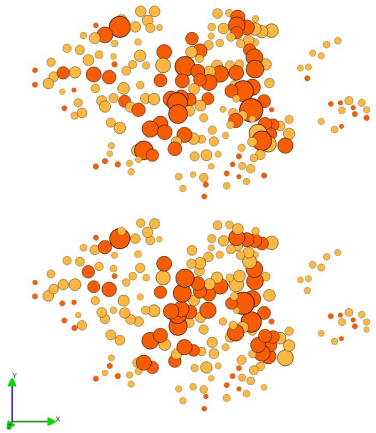


80%

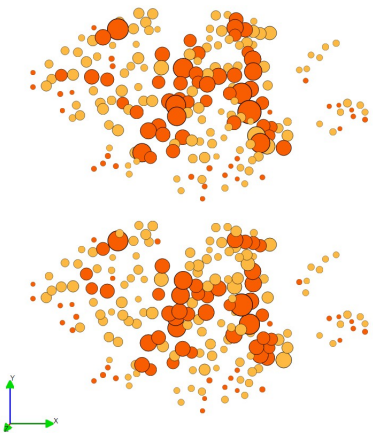
Layout LàC pour la classe d'âge $20,5$ en fonction du pourcentage total de nœuds fixes, pour un pourcentage de nœuds actifs réglé à 60%. Couleur des nœuds : orange=nœud constant, jaune=nœud libre. Couleur des arêtes : rouge=l'arête disparait à l'état suivant $(25,5)$.



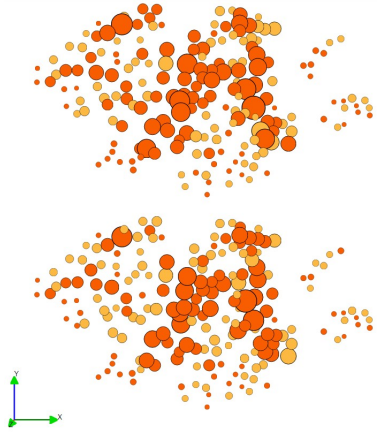
20%



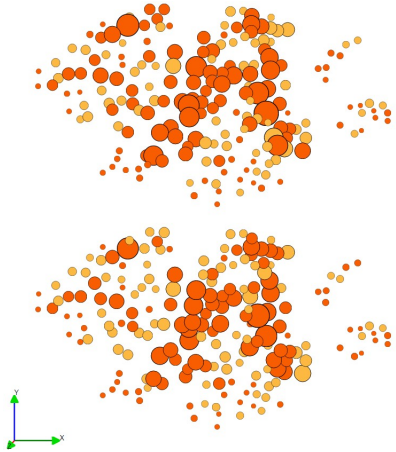
30%



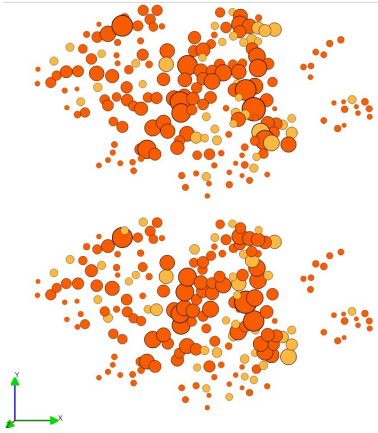
40%



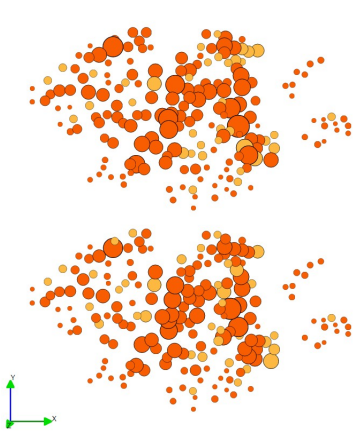
50%



60%



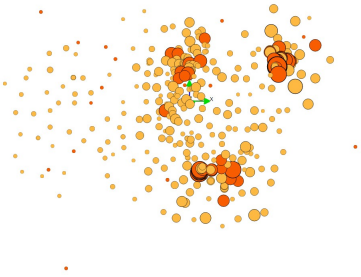
70%



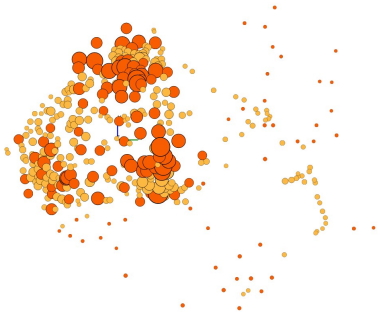
80%

Layout anatomique pour la classe d'âge $20,5$ en fonction du pourcentage total de nœuds fixes, pour un pourcentage de nœuds actifs réglé à 60%. Couleur des nœuds : orange=nœud constant, jaune=nœud libre. Diamètre des nœuds : degré des nœuds.

