INFORSID 2018

Actes du 36ème congrès INFORSID

Construire les Systèmes d'Information pour la Transformation des Organisations à l'ère de l'Innovation Numérique

Nantes, 28 au 31 mai 2018

Editeurs Scientifiques : Isabelle Comyn-Wattiau, Dalila Tamzalit

PARTENAIRES INSTITUTIONNELS

PARTENAIRES PRIVÉS

PARTENAIRES DE SOUTIEN

L'association INFORSID

Siège Social : 44, Chemin de la Caille - 31750 Escalquens

Web : http://inforsid.irit.fr/

INFORSID est une association régie par la loi de 1901 qui rassemble les chercheurs en informatique des organisations et systèmes d'information et qui a pour objectif de promouvoir les recherches effectuées dans ces domaines en faisant intervenir le plus largement possible les utilisateurs et les industriels. INFORSID centre son activité sur un ensemble de colloques et de séminaires périodiques au cours desquels le point est fait sur l'état des recherches en matière de système d'information et une orientation est donnée pour leur prolongement.

Composition du bureau

Présidente : Régine LALEAU, LACL, Université Paris-Est Créteil, IUT Sénart-Fontainebleau
Vice-président : Franck RAVAT, IRIT, Université Toulouse
Trésorier : Christian SALLABERRY, LIUPPA, Université de Pau et des Pays de l'Adour, IUT de Bayonne
Secrétaire : Agnès FRONT, LIG, Université Grenoble Alpes
Chargée de communication : Cécile FAVRE, Laboratoire ERIC, Université de Lyon 2.

Présidents d'honneur

Jean-Bernard CRAMPES (Toulouse)
Gilles ZURFLUH (Toulouse)
André FLORY (Lyon)
Claude CHRISMENT (Toulouse)
Michel SCHNEIDER (Clermont-Ferrand)
Corine CAUVET (Aix-Marseille)
Chantal SOULE-DUPUY (Toulouse)
Dominique RIEU (Grenoble)

PRÉFACE

A l'heure où les organisations, et plus généralement la société, vivent de grandes transformations largement dues aux nouvelles technologies, les ***systèmes d'information*** (SI) sont le socle sur lequel s'appuient tous leurs ***processus***. Les entreprises les plus dynamiques s'appuient sur des SI de plus en plus multiformes au sein desquels les ***réseaux sociaux*** jouent un rôle bidirectionnel pour mieux comprendre et capter les besoins des clients et diffuser des informations instantanément et tous azimuts. Néanmoins, ces opportunités supposent une capacité à capter, stocker et analyser toutes ces informations. La prise de ***décision*** doit être de plus en plus rapide en dépit de la complexité du contexte. La créativité repose ainsi sur une ***connaissance*** fine et multi-facettes de ce contexte. Saisir ces opportunités est incontournable pour la survie des entreprises. Toutefois, le déploiement de ces nouveaux SI génère aussi des risques juridiques, sociaux, environnementaux et financiers, rendant la ***sécurité*** des SI prégnante dans les préoccupations des directions des systèmes d'information (DSI). Les ***données massives*** que les SI doivent engranger sont elles aussi un sujet brûlant d'actualité et empreint de complexité.

Depuis 1982, le congrès annuel ***INFORSID*** (INFormatique des ORganisations et Systèmes d'Information et de Décision) constitue le lieu d'échange privilégié entre chercheurs et praticiens pour identifier et explorer les problématiques, les opportunités et les solutions que les SI apportent ou subissent. C'est aussi l'occasion de partager et de diffuser les expériences de mise en œuvre des ***méthodes, modèles, outils et solutions*** liés aux nouvelles technologies.

Le congrès INFORSID (INFormatique des ORganisations et Systèmes d'Information et de Décision) tient sa 36e édition en 2018. INFORSID rassemble, chaque année, chercheurs et professionnels autour de l'ensemble des problématiques d'ingénierie et de gouvernance des systèmes d'information, de gestion des données, de leur manipulation et de leur exploitation. En 2018, INFORSID est organisé en parallèle avec la conférence internationale RCIS (IEEE 12th International Conference on Research Challenges in Information Science).

Trois conférenciers invités, nous ont fait l'honneur d'accepter notre invitation. Il s'agit des Professeurs Yves Pigneur (Université de Lausanne, Suisse), Sudha Ram (Université de l'Arizona, Etats-Unis) et de Guillaume Tardiveau (Orange Labs Research, France).

Le programme comprend aussi différents ateliers dédiés à des problématiques spécifiques : Data Intelligence ; Evolution, Variabilité et Adaptabilité des Systèmes d'Information (EVA) ; Systèmes d'Information pour les Humanités Numériques (SIHN) ; Variété des Données SHS (Sciences Humaines et Sociales) ; Systèmes d'Information et de Décision et Démocratie ; Sécurité des Systèmes d'Information - Technologies et Personnes ; Evolution des Systèmes d'Information dans le contexte de l'Industrie 4.0.

Cette année, le congrès INFORSID a reçu 27 soumissions d'articles couvrant un grand nombre des problématiques liées à l'ingénierie des systèmes d'information. Chacun des articles soumis a été évalué par quatre membres du Comité de Programme. Une réunion plénière du Conseil du Comité de Programme a permis de sélectionner 10 articles longs et 3 articles courts pour présentation lors du congrès. A cela s'ajoutent trois résumés en français pour les trois articles qui ont été soumis conjointement aux deux conférences RCIS et INFORSID. La version

intégrale de l'article est publiée dans les actes de RCIS et ces actes d'INFORSID en proposent un résumé long en français.

J'ai le plaisir de remercier les membres du bureau de l'association INFORSID, sous la présidence de Régine Laleau, qui m'ont confié l'organisation scientifique du congrès. Je remercie également toutes celles et ceux qui ont permis la réalisation de cet événement : 1) les auteurs de tous les articles soumis qui nous ont fait confiance pour l'évaluation de leur recherche, 2) les conférenciers qui sont venus présenter les articles retenus, 3) les membres du comité de programme qui ont suscité des articles autour d'eux et, avec les relecteurs additionnels, ont lu avec soin et critiqué de façon constructive toutes les soumissions, 4) les membres du conseil de comité de programme qui ont, avec sagesse, participé à la décision pour chacun des articles soumis, 5) les conférenciers invités qui sont venus nous faire part de leurs passionnants sujets de recherche, 6) les porteurs des ateliers qui, sous l'impulsion de Jacky Akoka, ont défini un contenu, dynamisé le travail des contributeurs pour générer des échanges autour de sujets novateurs, 7) l'équipe d'organisation qui, pilotée par Dalila Tamzalit, a fait de ces journées une rencontre agréable et enrichissante, et 8) les concepteurs de l'outil easychair qui nous a grandement facilité la gestion du programme.

Isabelle Comyn-Wattiau
Présidente du comité de Programme INFORSID 2018
Mai 2018

COMITÉS INFORSID 2018

La réussite d'un congrès est le résultat d'un travail d'équipe. Qu'à nouveau tous ceux qui sont mentionnés ici et celles et ceux qui ont pu être omis en soient remerciés chaleureusement.

Le comité de la 36e édition d'INFORSID est composé par les responsables de l'organisation ainsi que les membres du comité de programme et les membres du conseil du comité de programme.

Présidence du Comité d'Organisation

Alain Bernard, Ecole Centrale Nantes, LS2N - équipe IS3P
Dalila Tamzalit, Université de Nantes, LS2N - équipe AeLoS

Comité d'organisation INFORSID 2018

Christian Attiogbe, Université de Nantes, LS2N - équipe AeLoS
Farouk Belkadi, Centrale Nantes, LS2N - équipe IS3P
Patricia Brière, Centrale Nantes, LS2N
Hugo Brunelière, Institut Mines Telecom Atlantique, LS2N - équipe Atlanmod
Karine Cantèle, CNRS, LS2N
Olivier Cardin, Université de Nantes, LS2N - équipe PSI
Laurence Drant, CNRS, LS2N
Sophie Girault, CNRS, LS2N
Pascale Kuntz, Université de Nantes, LS2N - équipe DuKe
Eric Languenou, Université de Nantes, LS2N - équipe DuKe
Jean-Marie Mottu, Université de Nantes, LS2N - équipe AeLoS
Jérôme Rocheteau, Institut Catholique d'Arts et Métiers, LS2N - équipe AeLoS
Séverine Rubin, Université de Nantes, LS2N
Patricia Serrano Alvarado, Université de Nantes, LS2N - équipe GDD
Gilles Simonin, Institut Mines Telecom Atlantique, LS2N - équipe TASC
Gerson Sunye, Université de Nantes, LS2N - équipe Atlanmod
Sandrine Thénot, Université de Nantes, LS2N

Conseil du Comité de Programme

Mireille Blay-Fornarino, Université Côte d'Azur, I3S, CNRS UMR 7271
Isabelle Comyn-Wattiau, ESSEC Business School
Agnès Front, Université de Grenoble Alpes, LIG
Régine Laleau, Université Paris-Est Créteil, LACL
Bénédicte Le Grand, Université Paris 1 Panthéon Sorbonne, CRI
Christian Sallaberry, LIUPPA, Université de Pau et des Pays de l'Adour, IUT de Bayonne
Dalila Tamzalit, Université de Nantes, LS2N

Présidence du comité de programme

Isabelle Comyn-Wattiau, ESSEC Business School

Présidence des ateliers

Jacky Akoka, CEDRIC-CNAM et IMT-TEM

Comité de programme

Adeel Ahmad, Laboratoire d'Informatique Signal et Images de la Côte d'Opale (LISIC)
Rachid Ahmed-Ouamer, LARI, Université de Mouloud Mammeri, Tizi-Ouzou, Algérie
Jacky Akoka, CEDRIC-CNAM & IMT-TEM
Pierre-Emmanuel Arduin, PSL, Université Paris-Dauphine, Laboratoire DRM (UMR CNRS 7088)
Said Assar, Institut Mines-Telecom
Djamal Benslimane, Université Lyon 1
Kamel Boukhalfa, Université des Sciences et de la Technologie Houari Boumediene, Algérie
Guillaume Cabanac, IRIT - Université Paul Sabatier, Toulouse 3
Sylvie Calabretto, LIRIS CNRS UMR5205 - INSA Lyon
Marie-FrancoiseCanut, IRIT, Toulouse
Corine Cauvet, Université d'Aix-Marseille
Jean-Pierre Chevallet, Université Grenoble Alpes
Célia Da Costa Pereira, Université Nice Sophia Anipolis
Thierry Delot, INRIA Lille Nord Europe & Université de Valenciennes, LAMIH
Chabane Djeraba LIFL
Eric Dubois, Luxembourg Institute of Science and Technology
Rim Faiz, IHEC, Université de Carthage, Tunisie
Agnès Front, LIG - SIGMA – Université de Grenoble
Mohand-Said Hacid, Université de Lyon 1
Patrick Heymans, Université de Namur, Belgique
Stéphane Jean,LISI/ENSMA et Université dePoitiers
Zoubida Kedad, Université de Versailles
Régine Laleau, Université Paris Est Créteil
Ilham Nadira Lammari, CEDRIC-CNAM
Bénédicte Le Grand, Université Paris 1 Panthéon - Sorbonne
Philippe Lopisteguy, LIUPPA - IUT Bayonne
Nadine Mandran, Université de Grenoble
Oscar Pastor Lopez, Universitat Politècnica de València, Espagne
Yves Pigneur, HEC - Université de Lausanne, Suisse
Olivier Pivert, IRISA-ENSSAT
Franck Ravat, IRIT, Université de Toulouse
Philippe Roose, LIUPPA
Malika Smail-Tabbone, Université de Lorraine
Dalila Tamzalit, Université de Nantes, LS2N - CNRS UMR 6004
Maguelonne Teisseire, Irstea - UMR Tetis
Olivier Teste, IRIT, Toulouse
Cassia Trojahn, UT2J & IRIT
Robert Viseur, CETIC, Université de Mons, Belgique
Amghar Youssef, INSA Lyon
Corinne Amel Zayani, MIRACL, Université de Sfax, Tunisie

Relecteurs additionnels

Patrick Etcheverry, Université de Pau et des Pays de l'Adour, Anglet
Christophe Marquesuzaà, Université de Pau et des Pays de l'Adour, Anglet
Nicolas Mayer, Luxembourg Institute of Science and Technology
Gilles Perrouin, Université de Namur, Belgique
Khouloud Salamah, Université de Pau et des Pays de l'Adour, Anglet
Jiefu Song, IRIT, Toulouse
Jason Vallet, Université de Bordeaux, CNRS UMR5800 LaBRI