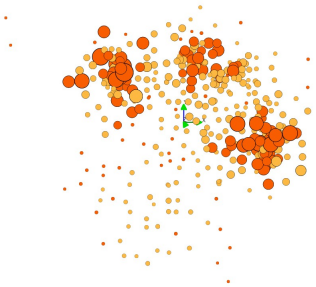
LàC $iPerFixedNodes$=20%



OpenORD sans fixation de nœuds

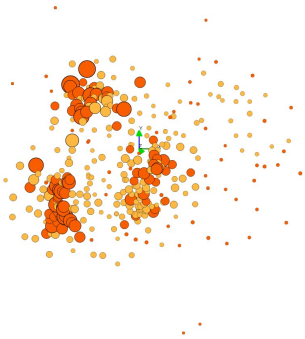


LàC $iPerFixedNodes$=30%

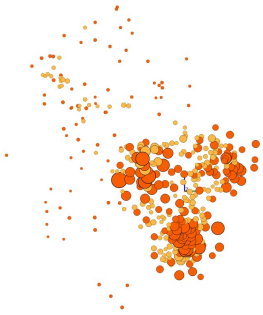


OpenORD sans fixation de nœuds

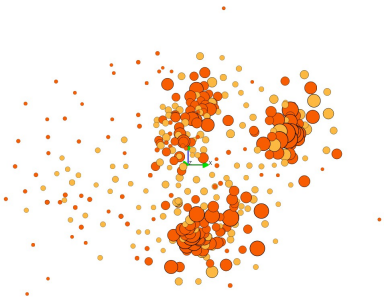
LàC $iPerFixedNodes$=40%



OpenORD sans fixation de nœuds

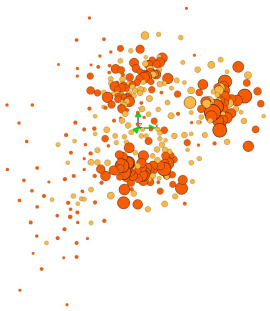


LàC $iPerFixedNodes$=50%

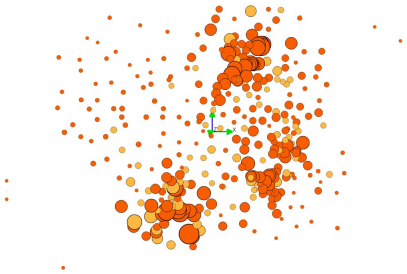


OpenORD sans fixation de nœuds

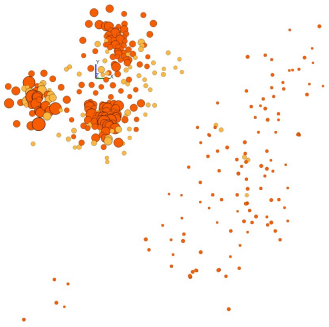
LàC $iPerFixedNodes$=60%



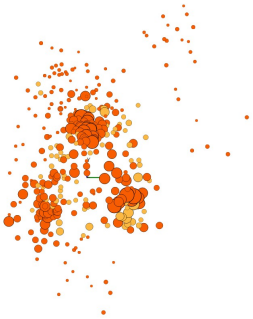
OpenORD sans fixation de nœuds



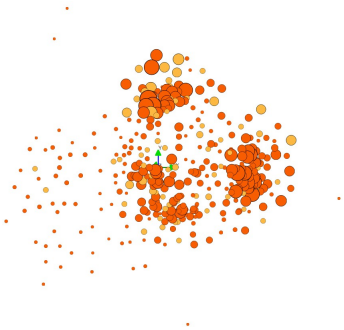
LàC $iPerFixedNodes$=70%



OpenORD sans fixation de nœuds



LàC $iPerFixedNodes$=80%



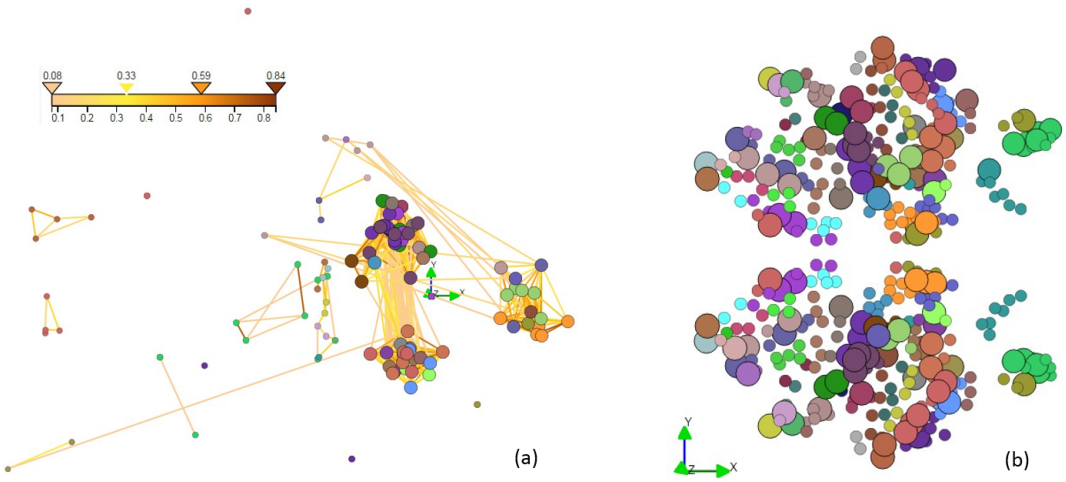
OpenORD sans fixation de nœuds

Résultats pour le jeu de paramètres choisi

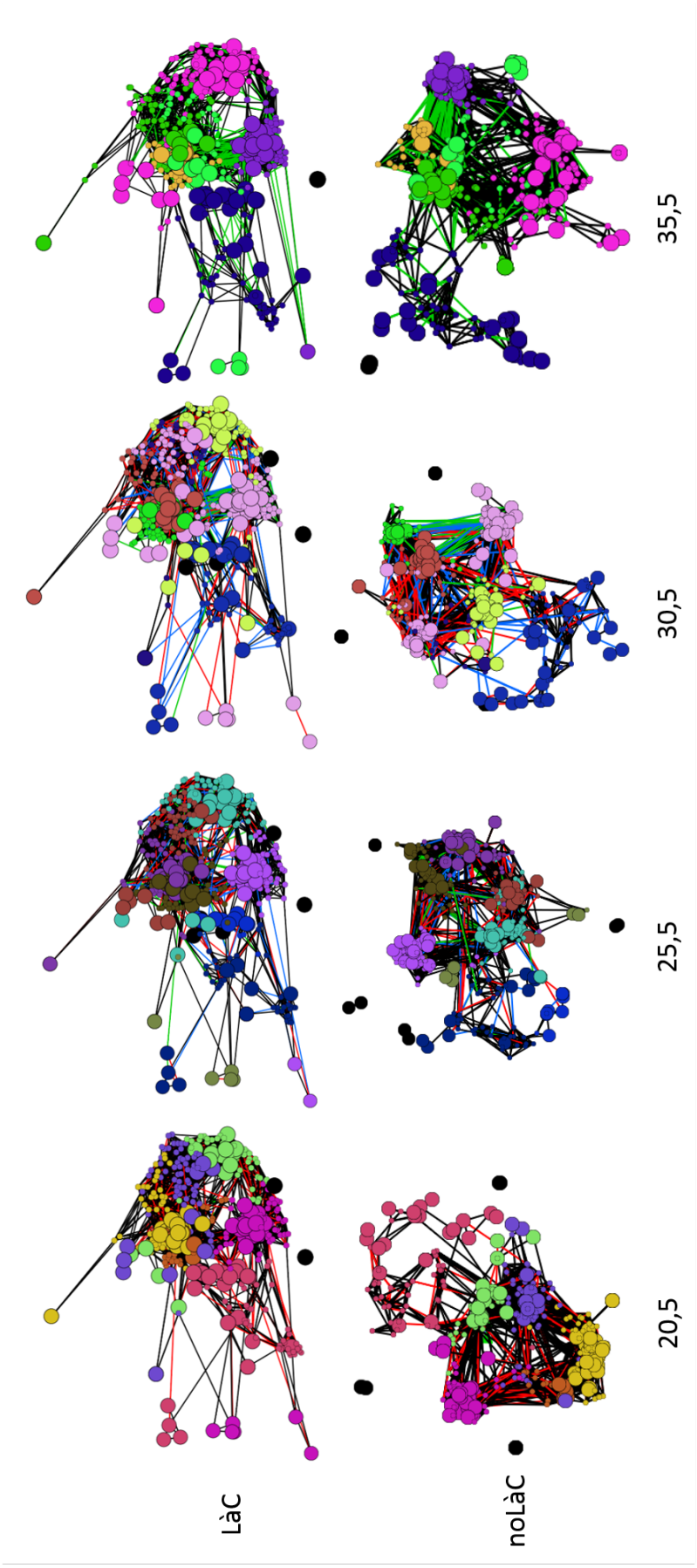
A la comparaison visuelle des résultats, nous retenons le jeu de paramètres suivant :

Paramètre	Valeur
$iThresholdEdges$	95%
$iPerFixedNodes$	30%
$iPerNodesAct$	60%
$iTheshold Act$	5.0
$iThresholdInac$	0.5
$iThresholdCc$	0.4
$iThresholdSdCc$	0.25

Graphe de synthèse obtenu sur la dimension âge : (a) LàC des nœuds constants, (b) identification des nœuds constants (taille=grand) sur le layout anatomique.



La figure ci-dessous montre le résultat du LàC pour les quatre classes d'âge, et donne à titre de comparaison le layout OpenORD sans fixation de nœuds. Les couleurs des nœuds correspondent à une colorisation aléatoire du clustering.


LàC
noLàC
35,5
30,5
25,5
20,5

Annexe M

OCL : exploration suivant le genre et la latéralité des sujets