Porteurs d'ateliers

Jacky Akoka, CEDRIC-CNAM et IMT-TEM, Paris
Pierre-Emmanuel Arduin, PSL, Université Paris-Dauphine, Laboratoire DRM (UMR CNRS 7088)
Farouk Belkadi, Ecole Centrale de Nantes – LS2N de Nantes
Fadila Bentayeb, Université Lumière Lyon 2, Laboratoire Eric
Mireille Blay-Fornarino, I3S, Nice
Raphaëlle Bour, Université de Toulouse 1 Capitole
Omar Boussaid, Université Lumière Lyon 2, Laboratoire Eric
Jérôme Darmont, Laboratoire ERIC, Université de Lyon
Cédric du Mouza, CEDRIC-CNAM, Paris
Sophie Ebersold, IRIT, Toulouse
Nathalia Grabar, CNRS, Lille et Orsay
Stéphane Lamassé, Laboratoire LAMOP, Université de Paris 1
Kathia Marçal de Oliveira, Université de Valenciennes
Néjib Moalla, Université Lumière Lyon 2 – Laboratoire DISP
Maryse Salles, Université de Toulouse 1 Capitole
Dalila Tamzalit, LS2N, Nantes
Olivier Teste, IRIT, Toulouse
Sylvain Vauttier, LIRMM, Montpellier

TABLE DES MATIÈRES

Conférences invitées

Systèmes d'information dédiés

Modèles et systèmes d'information

Session commune avec RCIS

Bases de données

Ontologies et contexte

Conférences invitées

Designing Business Tools for the Future - The Contribution of Information Systems

Yves Pigneur

Faculty of Business and Economics
University of Lausanne
Lausanne
Switzerland
Yves.pigneur@unil.ch

ABSTRACT. Based on his work with Alex Osterwalder on business models, Yves Pigneur will suggest that research in business innovation could be improved and enlightened by design science research in Information Systems. He will highlight three areas in which research in IS could inform the development of new business tools. The first area concerns the identification, formalization, and visualization of core constructs and models of interest related to business innovation and co-design of the "Enterprise of the future". The second area corresponds to the exploration of how design thinking techniques might contribute to improving the design and test of alternatives in business creation. The third area addresses the research in computer-aided design assisting the process of designing business objects.

KEYWORDS: Information systems, business model, design science research.

RESUME. En se fondant sur son travail avec Alex Osterwalder sur les modèles d'affaires, Yves Pigneur suggère que la recherche en innovation des affaires pourrait être améliorée et éclairée par la recherche en sciences de conception de systèmes d'information. Il met en avant trois domaines dans lesquels la recherche en systèmes d'information pourrait nourrir en information le développement de nouveaux outils pour les affaires. Le premier domaine concerne l'identification, la formalisation et la visualisation des construits principaux et des modèles d'intérêt liés à l'innovation dans les affaires et à la co-conception de l'« Entreprise du Futur ». Le second domaine correspond à l'exploration de la façon dont les techniques de « design thinking » pourraient contribuer à améliorer la conception et le test d'alternatives en création d'affaires. Le troisième domaine traite de la recherche en conception assistée par ordinateur pour le processus de conception d'objets pour les affaires.

MOTS-CLES : Systèmes d'information, modèle d'affaires, recherche en sciences de conception.

Leveraging Big Data and Analytics for Creating a Smarter and Better World

Sudha Ram

Eller College of Management
University of Arizona
Tucson
USA
ram@eller.arizona.edu

ABSTRACT. The phenomenal growth of social media, mobile applications, sensor based technologies and the Internet of Things is generating a flood of "Big Data" and disrupting our world in many ways. In this talk I will examine the paradigm shift caused by Big data and discuss how Analytics and Data science can be used to harness its power and create a smarter world. Using examples from health care, smart cities, education, and business in general, I will highlight challenges and opportunities for extracting value from Big Data to develop an enterprise of the future.

KEYWORDS: Big Data, Analytics, Data Science, Enterprise of the future.

RESUME. La croissance phénoménale des réseaux, des applications mobiles, des technologies à base de capteurs et de l'Internet des Objets génère un déluge de données massives et désorganise notre monde de multiples façons. Dans cet exposé, j'examine le changement de paradigme induit par les données massives (« Big Data ») et discute comment l'analytique et la science des données peuvent être utilisés pour exploiter leur puissance et créer un monde plus intelligent. En utilisant des exemples dans le domaine de la santé, des villes intelligentes, de l'éducation et des affaires en général, je soulignerai les défis et les opportunités d'extraction de valeur à partir des données massives pour développer l'entreprise du futur.

MOTS-CLES : Données massives, Analytique, Science des Données, Entreprise du futur.

Business Ecosystem in 2025

Guillaume Tardiveau

Orange Labs Research
France

ABSTRACT. Over the last decades, companies have used data as a substitute to paper and files. Lately internet, which is a digital native by all means, seems to be moving in the opposite direction by emerging in the physical world of objects. What should we expect from this encounter, and from the transformations that it will trigger on the way enterprises work? Anticipate the evolutions of enterprises on a 10-15 years timeframe is not easy, considering their diversity and the changing nature of their environment. This talk proposes to start from the Porter model and explore the relations between enterprises and their ecosystem (internal/external) as well as the way they work.

KEYWORDS: Internet, transformation, ecosystem, Porter model.

RESUME. Pendant les dernières décennies, les entreprises ont utilisé les données comme substitut au papier et aux fichiers. Plus récemment, Internet, qui est natif du numérique de toute évidence, semble se transformer dans le sens opposé en émergeant dans le monde des objets physiques. Que pouvons-nous attendre de cette rencontre et des transformations qu'elle va déclencher dans la façon dont les entreprises travaillent ? Anticiper les évolutions des entreprises sur une échelle de 10 à 15 ans n'est pas facile si l'on considère leur diversité et la nature changeante de leur environnement. Cette conférence propose de partir du modèle de Porter et d'explorer les relations entre les entreprises et leur écosystème (interne/externe) ainsi que la façon dont elles travaillent.

MOTS-CLES : Internet, transformation, écosystème, modèle de Porter.

Sessions d’articles sélectionnés

Systèmes d’information dédiés

Gouvernance des projets open source : le cas du logiciel Claroline

Robert Viseur[1], Amel Charleux[2]

1. *Université de Mons*
   Faculté Warocqué d'Economie et de Gestion
   17, Place Warocqué, 7000 Mons, Belgique
   robert.viseur@umons.ac.be

2. *Université de Montpellier*
   Montpellier Recherche en Management (MRM)
   Espace Richter (Bâtiment B), Rue Vendémiaire, CS19519, 34960 Montpellier
   amel.charleux@umontpellier.fr

RESUME. *Claroline est un projet de Learning Management System open source initié en Belgique par l'Université Catholique de Louvain. Plusieurs projets open source en sont, directement ou indirectement, dérivés (Dokeos, Chamilo, Claroline Connect). Compte tenu de sa diffusion, des opportunités de réalisation d'une étude longitudinale complète, de l'évolution de sa gouvernance, de ses forks et de sa résilience, Claroline présente un terrain de recherche idéal pour comprendre les dynamiques communautaires dans les communautés open source ainsi que les modalités de cohabitation avec des éditeurs publics ou privés. Dans cet article, nous proposons les résultats préliminaires d'une étude de cas basée sur des entretiens semi-directifs portant sur les modalités de gouvernance et de changement de modèle d'affaires au sein d'un écosystème open source. Nous montrons en particulier comment les choix successifs de gouvernance peuvent conduire à des mouvements de reconfiguration des communautés.*

ABSTRACT. *Claroline is an open source Learning Management System project initiated in Belgium by the Catholic University of Louvain. Several open source projects are, directly or indirectly, derived (Dokeos, Chamilo, Claroline Connect). Through its dissemination, the opportunities for conducting a comprehensive longitudinal study, the evolution of its governance, its forks and its resilience, Claroline presents an ideal research ground for understanding the community dynamics in open source communities as well as the modalities of cohabitation with public or private organizations. In this article, we propose the preliminary results of a case study based on semi-structured interviews about governance and business model changes in an open source ecosystem. In particular, we show how successive choices of governance can lead to community reconfiguration movements.*

MOTS-CLES : *open source, gouvernance, communauté, fork, innovation*.

KEYWORDS: *open source, governance, community, fork, innovation*.

1. Introduction

Les logiciels libres et open source ont connu une forte extension dès le milieu des années 90 avec l'essor du web, le développement des grandes communautés (p.ex. Apache) et l'implication croissante des entreprises (p.ex. Netscape ou IBM). Ils constituent par ailleurs un objet d'étude depuis près de 20 ans. Les recherches ont ainsi conduit à une bonne compréhension des modèles d'affaires open source (p.ex. dual licensing) et de leurs évolutions (p.ex. cloud computing et tendance « as a service »). La compréhension du fonctionnement des communautés open source reste par contre partielle malgré l'engouement récent pour les approches quantitatives (p.ex. data mining de répertoires de codes sources). Les conditions de la cohabitation entre les communautés et les éditeurs (qu'il s'agisse d'entités commerciales ou non commerciales) restent également mal connues, soient que les problèmes soulevés ne trouvent pas d'échos dans la littérature scientifique, soient que la traduction des solutions proposées s'avère difficile.

L'étude de cas porte sur le logiciel Claroline ainsi que sur les logiciels qui en ont été dérivés (Dokeos, Chamilo, Claroline Connect). Claroline est un logiciel open source de type Learning Management System (LMS) développé à partir de l'an 2000 au sein de l'Université Catholique de Louvain (Belgique), très vite rejointe par l'ECAM, l'Institut Supérieur Industriel associé à la HE Léonard de Vinci, puis repris par le Consortium Claroline à partir de 2007. Cet écosystème logiciel s'avère intéressant à plus d'un titre, en particulier : (1) l'accès aisé au terrain et l'opportunité de réalisation d'une étude longitudinale couvrant la vie du projet (depuis sa création), (2) la transformation progressive de la gouvernance du projet initial et (3) la survenance de plusieurs forks et l'éclatement de la communauté en plusieurs sous-communautés concurrentes.

Nous présentons dans cet article les résultats préliminaires de cette étude de cas. Nous chercherons en particulier à répondre à la question suivante : "*Comment les communautés open source réagissent-elles aux choix et aux changements de gouvernance* ?".

Notre article est décomposé en cinq sections. Dans une première section, nous procéderons à un état de l'art sur la gouvernance des projets open source, leurs facteurs de succès ainsi que sur les modèles d'affaires. Dans une seconde, nous présenterons la méthodologie utilisée pour l'étude de cas. Dans une troisième section, nous présentons l'étude de cas proprement dites ainsi que ses résultats. Nous les complétons ensuite par les modalités d'organisation et les résultats d'un atelier créatif. Dans une quatrième section, nous discutons les résultats. Dans une cinquième et dernière section, nous concluons par une synthèse des résultats et par les futures perspectives de recherche.

3. Etat de l'art

L'open source est aujourd'hui une réalité quotidienne dans le secteur informatique. Selon une étude réalisée en 2015 pour le Conseil National du Logiciel

Libre, le secteur du logiciel libre et open source représenterait ainsi en France un chiffre d'affaires cumulé de 4,1 milliards d'euros (2015) pour environ environ 50000 emplois (CNLL, 2015). Parmi les avantages cités par la centaine d'entreprises ayant participé à l'étude, citons un "*facteur d'innovation sans équivalent avec les logiciels propriétaires*" loué par 75% des répondants.

L'investissement croissant des entreprises dans les projets open source a notamment été analysée par Fitzgerald (2006). Quant à la question de l'existence ou non de modèles d'affaires open source, elle a été posée voici quelques années (Vasquez Bronfman et Miralles, 2007). Il ressortait de cette interrogation que la majorité des prestataires démarraient leur activité en tant que fournisseurs de services et sans réelle réflexion préalable sur le modèle d'affaires. Ce constat n'a rien de surprenant. Teece (2010) estime ainsi que "*le bon modèle d'affaires est rarement apparent tout de suite dans les industries émergentes*". Le modèle d'affaires décrit les modalités de création, de distribution et de capture de valeur. Il peut être générique, partagé par plusieurs entreprises parfois en concurrence (Teece, 2010). Dans le cas open source, certains modèles d'affaires plus originaux se distinguent, comme le principe de la double licence (dual licensing) étudié par Välimäki (2003). Viseur (2013a) présente une synthèse de ces modèles d'affaires et met en évidence le caractère non figé de ces modèles, pouvant conduire à des conflits avec la communauté rassemblée autour du logiciel, par exemple sous la forme d'un refus de changement de licence (Viseur et Robles, 2015). Ils évoluent ainsi avec la maturité de l'entreprise (p.ex. transformation vers un rôle d'éditeurs et structuration d'un réseau d'intégrateurs) ou des évolutions technologiques (p.ex. popularisation du cloud computing et succès des offres de type Software as a Service).

Les communautés open source sont susceptibles d'attirer des acteurs fort différents : passionnés, informaticiens indépendants, informaticiens salariés (secteurs privé ou public), chercheurs,... Lakhani et al. (2002) ont très tôt clarifié les motivations individuelles (contributeurs). Ils ont pu constater que les développeurs étaient souvent attirés vers l'open source par pragmatisme (top 3 : stimulation intellectuelle, amélioration de l'expertise et fonctionnalités), estimant pour un tiers d'entre eux que le code doit être open source, attendant des responsables capables de dialoguer, de fournir une vision et de produire ou intégrer du code source. Si les contributeurs ne sont pas des "croisés" (battre Microsoft apparaissait comme une motivation pour un peu plus d'un contributeur sur dix seulement), ils sont donc attachés à certaines valeurs garanties par les licences des logiciels open source. L'entreprise (p.ex. éditeur) doit donc composer avec cette culture ou faire le choix d'attirer des contributeurs pour lesquels la liberté du code est moins importante (p.ex. développeurs rémunérés). La participation de contributeurs rémunérés permettrait par ailleurs d'accroître l'activité sur les projets sans dégrader la qualité du code source (Roberts et al., 2006). L'open source apparaît dès lors comme un environnement complexe réconciliant motivations intrinsèques et extrinsèques. Les motivations des entreprises peuvent être de natures sociales, économiques ou technologiques, avec une prédominance des deux dernières (Bonaccorsi et Rossi, 2006 ; Dahlander et Magnusson, 2005 ; Feller et Fitzgerald, 2002). Il s'agit en particulier des opportunités d'innovation pour les petites entreprises, de la collecte de

feedback (p.ex. rapports d'erreurs), de la fiabilité et la qualité des logiciels open source et de la plus grande indépendance vis-à-vis des grandes entreprises (p.ex. prix et licences). La conformité aux valeurs du mouvement du logiciel libre arrive en cinquième position. Au delà des tensions possibles autour de l'interprétation des normes sociales et des valeurs internes aux communautés, il semble donc exister une adhésion aux logiques d'ouverture et de collaboration propres aux logiciels libres et open source, tant chez les développeurs qu'au sein des entreprises.

Les modalités d'interactions entre l'organisation assurant l'édition du logiciel (p.ex. éditeur ou fondation) et la communauté qui en soutient le développement varient fortement d'un projet à l'autre (p.ex. MySQL ou Eclipse) et ne sont pas figées dans le temps (p.ex. Netscape / Mozilla). La gouvernance du projet, soit les conditions d'exercice du pouvoir au sein de la communauté ou de l'écosystème, fait l'objet d'études depuis une dizaine d'années sur le plan de la structuration au cours du temps (De Laat,2007) ou des dimensions de cette structuration (Markus, 2007). Sur cette base, Viseur (2016) propose quatre logiques de gouvernance open source : (1) la logique informelle (cas général des petits projets hébergés sur des dépôts publics sans règles formelles autres que la licence), (2) la logique commerciale (cas des éditeurs open source souhaitant conserver le contrôle du projet), (3) la logique communautaire (cas des projets de grande taille ou critiques pérennisés par une entité légale) et (4) la logique industrielle (cas des acteurs industriels poursuivant un objectif de mutualisation dans un cadre coopétitif).

Les tensions entre le responsable d'un projet (p.ex. éditeur) et la communauté qui le soutient peut conduire à une scission de la communauté. Pour les projets open source, ce processus est baptisé "fork". Au travers de l'analyse de 26 forks populaires, Viseur (2012, 2016) a étudié les conséquences d'un fork sur le projet initial ainsi que les motivations qui y conduisent. La cohabitation entre les deux projets, c'est-à-dire le projet original et son fork, représente la situation la plus fréquente. Les motivations apparaissent par contre très diverses : arrêt du développement (sous sa forme open source), objectifs techniques divergents, changement de licence, conflit autour de l'utilisation d'une marque, conflit autour de la gouvernance du projet (capacité d'influencer le projet au travers de sa feuille de route ou de ses contributions), les différences "culturelles" (hétérogénéité de la communauté) et la recherche d'innovation. Les motivations liées à la capacité à influencer le projet et aux objectifs techniques interviennent dans plus des trois quarts des forks étudiés. Sur le plan théorique, les forks dus à des divergences d'objectifs techniques ou fonctionnels pourraient être évités par le développement d'une architecture modulaire permettant la personnalisation ou la verticalisation sous la forme de distributions (Viseur, 2016). MacCormack et al. (2006) parlent d'architecture de participation. L'impact des logiques de gouvernance sur le risque de fork reste par contre incertain.

3. Méthodologie

L'étude de cas de l'écosystème du logiciel open source Claroline nous a conduit à travailler sur 4 logiciels distincts : Claroline (2001) et son héritier direct Claroline

Connect (2014), le fork Dokeos (2003) et son fork Chamilo (2010). Elle s'apparente à un cas unique et longitudinal (Thietart, 2007 ; Yin, 2009).

Dans une première phase, huit entretiens semi-directifs, généralement individuels, en face-à-face, ont été réalisés, en binôme, sur base d'un guide d'entretien standard, avec prise de note et enregistrement. Les personnes interviewées couvrent les responsables actuels des différents projets ainsi que les principaux initiateurs du projet Claroline. De cette manière, la contamination intragroupe a été limitée par l'accès à des sous-groupes suffisamment disjoints car issus du projet initial et de ses scissions successives (forks). L'arrêt des interviews a été dicté par l'atteinte d'une saturation en informations. L'entretien démarrait par une question générale permettant de rester non-directif. Des questions plus précises étaient ensuite utilisées pour relancer les échanges ou préciser un point. Une attention particulière a été accordée à la liberté de parole des interviewés, en particulier lorsque des divergences de vue apparaissaient (p.ex. conflits). Les entretiens doivent faire l'objet d'une retranscription et d'un codage. Ce travail n'a cependant pas été réalisé, les résultats préliminaires, validés par les chercheurs (regards croisés), se basant essentiellement sur les notes d'entretiens et les enregistrements (vérification).

Outre les entretiens, des sources primaires et secondaires ont été utilisées, en particulier pour établir la chronologie des différents projets (p.ex. site web des projets, présentations publiques et rapports professionnels). Le projet Claroline a fait l'objet d'une première étude de cas par questionnaire en 2006 (Viseur, 2007). La totalité du matériel de recherche (questionnaire, réponses au questionnaire, références, sitographie, notes, enregistrements, résultats préliminaires,...) fait l'objet d'un partage entre les chercheurs.

Dans une seconde phase, une première présentation de l'historique, des résultats préliminaires de l'étude ainsi que des questions soulevées a été assurée par les auteurs de l'étude à l'occasion de la conférence annuelle réunissant à Bruxelles les utilisateurs du logiciel (ACCU 2017). A cette occasion, un atelier créatif a également été réalisé. Baptisé "*Recharge ton ACCU*", il visait à identifier des pistes de redynamisation de la communauté gravitant autour de Claroline Connect. Cette phase permettait par ailleurs un élargissement de l'accès au terrain et la captation du point de vue des utilisateurs proches du projet. La posture adoptée s'apparente à une posture constructiviste transformative, où les savoirs construits par les chercheurs sont confrontés à la perception des acteurs de terrain (Giordano, 2003).

4. Etudes de cas

4.1. Historique des projets

L'histoire du projet Claroline / Claroline Connect peut être découpée en quatre grandes étapes.

Tableau 1. Historique du projet Claroline

Année	Evènement marquant
2000	Démarrage des développements qui donneront naissance à Claroline au sein de l'IPM (UCL).
2001	Publication de la première version de Claroline.
2003	Départ de Thomas De Praetere pour former le fork Dokeos et la société Dokeos.
2004	Obtention d'un financement public (projet WIST) par l'ECAM.
2007	Création du consortium Claroline (fondation).
2010	Fork Chamilo issu de Dokeos.
2010	Sortie de la note d'orientation de Claroline Connect.
2012	Premiers développements de Claroline Connect.
2014	Sortie de Claroline Connect (RC).
2015	Création de la société Formalibre.

L'ère des pionniers (2000 - 2003)

En l'an 2000, l'Université Catholique de Louvain (UCL) cherche à mettre en place des services en ligne dédiés à l'elearning. Le logiciel propriétaire WebCT est retenu et l'Institut de Pédagogie universitaire et des Multimédias (IPM), devenu Louvain Learning Lab (LLL) en 2015, chargé de l'accompagnement des enseignants. Cette solution se révèle coûteuse, lente et peu adaptée aux besoins. Une petite équipe (2 développeurs) se lance dans le développement d'une solution, basée sur des composants open source, centrée sur les usages. Cette solution, bientôt baptisée Claroline, est publiée en 2001. Elle rencontre un vif succès tant à l'intérieur qu'à l'extérieur de l'institution. Sur le plan théorique, ces pionniers peuvent être assimilés à des utilisateurs de pointe décidant d'innover par eux-mêmes pour disposer d'une solution adaptée à leur besoin (Franke & Von Hippel, 2003). Cette période se termine avec le départ de Thomas de Praetere, un des initiateurs du projet, caractérisé par un profil d'intrapreneur, désireux de créer une entreprise de services autour du logiciel Claroline. En conflit avec la structure en charge de l'accompagnement des spin-offs, il décide de forker le projet, le rebaptise et lance la société Dokeos.

La pérennisation et l'institutionnalisation (2004 - 2007)

L'énergie des pionniers laisse la place à une consolidation et une recherche de ses conditions de pérennisation (p.ex. financement). Cela se traduit, d'une part, par l'arrivée d'un nouveau partenaire, l'ECAM (HE Vinci), et le dépôt d'un projet WIST permettant le financement du projet, et, d'autre part, le renfort du projet par une équipe de professionnels de l'informatique. De manière à maintenir le contrôle de la feuille de route (roadmap) et la qualité des développements, la priorité est

donnée à des pôles de développeurs issus des institutions partenaires plutôt qu'à l'activité communautaire. Cette période prend fin avec l'arrivée à terme du WIST et les difficultés à trouver de nouveaux financements.

L'autonomisation et la quête d'une gouvernance (2008 - 2011)

Le projet s'autonomise progressivement des institutions qui l'ont créées. Formellement créé en 2007, le consortium Claroline fournit un cadre permettant de veiller aux intérêts du projet Claroline. Le logiciel Claroline est stabilisé. Les lignes directrices de sa nouvelle version, Claroline Connect, sont présentées en 2010. En parallèle, le projet Dokeos ferme progressivement son modèle d'affaires, évoluant vers un modèle de double licence peu ouvert à la communauté ; son fork Chamilo voit le jour en 2010 et se structure en une association ouverte à la communauté et aux contributeurs internationaux.

Le redéploiement local et international (depuis 2012)

A partir de 2012, le logiciel Claroline Connect est mis en développement. En 2014, le projet doit faire face à l'arrêt du soutien de l'Université Catholique de Louvain (UCL) et à sa migration vers le logiciel open source Moodle. Cette défection est compensée par l'arrivée de l'Université de Lyon I, via le service iCAP (Innovation Conception Accompagnement pour la Pédagogie), qui apporte l'expertise et les ressources associées au logiciel Spiral Connect[1]. En 2014 également, les premières versions utilisables de Claroline Connect sont présentées, avec une première version stable officiellement lancée fin mai 2015. A cette occasion, un prestataire privé, baptisé Formalibre, est également créé. Le projet Claroline Connect peut se redéployer et, en particulier, relancer sa communication vers les utilisateurs.

4.2. Gouvernance et forks

Le projet Claroline et ses dérivés présentent des gouvernances distinctes et reflètent les logiques de gouvernance identifiées par Viseur (2016).

Logique informelle

A sa naissance, Claroline suit une logique informelle. L'équipe de développement est autonome. Les règles sont définies par la licence (GPL).

Logique commerciale

L'éditeur du fork Dokeos évolue rapidement vers une logique commerciale. La communauté est perçue comme un frein en termes de rentabilité et de time-to-market. Le projet se referme progressivement jusqu'à proposer un modèle de double licence (sans réelle communauté associée au développement) puis un re-développement sous licence propriétaire à la suite du fork Chamilo.

Logique industrielle

[1] Le nom Claroline Connect est une contraction de Claroline et Spiral Connect.