L'impact du genre et de la latéralité des sujets sur la connectivité fonctionnelle cérébrale est analysé à travers quatre classes d'étude. La répartition des effectifs (231 sujets au total) dans chaque classe est donnée dans le tableau ci-dessous :

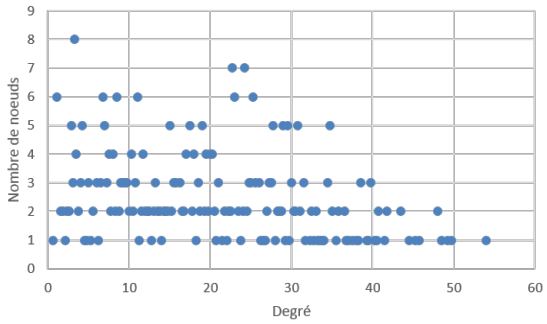
Genre	**Latéralité**	**Effectifs**
Femme	Gauche	56
	Droite	58
Homme	Gauche	55
	Droite	62

Dans un premier temps les paramètres de préparation des données sont déterminés de façon empirique. Puis, dans un second temps, des copies d'écran de l'exploration OCL sont présentées pour le jeu de paramètres retenu.

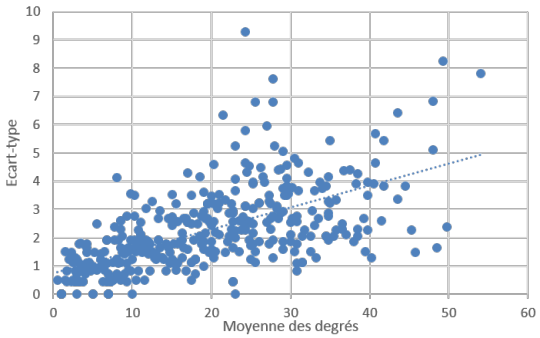
Détermination empirique des paramètres

La moyenne et l'écart-type des degrés des nœuds ainsi que de la mesure du change centrality sont calculés sur les quatre classes de genre et de latéralité {Femme Droitière, Femme Gauchère, Homme Gaucher, Homme Droitier}. Les figures ci-dessous montrent : le nombre de nœuds par degré (a) et l'écart-type sur la moyenne en fonction du degré (b) ; l'écart-type sur la moyenne en fonction de la mesure du change centrality (c). Nous constatons que les treize sujets supplémentaires inclus dans les classes de genre et de latéralité – ils sont en dehors des quatre classes d'âge – n'ont pas d'impact significatif sur la moyenne des degrés des nœuds. Concernant la moyenne de la mesure du change centrality, nous constatons qu'elle est plus basse sur l'ensemble des nœuds qu'avec les classes sur l'âge : à première vue, il y a moins de changements sur les arêtes entre les classes de genre et de latéralité qu'entre les classes sur l'âge. Nous conservons donc les valeurs des paramètres $iThesholdAct$ et $iThresholdInac$ à ceux choisis pour l'étude sur l'âge, mais nous adaptons les valeurs des paramètres $iThresholdCc$ et $iThresholdSdCc$. Nous obtenons