Le fonctionnement du consortium Claroline confirme la préférence pour les pôles de développeurs, apportés par des institutions partenaires. L'accès au statut de Membre est ainsi conditionné à l'apport de ressources (p.ex. développeurs ou financements) sur le projet. Ce choix conduit, sans intention maligne, à écarter certaines franges de la communauté.

Logique communautaire

Le projet Chamilo récupère dès son fork une partie de la communauté d'utilisateurs et mise d'emblée sur une diffusion maximale du projet (p.ex. installation en 1 clic). Si le fonctionnement de l'association peut sembler similaire à celui du consortium Claroline, elle se révèle cependant ouverte aux membres actifs (méritocratie) sans condition d'apport de ressources.

Tableau 2. Claroline, la gouvernance et les forks

Gouvernance	Forks	Editeurs
Logique informelle	----- Claroline (2001) | | | |	
Logique commerciale	| Dokeos (2003) | |	Dokeos
Logique communautaire	| | | Chamilo (2010) |	Beeznest,...
Logique industrielle	| Claroline Connect (2014)	Formalibre (2015),...

Forks (Dokeos et Chamilo)

L'écosystème Claroline a été traversé par différents soubresauts. Le premier est provoqué par le fork Dokeos, issu de Claroline. Le second est Chamilo, issu de Dokeos. Il n'y a pas d'échanges de code entre variantes, malgré des tentatives de rapprochement entre les équipes de Dokeos, de Claroline Connect et de Chamilo.

Le fork Dokeos a été initié par Thomas de Praetere, un des créateurs du projet Claroline. Désireux de lancer une entreprise de services, il entre en conflit avec la structure en charge de l'accompagnement des spin-offs au sein de l'université à propos des conditions de valorisation de la marque Claroline (condition standard de prise de participation dans la spin-off contre un droit d'utilisation de la marque). Le fork Dokeos est accolé à l'entreprise éponyme.

Le fork Chamilo a été initié par Yannick Warnier, un temps partenaire de la société Dokeos en Amérique latine et fondateur de la société Beeznest. Ce fork est motivé par la fermeture progressive du projet Dokeos ; ses initiateurs souhaitent une

ouverture accrue vis-à-vis de la communauté. La création d'une association est voulue comme un moyen de garantir cette ouverture sur le long terme et de limiter le pouvoir des initiateurs du projet Chamilo.

4.3. Résistances au changement

Des résistances à différents changements ont par ailleurs été constatées.

1) Innover avec les utilisateurs (Claroline)

Les initiateurs du projet Claroline sont des utilisateurs de pointe confrontés à la politique de l'institution (sélection du logiciel propriétaire WebCT jugé insatisfaisant). Portés par les utilisateurs (enseignants), favorables à l'innovation centrée sur les utilisateurs plutôt qu'à une approche fonctionnelle, ils s'opposent à la structure soucieuse de faire respecter ses choix technologiques. Un parallèle pourrait être dressé avec le projet belge CommunesPlone, porté par des informaticiens communaux contre des choix ministériels.

2) Pérenniser le projet (Claroline)

Le projet se structure en vue d'assurer sa pérennisation. Les pionniers se sentent dépossédés du projet qu'ils ont créés. Il en résulte le fork du projet par un des fondateurs[2] puis le départ du second développeur.

3) Privatiser le projet (Dokeos)

Dokeos fait évoluer son modèle d'affaires. L'entreprise tente une privatisation partielle du projet par le passage à un modèle de double licence et l'ajout de modules propriétaires permettant de différencier les versions communautaires et commerciales. Il en résulte un second fork, baptisé Chamilo, rassemblant la communauté.

4) Fusionner deux variantes (Claroline Connect et Chamilo)

Le fork Chamilo et la validation de la feuille de route de Claroline Connect sont contemporains. Une fusion de Chamilo et de Claroline est dès lors tentée. Il s'agit d'un échec, dont les raisons mériteraient un approfondissement. Les différences d'approche qualité et d'architecture font partie des motifs (path dependency ?). Un parallèle pourrait être dressé avec Nokia et la fusion, tardivement réussie, de Maemo et Moblin au sein de Meemo, aujourd'hui devenu Sailfish OS, marquée par les difficultés d'homogénéisation des pratiques entre partenaires et communautés (Viseur, 2013b).

5) Rassembler la communauté (Claroline Connect)

[2] Dans un projet open source, la propriété prend différentes formes. Le partage du code source est régulé par la licence mais peut cohabiter avec la pleine propriété d'une marque. La nature open source permet ici à un employé de s'affranchir de son employeur suite à un désaccord sur les conditions d'exploitation du logiciel.

Le basculement sur la nouvelle version de Claroline, baptisée Claroline Connect, apportant différentes évolutions notamment en matière d'innovation pédagogique, s'est accompagné de ruptures radicales chez les institutions utilisatrices, sous la forme notamment de migrations vers le logiciel open source Moodle (p.ex. UCL). Les causes des résistances au changement à la migration vers Claroline Connect sont multiples (mauvaise communication, rapports de force internes aux organisations, temps d'attente de la nouvelle version,...).

6) Aligner les stratégies (Claroline Connect)

La création de Formalibre, du fait de sa capacité de production et de ses sources de financement (opérateurs privés), pourrait entraîner de nouvelles formes de résistances dues aux objectifs divergents entre l'entreprise privée (p.ex. priorité aux clients privés) et les institutions membres ou utilisatrices (p.ex. complexité d'installation de la solution actuelle et objectif d'innovation pédagogique).

4.4. Relations à la communauté

Ces résultats préliminaires apportent de nouveaux éléments, incluant de nouvelles interrogations, principalement relatives à trois thématiques : la gouvernance, la communication et l'animation de la communauté.

Gouvernance

La création du consortium Claroline (fondation internationale) a apporté plusieurs bénéfices pour le projet. Premièrement, le consortium permet la négociation des règles. Le fonctionnement du projet ne se fait plus en fonction d'un leader plus ou moins éclairé mais suivant des règles discutées et amendées suivant des procédures prédéfinies. Deuxièmement, le consortium permet de centraliser la gestion des ressources. Il permet la collecte et la mise à disposition des financements dans un sens décidé collectivement au travers du conseil d'administration (CA) et de son assemblée générale (AG) annuelle. Troisièmement, le consortium garantit la neutralité du projet. La fondation ne se confond pas avec ses membres. L'étiquette "catholique" associée à l'UCL peut ainsi amener des difficultés inattendues, que ce soit en local (p.ex. réminiscence des guerres scolaires en Belgique) ou à l'international (p.ex. états confessionnels non chrétiens). Quatrièmement, le consortium apporte davantage de stabilité au projet. La fondation garantit une pérennité face à des changements internes aux institutions membres ainsi qu'aux départs de personnalités importantes. De la sorte, elle réduit le risque perçu par les utilisateurs, notamment institutionnels. Cinquièmement, le consortium renforce l'attractivité du projet. L'existence d'une fondation, par laquelle le projet ne se confond pas sur le plan juridique avec une institution influente, rassure les partenaires potentiels qui hésitent ainsi moins à franchir le pas et à adopter la solution.

Communication

Le cas de Claroline / Claroline Connect, mais aussi d'autres projets open source étudiés, offrent le constat d'un manque d'efficacité de la communication autour du projet. Cela se traduit par la difficulté de communiquer de manière efficace vers les

différentes parties prenantes du projet : membres du consortium, utilisateurs institutionnels, développeurs tiers,... Dans le cas particulier de Claroline / Claroline Connect, nous posons le constat d'une connaissance très imparfaite de l'histoire du projet (et de ses forks) au sein des institutions d'enseignement (supérieur ou universitaire) mais aussi d'un manque de communication efficace dans le cadre de l'abandon du support de Claroline et de la migration vers Claroline Connect, conduisant à considérer la solution morte et enterrée. Ce dernier constat pourrait s'expliquer par les faibles ressources disponibles pour gérer la communication du projet mais aussi par la difficulté à diffuser le bon message au bon moment vers les très nombreuses parties-prenantes.

Animation

En matière d'animation, les interviews ont conduit à des interrogations sur les causes d'un manque de contributions externes sur le projet (excepté les nécessaires traductions). Est-ce dû à la priorité accordée dès le départ aux pôles de développeurs (collaboration entre institutions) plutôt qu'aux communautés de développeurs ? Est-ce dû à un manque d'animation quant à la possibilité de créer des extensions pour Claroline / Claroline Connect ? Est-ce dû à un manque d'animation du réseau de développeurs ou de prestataires de services développant sur le projet sans être membre du consortium ? Est-ce une stratégie délibérée qui permet de garder le contrôle sur les évolutions de la solution (roadmap) ?

4.5. Atelier "Recharge ton ACCU"

La participation à la conférence annuelle (ACCU 2017) a permis de confronter la compréhension de la situation suite aux entretiens à la perception de la communauté dans son ensemble, incluant des éléments jusqu'alors inconnus (p.ex. écoles secondaires). Les réactions ont notamment confirmé la préférence marquée, dès avant la création du consortium, pour les pôles de développeurs, conduisant à une absence de dynamisation de la communauté des développeurs.

A la suite de ces discussions, un atelier a été organisé avec une dizaine de participants selon le processus suivant. Dans un premier temps, un brainstorming a été réalisé sur base de la question suivante : *"Comment améliorer la vitalité de la communauté Claroline Connect ?*". Les participants, au nombre de dix environ, stimulés par des inducteurs visuels (mots projetés), étaient invités à proposer des améliorations en termes de communication et d'animation de la communauté. Cette séance a débouché sur quarante idées environ, regroupées en 12 propositions. Dans un second temps, ces 12 propositions ont été soumises à un vote secret (3 votes "pour", 3 votes "contre" et 1 "coup de coeur" par personne). Le caractère secret du vote permet d'éviter les effets de mimétisme et de plus facilement faire apparaître les divergences. Les idées polémiques peuvent ensuite être discutées.

Cette séance a permis l'émergence de deux possibles actions prioritaires : (1) la communication sur les éléments de différenciation de Claroline Connect et (2) le développement de la documentation et de l'entraide en ligne (p.ex. forums). Elle a également permis la mise en évidence de blocages existants en matière de représentation de certaines franges de la communauté (p.ex. utilisateurs issus de

l'enseignement secondaire), pouvant faire l'objet d'une analyse plus approfondie. En particulier, le manque de représentation (memberships) des contributeurs hors institutions membres du consortium pourrait expliquer le manque de participation (West et O'Mahony, 2008) ; quant au manque d'animation et d'attention à la diffusion du logiciel (p.ex. effectivité des procédures d'installation et promotion des outils de migration), il réduit tant les bénéfices directs (p.ex. contributions en code) qu'indirects (p.ex. augmentation du nombre d'utilisateurs).

5. Discussion

Ces résultats préliminaires apportent aussi des réponses quant à la question des réactions de la communauté aux choix et aux changements de gouvernance en lien avec l'évolution des modèles d'affaires.

Modèles d'affaires et logiques de gouvernance

S'ils présentent de fortes similarités sur le plan fonctionnel, Claroline et ses variantes présentent des spécificités sur le plan des modèles d'affaires.

Tableau 3. Modèles d'affaires associés aux projets

	Claroline	**Dokeos**	**Chamilo**	**Claroline Connect**
Capture de la valeur	Financements (partenaire et secteur public)	Projets commerciaux, double licence	Activités de services	Financements (partenaires), projets commerciaux (Formalibre)
Création de la valeur	Pôles de développeurs	Editeur	Mutualisation	Pôles de développeurs (incluant un éditeur)
Distribution de la valeur	Code source publié	Accès progressivement restreint	Accès au code source et installation simplifiée (p.ex. 1-clic)	Code source publié
Logique de gouvernance	Informelle, puis industrielle	Commerciale	Communautaire	Industrielle

Configuration des communautés

En cas de logique informelle, la communauté ne dispose pas de de garde-fous culturels ou réglementaires, excepté la licence du logiciel, qui fixe des droits et des devoirs fondamentaux. Cette configuration nous paraît propice à une exacerbation des conflits, conduisant à la survenance de forks (p.ex. Dokeos). En cas de logique communautaire, la gouvernance veille à réguler, dans un soucis de recherche d'équilibre, les divergences d'intérêts pouvant survenir entre membres de la communauté. Cette configuration nous paraît propice à un large rassemblement

d'utilisateurs et de développeurs (p.ex. Chamilo). Par contre, elle est plus difficilement conciliable avec les priorités d'organisations ayant une feuille de route nécessitant un contrôle minimum du projet (p.ex. contrainte forte de type time-to-market). Ce type d'organisation privilégiera (et s'orientera donc progressivement vers) une logique commerciale (p.ex. Dokeos) ou une logique industrielle (p.ex. Claroline). La conséquence d'un passage vers une logique commerciale ou industrielle est qu'elle tend à opérer un choix de segmentation (Table 3) parmi les partenaires et contributeurs, susceptible de conduire à un fork (p.ex. Chamilo) ou à une séparation progressive (p.ex. Claroline post-consortium). L'architecture du logiciel (p.ex. modularité) et l'animation de la communauté pourraient limiter cet effet d'éviction (p.ex. stimuler la création de modules et verticaliser sous la forme de distributions).

Tableau 4. Logique de gouvernance et configuration de la communauté

	Logique commerciale	**Logique industrielle**	**Logique communautaire**
Acteur(s) dominant(s)	Editeur	Grande(s) organisation(s)	Méritocratie égalitaire
Communauté	Partenaires (réseau structuré)	Grandes organisations (coopétition)	Petites organisations et utilisateurs individuels
Développeur(s) dominant(s)	Editeur et partenaires	Grandes organisations	Core team et développeurs individuels
Motivations	Rentabilité et time-to-market	Contrôle des développements	Mutualisation la plus large possible
Dangers	Fermeture progressive (p.ex. open core)	Déséquilibres entre partenaires	Accroissement des coûts de négociation

Formes de résistance au changement

En pratique, la communauté peut s'opposer à une situation ou à un changement de manière graduelle (1) en exprimant son mécontentement (p.ex. forums ou conférences), (2) en continuant à utiliser le logiciel mais sans plus y contribuer, (3) en cessant d'utiliser le logiciel (p.ex. migration) et (4) en organisant une scission de la communauté (fork).

Conclusion

Résumé

Claroline -et les projets qui en sont dérivés : Dokeos, Chamilo, Claroline Connect- fournit un terrain d'étude idéal pour la compréhension des mécanismes de

gouvernance et de transformation organisationnelle des projets open source. Cette recherche a permis de dresser l'historique des différents projets, de mieux comprendre les mécanismes conduisant aux forks, d'analyser les bénéfices associés à la création d'un consortium, d'identifier les difficultés associées à l'animation d'une communauté et, enfin, d'explorer les liens existant entre modèles d'affaires, logiques de gouvernance et configuration des communautés.

Transformation organisationnelle

Le projet Claroline illustre la difficulté de faire évoluer le projet tout en maintenant la cohésion de la communauté et en évitant les effets d'éviction. La pérennisation du projet implique des choix de modèle d'affaires et de gouvernance susceptibles d'éloigner certaines franges de la communauté. La prédilection pour les pôles de développeurs, pour des raisons de contrôle de la feuille de route et de qualité des développements, a entraîné un effet d'éviction sur les développeurs, par ailleurs peu nombreux, issus de la communauté des utilisateurs. Dans le cas de Dokeos, la recherche de rentabilité et d'un time-to-market réduit a également conduit à rompre avec la communauté. Cependant, la communauté apparaît comme une source de résistance au changement parmi d'autres.

Négociation avec la communauté

Pour les trois projets étudiés (Claroline / Claroline Connect, Dokeos et Chamilo), la communauté apparaît comme une force avec laquelle il faut composer (règles) et un ensemble qu'il faut pouvoir canaliser (animation). Il en résulte une lourdeur ainsi qu'un coût pour l'éditeur, en principe compensé par les contributions (promotion du projet, entraide sur les forums, documentation des pratiques, création de modules,...) issues de la communauté. En cas de faibles contributions, le recentrage sur des équipes internes ou apportées par des partenaires, plus facilement contrôlables, peut apparaître comme un choix rationnel. Des efforts en matière de communication et d'animation de la communauté pourraient cependant conduire à une solution plus équilibrée.

Reconfiguration des communautés

Les projets open source suivent généralement à leur création une logique informelle et évoluent ensuite, si nécessaire, vers une autre logique. En pratique, les trois logiques de gouvernance plus matures semblent pouvoir cohabiter sur une même niche fonctionnelle, avec cependant des publics distincts pour communauté. Les entreprises se rassembleraient alors progressivement autour de la logique commerciale (éditeurs et réseau structuré de partenaires) ; les grandes organisations, autour de la logique industrielle (mutualisation dans un cadre coopétitif) et les autres types d'acteurs (p.ex. utilisateurs isolés et très petites entreprises), autour de la logique communautaire (méritocratie et garantie d'équilibre des forces) (Table 3). Si ce mode d'évolution était validé, il annoncerait d'autres mouvements de reconfiguration des communautés au sein de l'écosystème Claroline, autour de 3 projets, incluant Claroline, Chamilo et un troisième projet occupant la place laissée vacante par Dokeos passé en logique propriétaire.

Perspectives

Les codes sources de Claroline, Dokeos, Chamilo et Claroline Connect sont disponibles en ligne. L'activité communautaire peut faire l'objet d'une analyse (métrique) et être comparée projet par projet (p.ex. importance des contributions et ventilation par partenaire). Le site OpenHub fournit des métriques précalculées ainsi que des graphiques. Ces informations n'étant pas disponible pour Claroline Connect (migration du dépôt de Sourceforge vers Github), un travail supplémentaire de collecte et d'homogénéisation des métriques est donc à prévoir.

Les dimensions de la gouvernance ont été détaillées par Markus (200) et Laffan (2012). Les modalités de gouvernance pourraient ainsi être caractérisées plus précisément, notamment à des fins de comparaison objective.

Le point de vue des utilisateurs a été approché au cours de l'étude, que ce soit par des entretiens plus courts ou l'atelier créatif organisé lors de la conférence annuelle. La réalisation d'interviews d'acteurs ayant migré permettrait d'obtenir un éclairage complémentaire sur les faiblesses du projet Claroline en matière de communication. Le constat d'un manque de communication et d'animation suppose un travail davantage ancré dans la réalité quotidienne des projets pour (1) valider ce constat, (2) proposer des mesures correctives et (3) en tester l'efficacité. Ce travail de recherche-action est une suite possible au traitement complet des entretiens réalisés pour cette étude de cas.

Bibliographie

Bonaccorsi, A., & Rossi, C. (2006). Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology & Policy*, 18(4), pp. 40-64.

CNLL (2015). Impact du Logiciel Libre / Open Source Software en France 2015-2020 - Quels enjeux de marchés, d'emploi, de formation et d'innovation . Pierre Audoin Conseil, 19 novembre 2015 ; en ligne : http://cnll.fr/static/pdf/pac-logiciels-libres-2015.pdf.

Dahlander, L., & Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research policy*, 34(4), pp. 481-493.

De Laat, P. B. (2007). Governance of open source software: state of the art. *Journal of Management & Governance*, 11(2), pp. 165-177.

Feller, J. & Fitzgerald, B. (2002). *Understanding open source software development*, Addison-Wesley.

Fitzgerald, B. (2006). The transformation of open source software. *Mis Quarterly*, pp. 587-598.

Franke, N., & Von Hippel, E. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research policy*, 32(7), 1199-1215.

Giordano, Y. (2003). *Conduire un projet de recherche. Une perspective qualitative*. Editions EMS.

Laffan, L. (2012). A new way of measuring openness: The open governance index. *Technology Innovation Management Review*, 2(1).

Lakhani, K., Wolf, B., Bates, J., & DiBona, C. (2002). *The boston consulting group hacker survey*. The Boston Consulting Group.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science*, 52(7), 1015-1030.

Markus, M. L. (2007). The governance of free/open source software projects: monolithic, multidimensional, or configurational?. *Journal of Management & Governance*, 11(2), pp. 151-163.

Roberts, J. A., Hann, I. H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. *Management science*, 52(7), pp. 984-999.

Teece, D. J. (2010). Business models, business strategy and innovation. *Long range planning*, 43(2), pp. 172-194.

Thiétart, R. A. (2007). *Méthodes de recherche en management - 3ème édition*. Dunod.

Valimaki, M. (2003). Dual Licensing in Open Source Software Industry. *Systèmes d'Information et Management*. Vol. 8 : Iss. 1 , Article 4.

Vasquez Bronfman, S., Miralles, F. (2007). Business Models in Open Source Software: do they exist?. In *12ème conférence de l'Association Information et Management (AIM)*, Lausanne (Suisse), 18-19 juin 2007.

Viseur, R. (2007). Gestion de communautés Open Source. In *12ème conférence de l'Association Information et Management (AIM)*, Lausanne (Suisse), 18-19 juin 2007.

Viseur, R. (2012). Forks impacts and motivations in free and open source projects. *International Journal of Advanced Computer Science and Applications*, 3(2), pp. 117-122.

Viseur, R. (2013a). Evolution des stratégies et modèles d'affaires des éditeurs Open Source face au Cloud computing. *Terminal. Technologie de l'information, culture & société*, (113-114), pp. 173-193.

Viseur R., Pinchart L. (2013b). Developing Free Software within a Major ICT Company, *CommEx*, Capodistria (Slovenia).

Viseur, R., & Robles, G. (2015). First Results About Motivation and Impact of License Changes in Open Source Projects. In *IFIP International Conference on Open Source Systems*, pp. 137-145, Springer.

Viseur, R. (2016). Gouvernance des projets open source. In *INFORSID*, Grenoble (France), pp. 181-198.

West, J., & O'mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and innovation*, 15(2), pp. 145-168.

Yin, R. K. (2009). *Case study research: Design and methods*. Sage publications.

Gestion d'échantillons pour la recherche scientifique avec Collec-Science

Eric Quinton [1], Christine Plumejeaud-Perreau [2], Hector Linyer [2], Julien Ancelin [2, 3], Cécile Pignol [4], Sébastien Cipière [5], Wilfried Heintz [6], Sylvie Damy [7], Vincent Bretagnolle [8]

1. *IRSTEA - Unité de recherche Écosystèmes aquatiques et changements globaux*
   50, avenue de Verdun
   33612 CESTAS, France
   eric.quinton@irstea.fr

2. *Littoral Environnement et Sociétés, UMR 7266*
   2 rue Olympe de Gouges
   17000 La Rochelle, France
   christine.plumejeaud-perreau@univ-lr.fr,hector.linyer@univ-lr.fr

3. *UE0057 DSLP Domaine expérimental de Saint-Laurent de la Prée INRA*
   545 route du bois mâché
   17450 Saint-Laurent de la Prée, France
   julien.ancelin@inra.fr

4. *Laboratoire EDYTEM – UMR 5204*
   Bâtiment Pôle Montagne F-73376 LE BOURGET DU LAC Cédex
   cecile.pignol@univ-smb.fr

5. *Université Clermont Auvergne – EDSPI UBP*
   Campus Les Cézeaux
   63170 AUBIERE
   sebastien.cipiere@uca.fr

6. *INRA – DYNAFOR – UMR 1201*
   24 chemin de Borde-Rouge – Auzeville CS 52627
   31326 CASTANET-TOLOSAN CEDEX
   wilfried.heintz@inra.fr

7. *Université de Bourgogne Franche-Comté – UMR6249 – Laboratoire Chrono-environnement*
   16 route de Gray
   25030 Besançon cedex
   Sylvie.Damy@univ-fcomte.fr

Document numérique – n° 2-3/2018, 1-21

8. Centre d'études biologiques de Chizé
CNRS UMR7372 – Univesité de La Rochelle
405, Route de la Canauderie
79360 Villiers-en-Bois
vincent.bretagnolle@cebc.cnrs.fr

RÉSUMÉ. *Les acteurs des laboratoires de recherche scientifique environnementale collectent régulièrement de nombreux échantillons qui sont ensuite analysés et stockés. Leur gestion sur le long terme s'inscrit dans une stratégie qu'il s'agit de définir puis de mettre en œuvre via des outils informatiques adaptés. Cet article présente cette stratégie, puis sa déclinaison dans un système d'information développé sous le nom de Collec-Science, offrant un support adéquat pour la traçabilité, la diversité des données à traiter et l'autonomie des utilisateurs. Il présente également les perspectives de ces travaux en matière d'animation de communauté scientifique, à la fois sur les plans organisationnels et opérationnels, dans le contexte d'une science ouverte.*

ABSTRACT. *Scientific teams for environmental research collect many samples (biological or physical) from fields for their analysis, and have to store them for a long while. The management of such samples requires a strategy relying on an efficient Laboratory Information Management System, with regards to the specific needs of this domain. This paper exposes such a strategy, and how it is implemented inside a software named Collec-Science. In particular, it adresses the need for tracability, security, and a greater genericity and freedom for researchers. The whole information system has to be integrated inside an ecosystem of tools for the research, and we explain how we face the challenge in terms of organisation and interoperability around the solution.*

MOTS-CLÉS : *échantillon, traçabilité, organisation, QR code, ouverture*

KEYWORDS: *sample, traceability, organisation, QR code, open-science*

DOI:10.3166/DN.99.2-3.1-21

1. Introduction

Pour mener à bien les travaux de recherche dans le domaine des sciences de l'environnement, les scientifiques effectuent régulièrement des campagnes de prélèvement d'échantillons sur le terrain. Par exemple, l'unité de recherche *Ecosystèmes aquatiques et changements globaux* (EABX) d'IRSTEA réalise depuis plusieurs dizaines d'années des campagnes de prélèvements de poissons dans l'estuaire de la Gironde (Lobry *et al*., 2003) (Chevillot *et al*., 2016). Ceux-ci sont placés dans des récipients adaptés, avec ou sans produit de conservation (éthanol pour les tissus organiques par exemple). Une fois revenus au laboratoire, les échantillons font l'objet de diverses mesures et analyses. Ils peuvent être subdivisés en de nouveaux échantillons. Ainsi, à partir d'un poisson, il est possible de réaliser un prélèvement de tissu, ou d'en extraire des écailles ou des organes pour des analyses complémentaires. Enfin, des réanalyses sont parfois effectuées, par exemple pour confirmer la détermination du taxon (Rougier *et al*., 2012). Dans ce contexte, il est indispensable de connaître ceux qui

sont disponibles, de pouvoir les retrouver, et de connaître le produit de conservation utilisé.