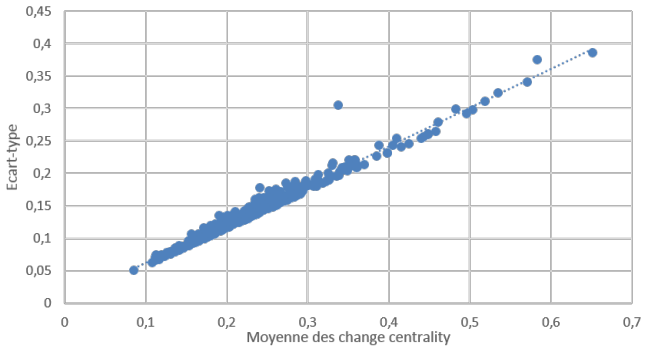
le jeu de paramètres à tester : $iThesholdAct$=5.0, $iThresholdInac$=0.5, $iThresholdCc$=0.3, $iThresholdSdCc$=0.2.



(a)



(b)



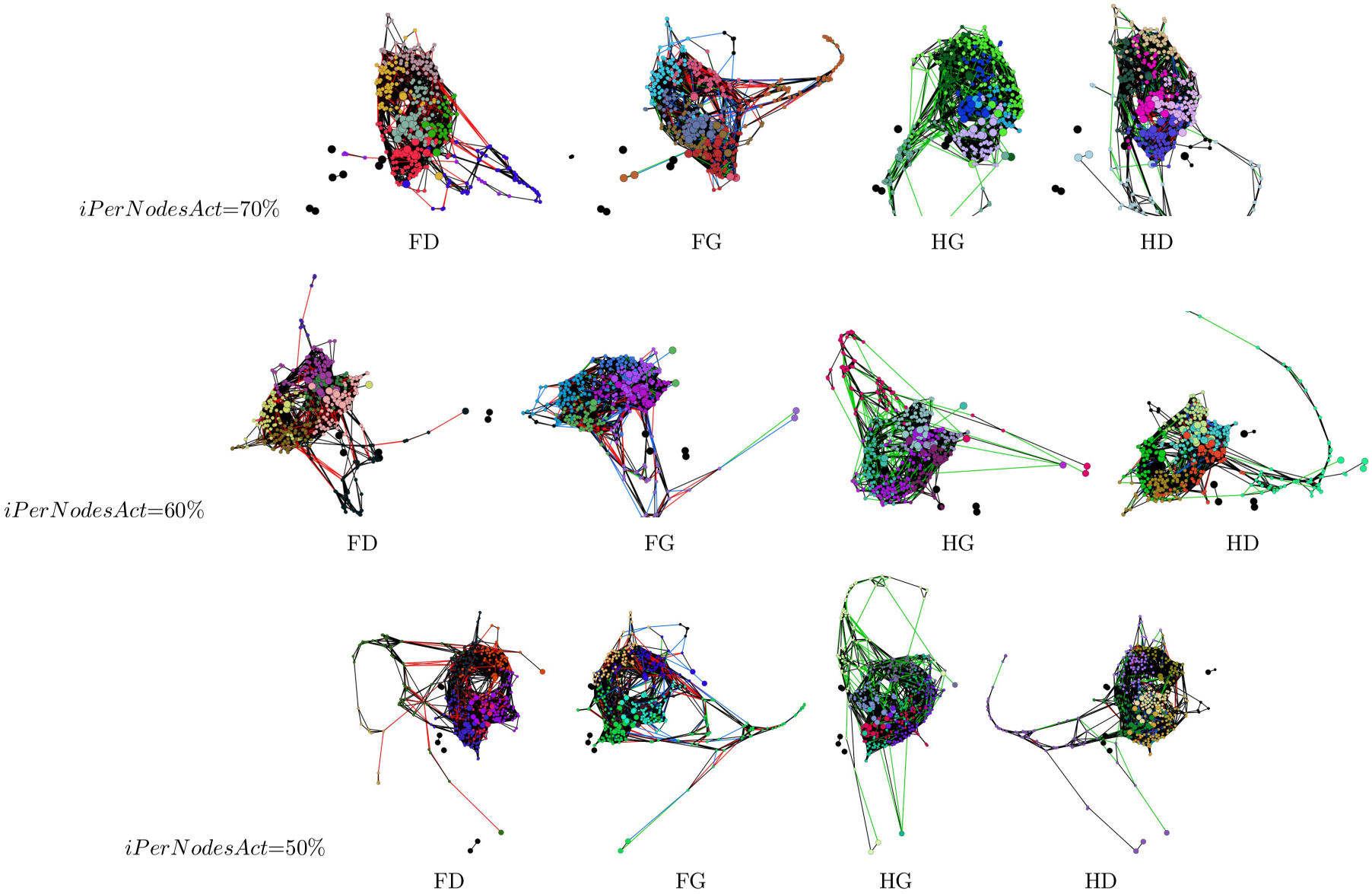
(c)

Comme pour l'étude avec des classes d'âges, nous réduisons donc notre analyse à la variation de $iPerFixedNodes$ entre 10% et 50%, à laquelle nous ajoutons la variation de $iPerNodesAct$ entre 50% et 70%. Le tableau ci-dessous présente le nombre de nœuds constants (actifs et inactifs) obtenus, ce qui permet d'évaluer au passage la pertinence de la détermination précédente de $iThesholdAct$, $iThresholdInac$, $iThresholdCc$ et $iThresholdSdCc$.