Il s'agit donc ici de proposer une stratégie pour la gestion informatisée de ces échantillons au moyen d'un outil adapté. Le retour sur investissement d'un tel projet est attendu sur plusieurs axes : optimisation des emplacements de stockage, protection des échantillons qui ont une forte valeur ajoutée, réutilisation avec des échanges facilités entre laboratoires (réanalyses par exemple).

Comme dans beaucoup de laboratoires français, dans l'unité de recherche EABX, la gestion des échantillons n'était pas informatisée jusqu'en 2016. Au mieux, des feuilles Excel étaient disponibles, mais souvent, c'est la mémoire des opérateurs et la recherche directe dans les locaux de stockage qui permettait de retrouver les échantillons. Cette situation ne se limite pas au domaine de la recherche environnementale française (McNutt *et al*., 2016 ; List *et al*., 2015). La gestion des échantillons a été informatisée dans d'autres domaines, comme la santé et les études cliniques (Krestyaninova *et al*., 2009), les sciences de la vie (List *et al*., 2015) ou la gestion du patrimoine naturel avec, par exemple, l'Infrastructure de Recherche Recolnat (Museum National d'Histoire Naturelle, 2016) en France, ou le programme *Advancing Digitization of Biological Collections* aux Etats-Unis (Foundation, 2011). La mise en place de *Laboratory Information Management Systems* (LIMS), sous forme commerciale ou libre répond aux besoins de gestion des analyses, notamment en routine, et sont largement utilisés dans le domaine de la bio-informatique (Schuh, 2012 ; Dondeh *et al*., 2014 ; Müller *et al*., 2017). Un certain nombre de solutions existantes peuvent être classifiées selon leur finalité : gestion de collections patrimoniales, analyses de laboratoire, gestion de stock, métrologie, gestion de bibliothèques. La plupart répondent à un ou plusieurs des besoins identifiés, mais aucune n'est pleinement satisfaisante.

Cet article détaille d'abord les besoins qui ont été identifiés – la gestion et le suivi à long terme des échantillons collectés – et la solution envisagée pour y répondre. Après un tour d'horizon de quelques solutions actuellement existantes, la conception et l'architecture logicielle mise en place sont décrites. Il aborde également les questions soulevées par le développement d'un modèle logiciel *open-source*, celles relevant de l'inter-opérabilité et celles liées à l'animation d'une communauté autour du projet, et présente la façon dont elles ont été abordées. Enfin, la conclusion résume les points saillants de cette stratégie et présente les perspectives qu'elle offre pour les systèmes d'information dédiés à la gestion d'échantillons et des données associées.

2. Gestion d'échantillons dans le contexte d'une science ouverte

L'analyse des besoins a débuté par des interviews informelles auprès des scientifiques et des techniciens du laboratoire EABX d'Irstea. Elle a été complétée par des échanges avec d'autres laboratoires, dont les unités mixtes de recherche Littoral

Environnement et Sociétés [1] et Environnements et Paléoenvironnements Océaniques et Continentaux [2] en Nouvelle Aquitaine, qui travaillent également sur des problématiques environnementales et qui, de part leur nature intrinsèquement interdisciplinaire, traitent une grande diversité d'échantillons.

En parallèle, un dialogue a été mené durant neuf mois à partir de janvier 2016 au niveau des Zones Ateliers [3] (Plumejeaud-Perreau *et al*., 2017), qui sont amenées à collecter et à conserver sur le long terme un grand nombre d'échantillons biotiques et abiotiques. Ce dialogue a pris la forme de cinq réunions mensuelles par visioconférence, puis d'un recueil de besoins dans six documents Word rédigés par les chercheurs-utilisateurs de six Zones Ateliers, sur la base d'un exemple proposé par l'animatrice de l'action. Un document contient, par exemple dans le cas de la Zone Atelier Arc Jurassien, l'expression de trois cas d'usage très différents, détaillant le type d'étiquette souhaité et le déroulé opératoire habituel du laboratoire sur le terrain et pour le recueil d'échantillons. Nous avons également visité leurs salles de rangement lors de deux grandes réunions en septembre 2016 à Chambéry et octobre 2016 à Besançon, avec la démonstration d'un petit prototype (non conservé) sur du vrai matériel, pour obtenir des retours plus complets sur les besoins.

2.1. Traçabilité des objets et des opérations

La plupart des programmes scientifiques de suivi des milieux naturels et biophysiques sur le long terme sont amenés, en raison de leurs activités, à mettre en place des systèmes d'informations complets pour la gestion des données et des échantillons issus de l'observation sur le terrain, ainsi que des données subsidiaires résultant des analyses réalisées en laboratoire. Nous considérons qu'un système d'information est composé (1) d'une ou plusieurs bases qui conservent les données issues des enquêtes sur le terrain et des analyses au laboratoire, (2) d'interfaces qui permettent de lire et écrire dans celles-ci. Les acteurs qui interviennent sur les données (acquisition, intégration, analyse, ré-analyse) sont divers (chercheurs, stagiaires, personnel temporaire), et n'ont pas toujours connaissance de ce qui s'est fait en amont dans la chaîne de traitement d'un échantillon, et de ce qui se fera en aval. Il est toutefois indispensable de documenter systématiquement la chaîne de traitement de ces données et d'éviter des erreurs humaines afin de garantir la qualité des analyses et des conclusions scientifiques qui seront tirées de ces données. La traçabilité des données et des protocoles est un enjeu majeur au niveau de la recherche internationale (Wilkinson *et al*., 2016) pour sa reproductibilité. Notre stratégie s'inscrit dans une politique d'*open-science*, (Fecher, Friesike, 2014) et insiste notamment sur les capacités de traçabilité des opérations de la recherche.

1. https://lienss.univ-larochelle.fr/

2. http://www.epoc.u-bordeaux.fr/

3. Les Zones-Ateliers sont labellisées depuis 2017 Infrastructures de Recherche sur le long terme pour les Socio-Écosystèmes et s'inscrivent dans un réseau de recherche européen inter-établissement (e-LTSER).

2.1.1. Barcoding pour suivi informatisé des objets

L'utilisation d'un système de barcoding est très utile pour le suivi et la traçabilité des objets (échantillons, contenants) (Thompson, 1994 ; Campbell *et al*., 2012). Il nous est apparu que la gestion du stockage des échantillons s'apparente en effet grandement à celle de la gestion du stock dans un entrepôt. L'entrée d'une marchandise dans une étagère doit pouvoir être enregistrée très rapidement, et l'utilisation de systèmes automatiques de lecture par douchettes s'impose naturellement. Nous avons opté pour l'utilisation de codes-barres à deux dimensions imprimés sur des étiquettes dont le support a été sélectionné pour sa durabilité, même si, dans des cas spécifiques, des puces RFID peuvent être utilisées pour identifier certains échantillons.

2.1.2. Traçabilité des mouvements de stocks

La gestion d'un stock implique de savoir déterminer la localisation de tout échantillon, mais également de connaître et lister le contenu de tous les contenants (dénommés également *containers* en anglais). Il doit ainsi être possible de savoir s'il reste de la place pour ranger d'autres éléments dans un contenant donné (pièces, armoires, piluliers, etc.). Ce stock ne cesse d'évoluer au fil du temps, au gré des opérations menées par les expérimentateurs. La traçabilité implique une historisation [4] de ces mouvements afin de savoir qui les a réalisés, quand, et parfois, pourquoi, notamment pour les opérations de sortie.

2.1.3. Généalogie des échantillons

Il est fréquent qu'un échantillon, une fois ramené au laboratoire, fasse l'objet de prélèvements complémentaires. Par exemple, les otolithes, qui sont des os de l'oreille interne, sont prélevés sur les poissons pour calculer leur âge ou analyser les milieux traversés (Daverat *et al*., 2005) ; de même, des morceaux de tissus organiques peuvent être prélevés pour réaliser des analyses ADN ou des dosages enzymatiques. Ces échantillons dérivés doivent pouvoir être rattachés au parent pour conserver la traçabilité de leur origine.

Il est également nécessaire de pouvoir gérer des échantillons constitués de plusieurs éléments non identifiables individuellement : pour certains poissons, entre cinq et dix écailles sont prélevées, et c'est l'ensemble de celles-ci qui forment l'échantillon. Toutefois, pour les analyses, une seule est prélevée. Dans ces conditions, un des besoins récurrents est de pouvoir déterminer l'état (*i.e.* le volume) du stock restant disponible.

2.1.4. Des étiquettes adaptables

Les étiquettes doivent pouvoir s'adapter à tous les cas de figure, la taille des récipients ou des containers pouvant être très variable, depuis des caisses ou des armoires

4. mécanisme de conservation des informations précédentes – mais obsolètes – en recourant notamment à un horodatage, voire à l'enregistrement du nom des opérateurs impliqués dans la manipulation de l'information

à des tubes de laboratoire. Plusieurs étiquettes différentes doivent pouvoir être imprimées pour le même échantillon, par exemple une étiquette ronde ou carrée posée sur le bouchon d'un tube, et une autre rectangulaire sur son corps. L'utilisateur doit être à même de créer facilement les différents modèles dont il aura besoin. Selon la nature des échantillons, il doit pouvoir imprimer le produit utilisé pour la conservation, les risques associés, les métadonnées, etc.

Figure 1. Exemples d'étiquettes : à gauche pour des échantillons d'insectes, à droite, pour des extraits de carottes sédimentaires.

2.2. Autonomie des usagers et pérennité à long terme

Les échantillons et les données associées portent une valeur économique ajoutée très forte, ne serait-ce que si l'on considère le coût de collecte de la donnée, sans parler de leur conservation sur le long terme. Le protocole de Nagoya, élaboré dans le cadre de la *Convention on Biologicial Diversity* (Biologicial Diversity, 2010), rappelle notamment l'importance de la conservation des ressources génétiques et, par extension, des prélèvements biologiques, pour le patrimoine de l'humanité. Disposer d'un outil qui permette de mieux maîtriser leur suivi sur le long terme devient, dans ce contexte, indispensable. Pour la garantir, nous entendons développer une stratégie de stockage des informations sur les échantillons qui soit décentralisée, et qui permette à chaque laboratoire et chaque établissement de recherche de développer la politique qui lui semble la plus appropriée. Les données devraient rester en leur possession compte-tenu de la très longue durée de conservation envisagée. Il est donc hors de question de sous-traiter la gestion et le stockage à un organisme tiers, pour des questions de pérennité et de droits d'accès sur le long terme.

De même, nous pensons qu'il faut privilégier une solution qui soit basée sur des briques logicielles libres, et qui doit elle-même être libre. Cela garantit par ailleurs la pérennité sur le long terme de la solution, car un logiciel commercial expose les usagers à des mises à jour imposées, voire à des changements de produits selon les stratégies de rachats d'entreprises, avec des risques de pertes de données non négligeables dans ces opérations. De plus, le coût total de possession d'un logiciel com-

mercial, sur de longues périodes, est quasiment impossible à estimer, en raison des risques de changement des politiques tarifaires des éditeurs. C'est un argument largement défendu dans la communauté des LIMS en bio-informatique (List *et al*., 2015).

Les cas d'usage ont mis en évidence que les premiers étiquetages d'échantillons pouvaient être réalisés sur le terrain, dans des zones où une connexion Internet n'est pas forcément disponible. La solution adoptée doit donc pouvoir être embarquée dans des matériels portables, et il doit être possible de récupérer les échantillons saisis sur le terrain dans la base de données du site de rattachement des opérateurs. Nous avons opté pour le recours à une interface **Web** adaptable à tous types de terminaux, *via* des solutions de mise en page de type *Adaptive responsing* [5].

2.3. Plasticité des métadonnées associées aux échantillons

Dans le cadre d'une gestion d'un stock d'échantillons *stricto-sensu*, le stockage d'informations liées à leur nature ou aux conditions de collecte, comme le taxon ou le milieu de prélèvement par exemple, n'est pas indispensable : ces informations devraient être traitées par des logiciels spécialisés dédiés.

Cependant nos entretiens avec les utilisateurs-chercheurs et l'explicitation qu'ils font de leurs besoins a mis en évidence que l'ajout d'informations spécifiques, ou données « métier », était indispensable, d'une part pour faciliter l'acquisition des données sur le terrain, et d'autre part pour mieux qualifier les échantillons récoltés (ajout du taxon sur une étiquette, par exemple). Au vu de la diversité des données collectées, ces informations complémentaires (dites *métadonnées*) ne peuvent pas être définies lors de la conception du logiciel. Cela implique d'offrir un mécanisme qui permette de créer dynamiquement leur schéma. Cette fonctionnalité qui permettrait de s'adapter à la diversité des usages est caractéristique de notre environnement interdisciplinaire.

2.4. Analyse des solutions existantes

Par rapport à l'ensemble d'objectifs visés, plusieurs solutions ont été étudiées, voire testées, dans ce vaste ensemble que représente les LIMS (Dondeh *et al*., 2014 ; List *et al*., 2015), très répandus pour le domaine de la bancarisation des données biologiques (Müller *et al*., 2017), en se concentrant principalement sur les solutions libres. Afin d'en avoir une vision plus claire, nous proposons de classer ces solutions logicielles suivant la typologie décrite dans le tableau 1.

La plupart des solutions ont été conçues pour gérer des collections d'un type prédéterminé de matériel, que ce soit des cellules, des gènes, des objets d'art, des œuvres littéraires, ou des spécimens biologiques. Elles se spécialisent dans le domaine considéré : on peut citer *Omeka, Cyber-Carothèque, Specify, RecolNat* pour les collections

5. mécanisme permettant le redimensionnement et la réorganisation des éléments de la page en fonction de la taille de l'écran utilisé.

Tableau 1. Typologie des solutions étudiées

Type	Caractéristiques	Exemples
Collections patrimoniales	Données ouvertes, partagées, base centralisée, entrée par la taxonomie	*Recolnat* [a], *Cyber-carothèque* [b], *Specify* [c], *Omeka* [d], *VoSeq* [e]
Analyses de laboratoire en routine	échantillons détruits après analyse, récupération automatique des résultats issus des automates, facturation	*EnzymeTracker* (Triplet, Butler, 2012), *OpenLabFramework* [f], *OpenSpecimen* [g]
Échantillons collectés dans le cadre de projets de recherche	Durée de conservation longue (> 40 ans), échanges avec d'autres labos possible	*Barcode* (Salin, Fève, 2017), *Baobab* (Bendou *et al.*, 2017), *GeCol* [h]
Matériel d'exp. (terrain, aquariums...)	Gestion de stock	
Matériel de laboratoire	métrologie, suivi de l'entretien, assurance-qualité	*Split* [i]
Bases documentaires	prêt, recensement, mise à disposition : gestion de bibliothèque	*PMB* [j]

a. https://www.recolnat.org/
b. https://cybercarotheque.fr/
c. http://specifyx.specifysoftware.org/
d. https://omeka.org/
e. https://github.com/carlosp420/VoSeq
f. https://github.com/NanoCAN/OpenLabFramework
g. https://openspecimen.atlassian.net/wiki/spaces/CAT/overview
h. https://gecol.ird.fr
i. https://www.split.io
j. http://www.sigb.net/index.php?lvl=cmspage&pageid=2&id_logiciel=18

patrimoniales, *PMB* pour la gestion de documents et de bibliothèques, *OpenSpecimen* pour l'analyse biologique, *OpenLabFramework* pour des analyses de cellules, *VoSeq* pour les séquences génomiques, *Split* pour le suivi qualité des matériels de laboratoire. Aucune ne répond à notre besoin de souplesse et adaptabilité. Très peu sont adaptées à la gestion des mouvements des échantillons (entrées et sorties quotidiennes des stocks). La logique des collections patrimoniales n'est pas celle d'une recherche scientifique qui va utiliser, ranger puis réutiliser ou prêter les échantillons. Si certaines s'en approchent, comme *OpenSpecimen* ou *Baobab*, elles n'assurent pas la traçabilité des mouvements du stock. *Split* est un logiciel commercial, qui fonctionne avec un serveur Windows et une connexion via le protocole *Terminal Server*. Dédié à la

métrologie et aux contrôles réglementaires, il n'a pas la souplesse nécessaire pour répondre aux besoins de notre gestion d'échantillons. Quant à *PMB*, c'est un logiciel développé pour gérer les bibliothèques qui est parfaitement adapté à la récupération des informations sur les ouvrages (codes ISBN) et aux opérations de prêt (relance des lecteurs après expiration du délai de prêt, par exemple). La transposition à une gestion d'échantillons semble complexe et certaines fonctionnalités nécessaires, comme le suivi de la généalogie d'échantillons ou le sous-échantillonnage, seraient difficiles à intégrer.

La sécurisation de ces solutions est souvent insuffisante au regard des obligations liées à la politique de sécurité des systèmes d'information de l'Etat français (Legifrance, 2014). La notion même de droits différenciés par groupes d'utilisateurs est parfois absente, comme c'est le cas dans *Specify*. Leur code source n'est pas toujours disponible facilement (c'était le cas de GeCol en Juillet 2016), ou la solution proposée ne fonctionne qu'en mode hébergé dans un serveur central (cas de *BarCode* (Salin, Fève, 2017) déployé à l'INRA), ce qui va à l'encontre des principes d'autonomie qui guident notre stratégie. De plus en plus de solutions offrent la possibilité d'un déploiement sur le *cloud*, comme *Specify*, mais cette option est contraire à la réglementation de la recherche française (Legifrance, 2014) : l'hébergement de toutes les données de la recherche et produites par la recherche doit se faire dans un serveur localisé sur le territoire français.

2.5. Positionnement dans le cycle de vie de la donnée

Des efforts conséquents sont mis en œuvre au sein des laboratoires pour mettre à disposition les données acquises tout en garantissant leur traçabilité et leur réutilisabilité. Des solutions comme Dataverse (*The Dataverse Project*, 2018) permettent de gérer les informations, depuis leur collecte jusqu'à leur publication, au besoin en facilitant le travail de rédaction de *Data papers*. Elles sont complémentaires des besoins que nous avons identifiés : elles s'intéressent à la donnée proprement dite alors que notre solution doit permettre la gestion des échantillons physiques.

Pouvoir confronter une donnée avec l'échantillon qui l'a produite est nécessaire pour garantir la véracité des résultats de la recherche. Nous souhaitons y répondre en proposant des services web d'interrogation qui s'appuieront sur des vocabulaires partagés, voire normalisés.

3. Conception de la solution

Suite à l'analyse des solutions existantes et par rapport à l'ensemble des spécifications fonctionnelles que nous avons définies, la décision a été prise de développer le logiciel *Collec-Science* au sein du laboratoire EABX d'Irstea. Dans un premier temps, nous décrivons les caractéristiques principales du modèle de données, puis nous nous focalisons sur les fonctionnalités de description des échantillons implémentées dans

l'optique d'adaptabilité et de souplesse qui est la nôtre. Enfin, nous explicitons comment nous avons implémenté la génération des étiquettes.

3.1. Assurer la traçabilité des échantillons

3.1.1. Analyse du stockage et des mouvements associés

Dans l'approche que nous avons adoptée, les échantillons et les contenants (ou rangements) sont des objets dont on modélise et enregistre les mouvements. Le mouvement se caractérise par un sens (entrée, sortie), une date, un opérateur et, lors d'une entrée, d'un contenant de destination. Le déplacement d'un échantillon d'un contenant à un autre peut être réalisé soit par une nouvelle entrée, soit par une opération de sortie, puis d'entrée.

D'un point de vue implémentation, les contenants (*Container*) et les échantillons (*Sample*) héritent d'un objet de base (*Object*), qui est celui qui pourra faire l'objet d'un mouvement (*Movement*). Dans le cas d'une entrée dans le stock, le mouvement référence le contenant considéré. Ainsi un échantillon peut être rangé dans un contenant, mais un contenant peut aussi être rangé dans un contenant. Le type de mouvement (*type*), sa date (*date*) et l'opérateur (*operator*) sont également enregistrés. Le modèle de données correspondant est présenté dans la figure 3, page 14.

Pour connaître le contenu d'un contenant, il suffit de rechercher tous les derniers mouvements d'entrée des objets qui l'ont pour cible. Pour savoir où se trouve un échantillon, il suffit de rechercher le dernier mouvement créé. Pour connaître tout son historique, il suffit de rechercher tous les mouvements qui le concernent. La traçabilité de l'objet et de ses mouvements est ainsi assurée simplement.

3.1.2. Caractéristiques de l'objet de base

Dès lors que les échantillons et les containers héritent d'une même classe (*object*), il est pratique de rattacher à celle-ci des attributs génériques ou des fonctions communes. Chaque instance d'*Object* (donc, tout échantillon ou container) est identifié de manière unique (attribut UID - *Unique IDentifier*). Cela permet de créer des fonctions de manipulation communes aux deux types d'objets, comme par exemple la génération des étiquettes.

Object est porteur de propriétés communes à la fois aux échantillons et aux contenants. Il est ainsi possible d'y rajouter un statut (*Status*), d'y associer des événements (perte, indisponibilité, prêt, etc.) (*Event*) etc. Nous avons également positionné dans celui-ci des coordonnées géographiques *wgs84_x, wgs84_y*, qui correspondent au lieu de collecte dans le cas d'un échantillon, et à l'emplacement physique pour les contenants.

3.1.3. Associer types d'échantillons et types de contenant

Les échantillons sont de forme très variables : carottes géologiques de 2 m. de long, pots-pièges d'insectes de 5 cm. de diamètre, échantillons de sang de mammifères de

2 ml., etc. Leur typologie est une donnée essentielle tant pour leur caractérisation que pour leur stockage ou leurs usages possibles. Chaque échantillon doit pouvoir être rattaché à un type défini préalablement.

Il en est de même pour les contenants, qui peuvent prendre la forme d'un bâtiment, d'une pièce, d'une boite, d'un flacon, etc. Pour protéger les opérateurs, il est nécessaire de pouvoir indiquer sur leurs étiquettes les produits de conservation utilisés (éthanol, formaldéhyde pour d'anciens échantillons) ainsi que les risques associés (brûlure, explosion, cancérogène, etc.), selon le règlement européen relatif à la classification, à l'étiquetage et à l'emballage des substances et des mélanges (Union Européeene, 2008). Nous n'avons pas choisi de définir les risques dans une table dédiée : en effet, cette information est très variable et la réutilisabilité entre deux types d'échantillons très faible. Il nous a semblé plus pertinent et plus simple que les opérateurs saisissent le libellé exact. C'est également ce que nous avons appliqué pour les produits, qui ne sont volontairement pas décrits dans une table dédiée.

Lors de notre analyse, nous avons identifié qu'un échantillon était rarement séparable de son support de stockage – son contenant. Ainsi, un bocal de pêche contient à la fois les poissons récoltés et le produit de conservation. D'un point de vue « métier », l'échantillon – les poissons – se confond avec le contenant – le bocal –, qui sera étiqueté. Ainsi, chaque type d'échantillon peut être associé à un type de contenant.

3.1.4. Généalogie d'échantillons et sous-échantillonnage

Dans de nombreux protocoles de collecte, les échantillons récupérés initialement sur le terrain sont ensuite décomposés pour créer de nouveaux échantillons. Par exemple, les bocaux d'un litre contenant des poissons, des tronçons de carottes de sondage de deux mètres de long, etc., ramenés au laboratoire, font l'objet de tris et de découpages. Les nouveaux éléments obtenus sont alors eux-mêmes gérés comme de nouveaux échantillons et donc peuvent faire ensuite l'objet de nouveaux traitements, extractions, etc. Dans *Collec-Science*, l'opération porte le nom de *dérivation* : le nouvel échantillon dérive du parent.