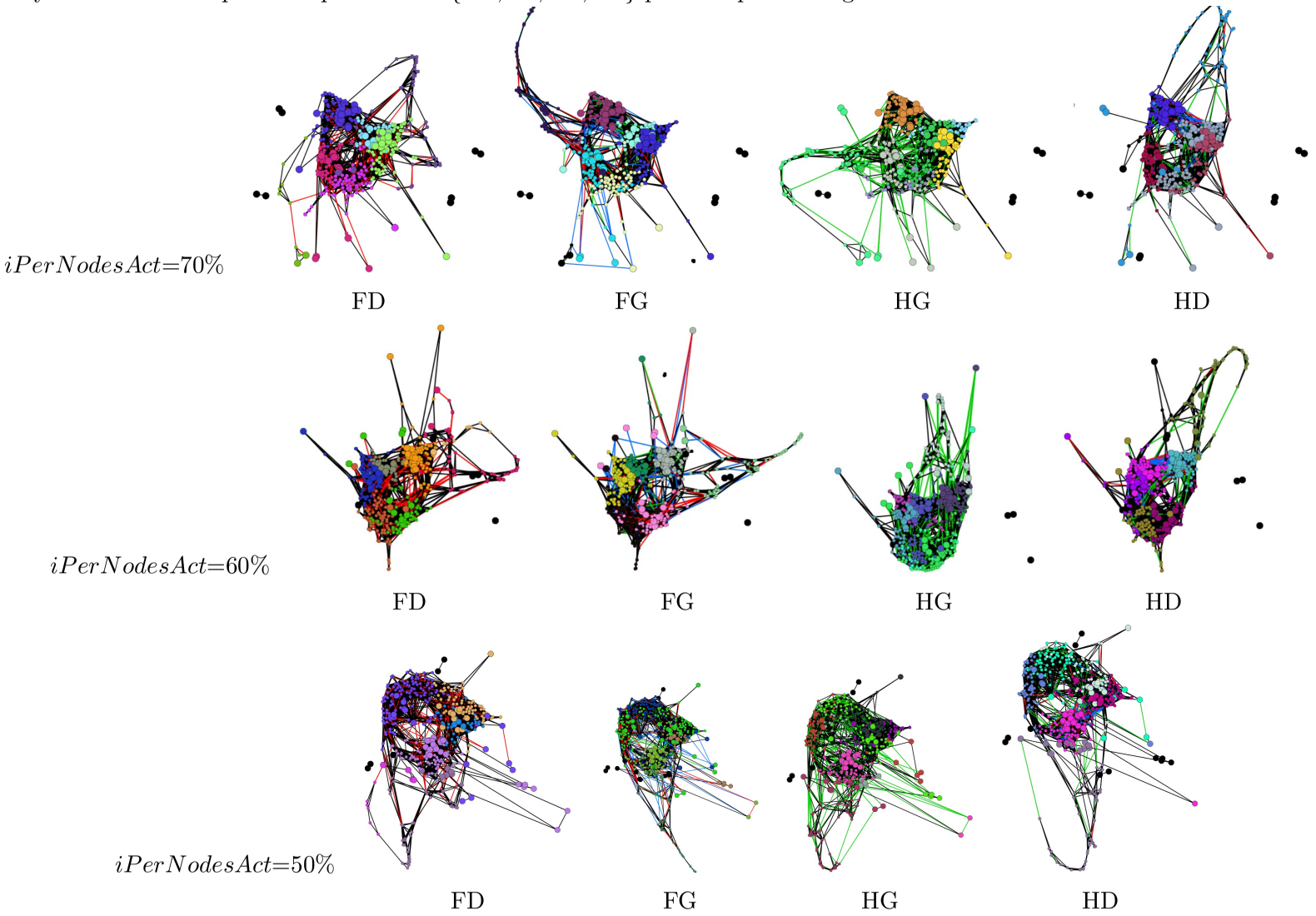
Les figures qui nous ont permis d'établir ce raisonnement sont données ci-après dans le document.

Nœuds constants				**Nœuds actifs**					**Nœuds inactifs**				
%th	th	re	e(%)	%th	th	re	e(%)	iter	%th	th	re	e(%)	iter
10	38.4	38	-1.0	70	26.9	27	0.4	6	30	11.5	11	-4.5	9
10	38.4	38	-1.0	60	23.0	27	14.7	6	40	15.4	11	-28.4	9
10	38.4	38	-1.0	50	19.2	27	28.9	6	50	19.2	11	-42.7	9
20	76.8	77	0.3	70	53.8	56	4.0	8	30	23.0	21	-8.9	7
20	76.8	77	0.3	60	46.1	56	17.7	8	40	30.7	21	-31.6	8
20	76.8	74	-3.6	50	38.4	36	-6.7	7	50	38.4	38	-1.0	10
30	115.2	141	22.4	70	80.6	103	21.7	10	30	34.6	38	10.0	7
30	115.2	119	3.3	60	69.1	71	2.6	9	40	46.1	48	4.2	8
30	115.2	110	-4.5	50	57.6	56	-2.9	8	50	57.6	54	-6.3	10
40	153.6	151	-1.7	70	107.5	103	-4.4	10	30	46.1	48	4.2	6
40	153.6	170	10.7	60	92.2	103	10.5	10	40	61.4	67	9.0	9
40	153.6	152	-1.0	50	76.8	71	-8.2	9	50	76.8	81	5.5	10
50	192	185	-3.6	70	134.4	131	-2.6	11	30	57.6	54	-6.3	6
50	192	212	10.4	60	115.2	131	12.1	11	40	76.8	81	5.5	8
50	192	198	3.1	50	96.0	103	6.8	10	50	96.0	95	-1.0	9

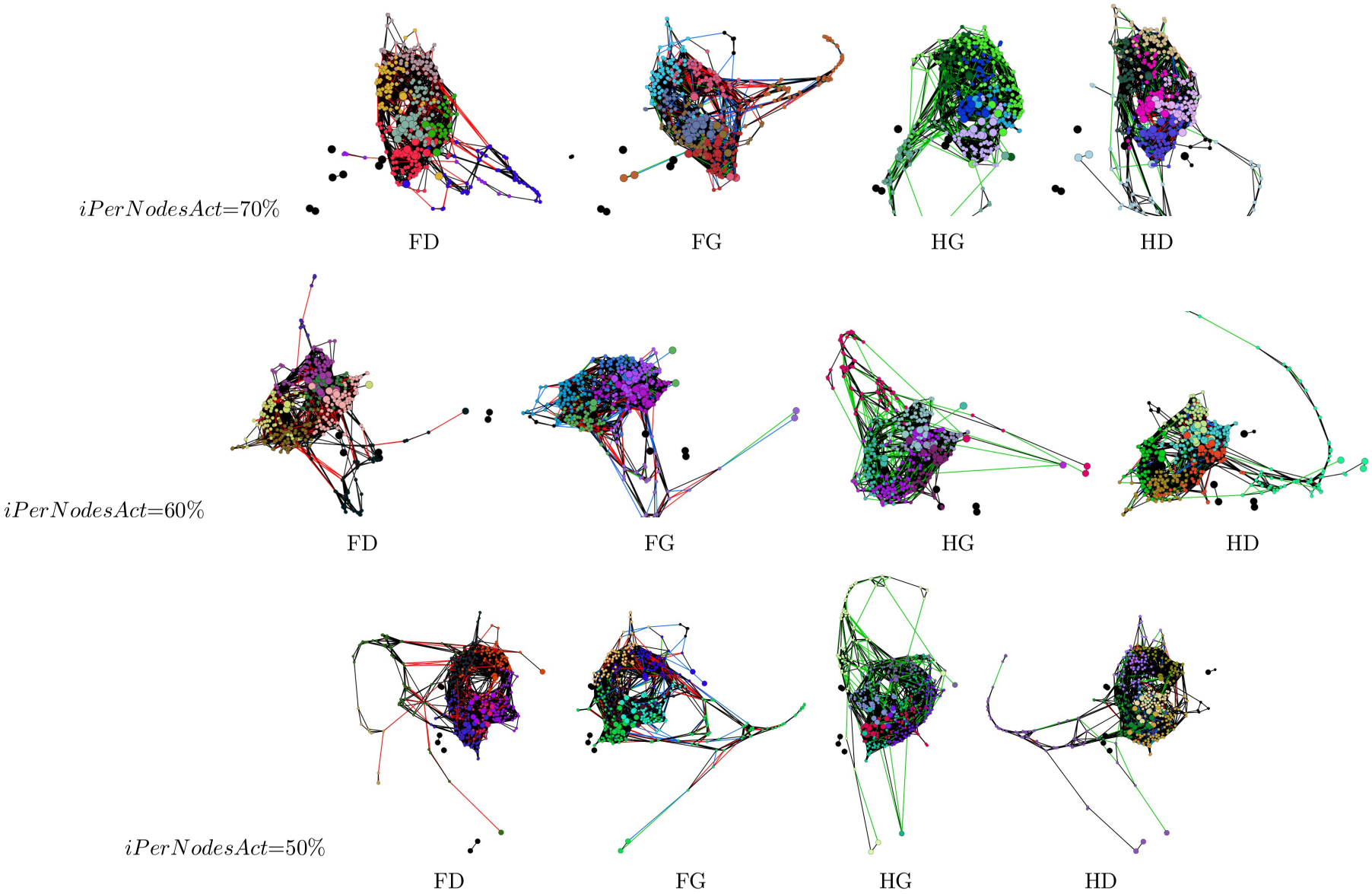
Layout LàC obtenu pour les quatre états {FD,FG,HG,HD} pour un pourcentage de nœuds fixes $iPerFixedNodes = 10\%$.



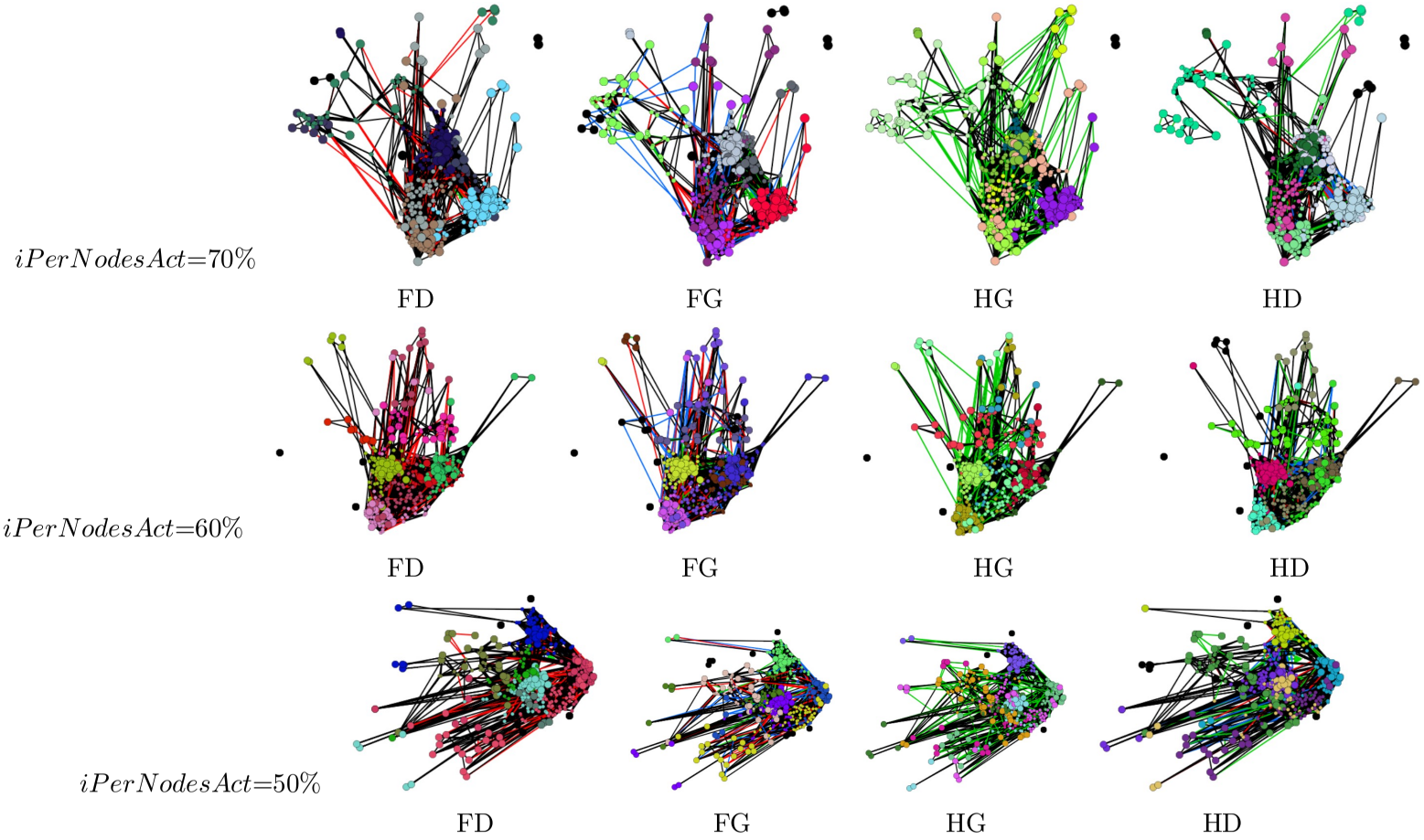
Layout LàC obtenu pour les quatre états {FD,FG,HG,HD} pour un pourcentage de nœuds fixes $iPerFixedNodes$ = 20%.



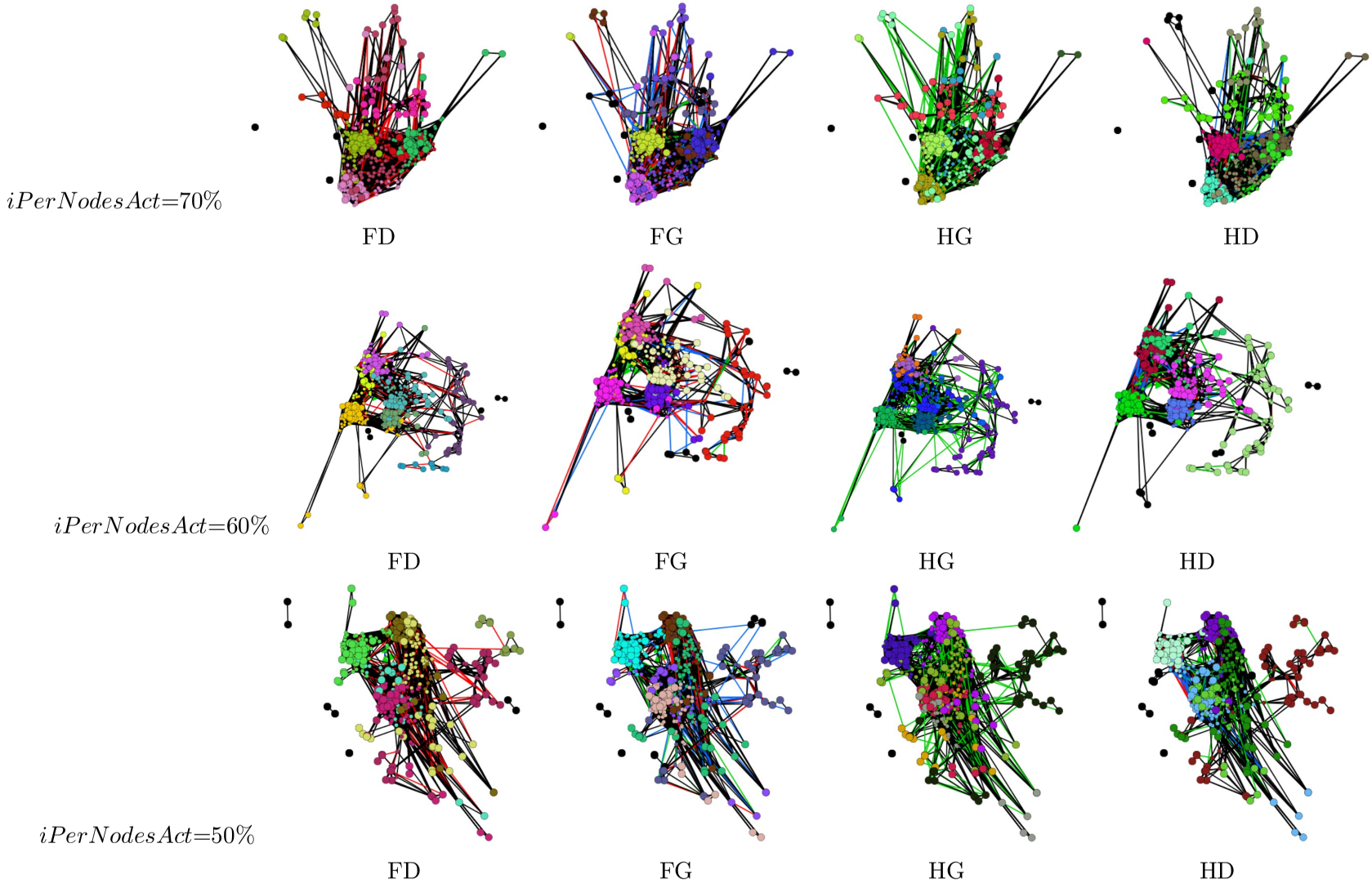
Layout LàC obtenu pour les quatre états {FD,FG,HG,HD} pour un pourcentage de nœuds fixes $iPerFixedNodes = 30\%$.



Layout LàC obtenu pour les quatre états {FD,FG,HG,HD} pour un pourcentage de nœuds fixes $iPerFixedNodes = 40\%$.



Layout LàC obtenu pour les quatre états {FD,FG,HG,HD} pour un pourcentage de nœuds fixes $iPerFixedNodes = 50\%$.

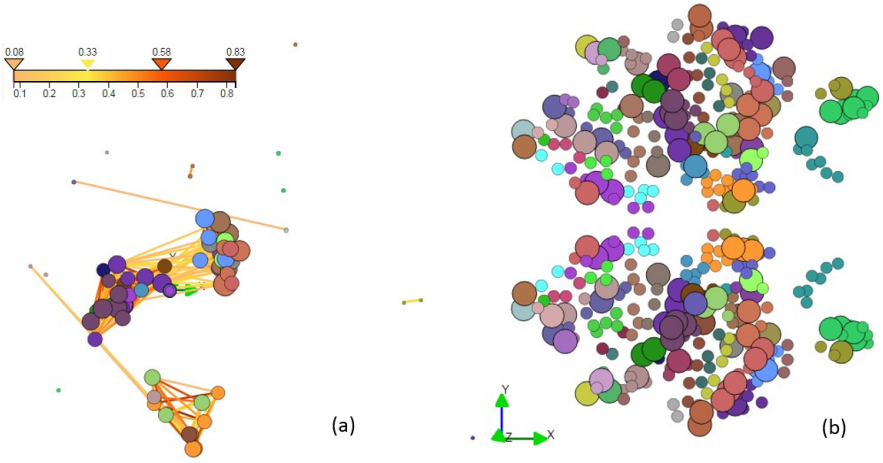


Résultats pour le jeu de paramètres choisi

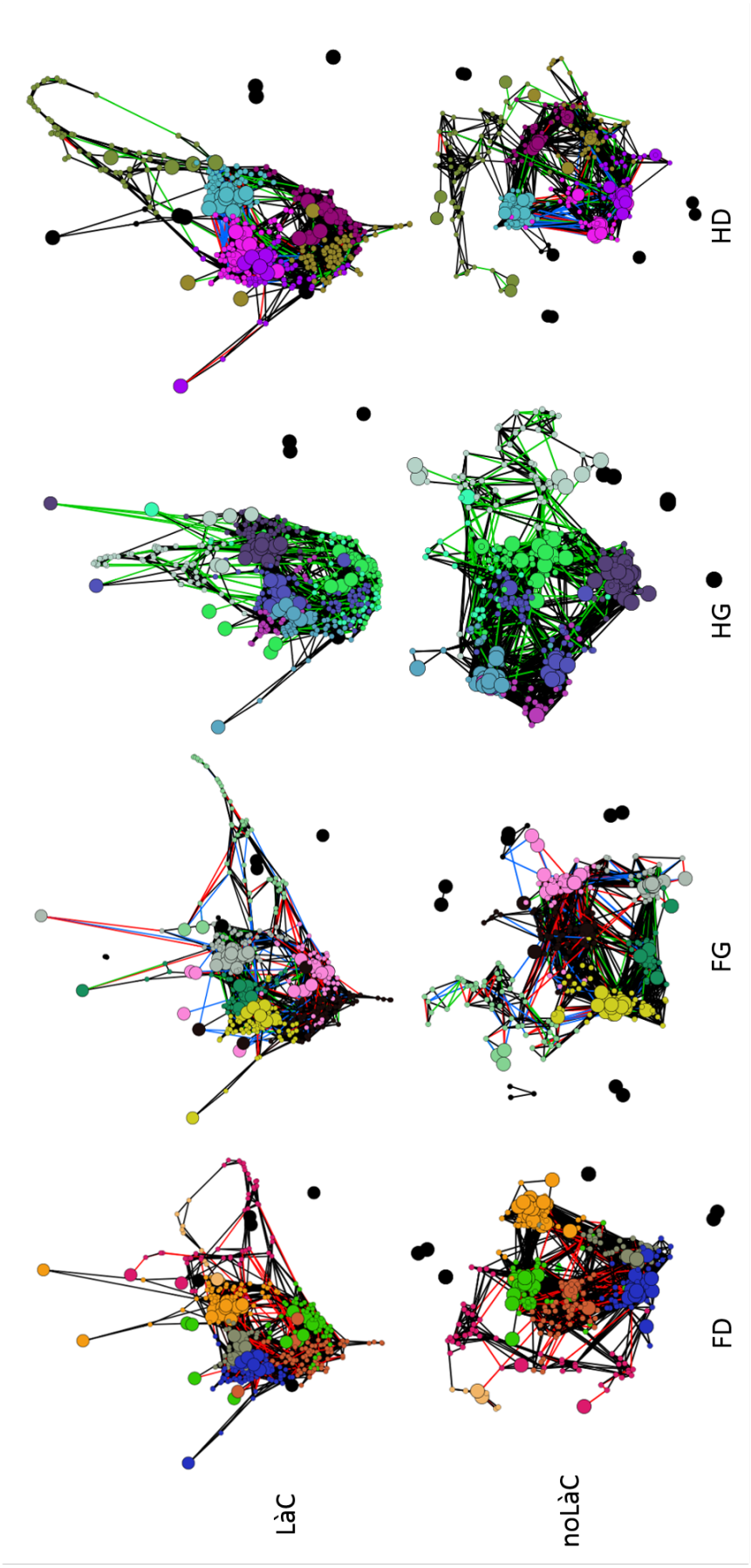
A la comparaison visuelle des résultats, nous retenons le jeu de paramètres suivant :

Paramètre	Valeur
$iThresholdEdges$	95%
$iPerFixedNodes$	20%
$iPerNodesAct$	70%
$iTheshold Act$	5.0
$iThresholdInac$	0.5
$iThresholdCc$	0.3
$iThresholdSdCc$	0.2

Graphe de synthèse obtenu sur les dimensions genre et latéralité : (a) LàC des nœuds constants, (b) identification des nœuds constants (taille=grand) sur le layout anatomique.



La figure ci-dessous montre le résultat du LàC pour les quatre classes de genre et latéralité, et donne à titre de comparaison le layout OpenORD sans fixation de nœuds. Les couleurs des nœuds correspondent à une colorisation aléatoire du clustering.


LàC
noLàC
HD
HG
FG
FD