Ce type de subdivision d'un échantillon est traité, d'un point de vue modélisation, en conservant la référence du parent dans le nouvel objet créé : cela permet de conserver la paternité et de retrouver toutes les informations afférentes.

Nous avons également tenu compte du cas où l'extraction d'une partie du matériau disponible n'est pas identifiable en tant que telle : cela peut être une écaille de poisson qui, fonctionnellement, ne peut pas être différenciée d'une autre, ou bien de quelques centimètres-cubes d'une carotte de sédiments (notion d'aliquote en chimie).

3.2. Plasticité pour dépasser l'hétérogénéité des cas d'usage

3.2.1. Une approche NoSQL pour les métadonnées métier

L'ajout d'informations spécifiques, liées soit aux conditions de la collecte, soit aux caractéristiques intrinsèques de l'échantillon (données « métier ») est nécessaire pour

faciliter à la fois le travail des opérateurs de terrain, l'étiquetage des matériaux récoltés et leur recherche dans la base de données.

Le modèle relationnel classique (Chen, 1976), tel que mis en œuvre dans les bases de données, ne permet pas d'atteindre le niveau de généricité requis : les attributs sont définis lors de la création de la base de données, et ne peuvent évoluer qu'au prix d'adaptations importantes et réservées aux développeurs. Une solution a été apportée par le modèle « entité – attribut – valeur » ou *EAV*, qui a été utilisé dans de nombreux secteurs, notamment médicaux (Dinu, Nadkarni, 2007). Dans ce modèle, les attributs sont vus comme des objets à part entière, et la valeur de l'attribut est la conjonction entre l'entité et la valeur. S'il répond au besoin du stockage, il est peu compatible avec le langage SQL, et extraire les informations oblige à des acrobaties au niveau du langage [6].

Depuis 2003, Postgresql supporte le format *hstore* (Bartunov, 2017), supplanté depuis par le format *Javascript Object Notation* ou *JSON*, qui permet une représentation, dans un seul champ, de multiples couples attribut – valeur. La souplesse d'utilisation de ce type de stockage lui confère un avantage décisif par rapport au modèle relationnel entité-attribut-valeur, d'autant que les concepteurs de PostgreSQL ont étendu le langage d'interrogation SQL en rajoutant des fonctions de recherche adaptées. La facilité de mise en œuvre de ces nouvelles syntaxes, et ce même pour des utilisateurs peu aguerris aux subtilités du langage, a donc définitivement joué en faveur du choix du type JSON pour le champ de métadonnées.

Lorsqu'un formulaire de métadonnées est renseigné pour un échantillon donné, il est sauvegardé dans le champ *metadata* de type JSON de l'échantillon (*Sample*). La structure du formulaire de métadonnées correspondante (*MetadataType*) est associée au type d'échantillon concerné (*SampleType*) : elle porte un nom (*name*) choisi par les usagers, et sa structure (*schema*) est elle-même décrite en JSON. La création des modèles de métadonnées et leur saisie dans le formulaire sont réalisées en utilisant la bibliothèque JavaScript *Alpaca.js* (Gitana Software, 2017).

Il devient ainsi possible de disposer sur le terrain d'une sorte de carnet de terrain électronique (Prud'homme, 2016) qui peut enregistrer quelques informations contextuelles lors de la création d'échantillons, tout en générant les étiquettes *ad-hoc*.

3.2.2. Définition et génération des étiquettes

Les étiquettes doivent être durables (plusieurs dizaines d'années) tant en ce qui concerne la pérennité des écritures que la résistance de la colle, et cela dans des conditions de stockage souvent difficiles (froid, chaleur, humidité). Des essais, me-

6. Il faut recourir soit à une multiplicité de vues, soit à des composants dédiés à ce type de recherche. PostgresSQL propose notamment des fonctions spécifiques (composant *tablefunc* – https://www.postgresql.org/docs/current/static/tablefunc.html), ou des fonctions de traitement de tableaux – https://www.postgresql.org/docs/current/static/functions-array.html – pour répondre à ces besoins

nés dans le cadre des Zones-Ateliers, ont permis de sélectionner des matériels adaptés (Plumejeaud-Perreau *et al.*, 2017 ; Plumejeaud-Perreau, 2017a).

Par ailleurs, il est souhaitable que chaque étiquette comprenne à la fois un code-barre pour faciliter les manipulations des échantillons, et du texte pour pouvoir identifier rapidement ce qui est manipulé, sans avoir à recourir à un dispositif de lecture.

Concernant le code-barre, nous nous sommes orientés vers le format *QR Code* (norme ISO/IEC 18004:2015), largement utilisé aujourd'hui, et recommandé dans le cadre de gestion de collections biologiques (Diazgranados, Funk, 2013). Nous avons testé et recommandé l'usage de douchettes de qualité industrielle (Datalogic, 2016) pour leur lecture, notamment pour la gestion du stock. *Collec-Science* offre la possibilité de mémoriser dans le code-barre toutes les informations concernant l'échantillon : non seulement ses données d'identification, mais également les métadonnées rattachées, et ceci dans un format JSON. Nous avons ainsi la possibilité d'imprimer, soit dans le code-barre, soit en clair, le nom de la collection, la localisation, la date de création, ou certaines informations du protocole renseignées dans les métadonnées, et ceci, à la discrétion des usagers.

La souplesse de création des modèles d'étiquettes est offerte par la bibliothèque Java FOP (*Formatting Objects Processor*) (Apache, 2016). FOP applique aux données (extraites au format XML de façon transparente pour l'utilisateur) une transformation décrite dans un fichier XSL, et produit un fichier PDF (une étiquette par page). Les *QR Codes* sont générés préalablement au format PNG, puis intégrés dans le fichier PDF à partir d'une instruction XSL. Par le biais d'une interface, les utilisateurs peuvent adapter les fichiers XSL à leurs besoins.

Le fichier PDF généré peut être imprimé soit à partir du navigateur de l'utilisateur (il est alors envoyé au client), soit être imprimé directement depuis le serveur, par une commande Linux CUPS. C'est ce qui est utilisé pour piloter les imprimantes lors des saisies réalisées sur le terrain. La figure 2 récapitule l'ensemble de la chaîne de traitement utilisée pour générer et imprimer les étiquettes.

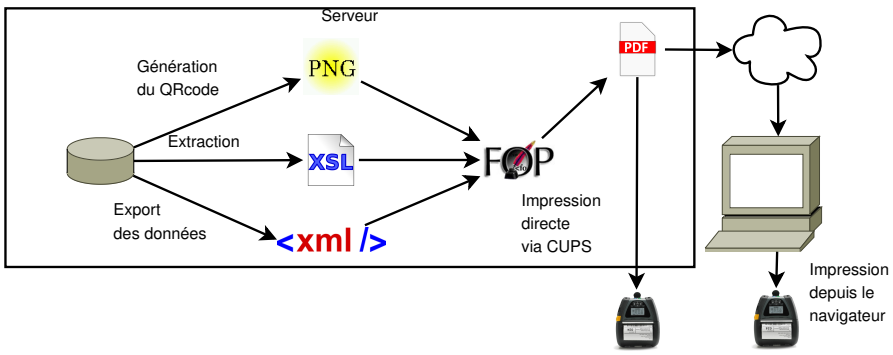


Figure 2. Processus de génération des étiquettes.

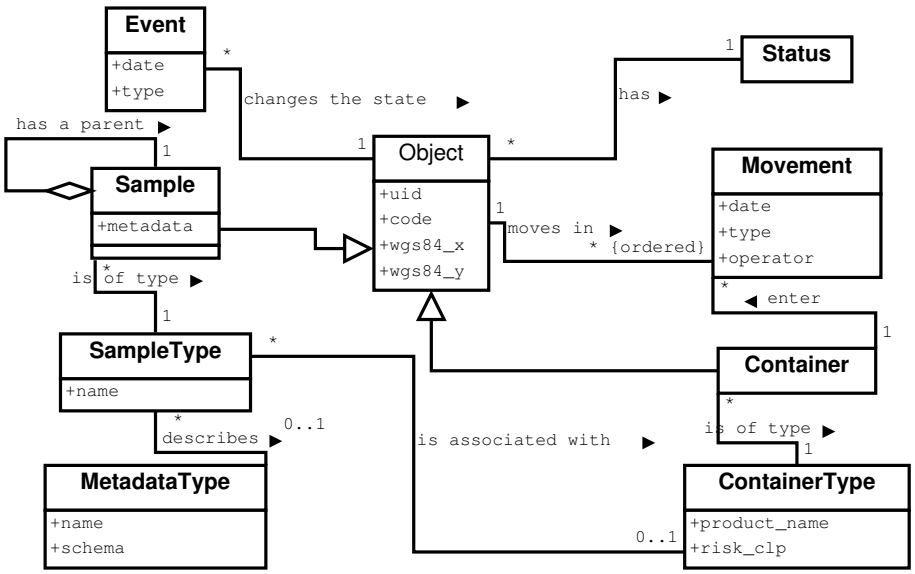


Figure 3. Diagramme des classes utilisées pour gérer les objets.

3.3. Synthèse du modèle de gestion des échantillons

La structure créée pour répondre à l'ensemble des points précédemment exposés correspond au modèle de la figure 3. Ce modèle est implémenté dans un schéma relationnel sous PostgreSQL [7].

Un objet (*Object*) se spécialise soit en un contenant (*Container*), soit en un échantillon (*Sample*). Il peut subir des événements (*Event*). Tout objet peut être stocké dans un contenant ou sorti du stock (*Movement*). Un échantillon peut être obtenu à partir d'un autre échantillon : on parle alors d'échantillon dérivé, et il peut être d'un autre type. Un type d'échantillon (*SampleType*) peut être associé à un type de contenant (*ContainerType*) même si, dans la pratique, les cas où l'association n'existe pas est rare [8]. Enfin, un modèle de métadonnées (*MetadataType*) peut être associé à un type d'échantillon.

7. https://www.postgresql.org/

8. Cela peut être le cas pour un tronc d'arbre ou une rondelle de bois de forte dimension, qui sont stockés sans être protégés par un emballage.

4. L'écosystème autour du système d'information *Collec-Science*

4.1. Vers une gestion de communauté

Pour s'inscrire dans notre perspective de science ouverte et d'autonomie des usagers, le logiciel a fait l'objet d'un dépôt à l'Agence de Protection des Programmes [9] et a été publié en Open-Source dans Github [10] sous licence AGPL [11].

Bien que le code soit publié sur la plate-forme Github qui dispose de quelques outils complémentaires comme la gestion de tickets (5 sont ouverts et 82 résolus début décembre 2017) ou un wiki, il nous a semblé nécessaire d'organiser la communication et les échanges à travers des dispositifs complémentaires. Nous avons décidé de mettre en place les premières briques d'une gestion de communauté, en nous appuyant en partie sur les recommandations du livre *Logiciels et objets libres. Animer une communauté autour d'un projet libre* (Ribas *et al*., 2016). Un site Web vitrine a été créé [12], des listes de diffusion sont maintenant accessibles, l'une pour les développeurs [13] et l'autre pour les usagers [14] de l'application. Nous avons également mis en ligne un site de démonstration [15].

4.2. Faciliter le déploiement

En raison de la stratégie *open-source* déployée, le logiciel a été conçu pour être hautement configurable et adaptable à tout type d'environnement technique. Outre un manuel d'installation (Quinton, 2017) surtout accessible à des développeurs ou des administrateurs de systèmes, des scripts complémentaires ont été écrits, soit génériques (création d'une base de données standard), soit basés sur des cas d'utilisation comme la collecte d'insectes utilisant des pots-pièges ou le stockage de carottes géologiques sédimentaires (Plumejeaud-Perreau, 2017b).

Pour faciliter le déploiement rapide de la solution sur des terminaux portables, *Collec-Science* a fait l'objet d'une containérisation à l'aide de *Docker* [16], disponible depuis le site Github [17]. Cette approche est directement inspirée du système employé pour le carnet de terrain électronique *GeoPoppy*, (Ancelin *et al*., 2016). Les containers sont déclinés pour trois systèmes d'exploitation différents : *Windows 10 Pro* pour une

9. https://www.app.asso.fr/
10. https://github.com/Irstea/collec
11. https://www.gnu.org/licenses/agpl-3.0.fr.html
12. https://www.collec-science.org/
13. https://groupes.renater.fr/sympa/info/collec-dev
14. https://groupes.renater.fr/sympa/info/collec-users
15. https://collec-science.irstea.fr
16. Docker est un logiciel qui permet de faire fonctionner un serveur à l'intérieur d'un autre serveur, quel que soit le système d'exploitation sous-jacent, et en automatisant notamment les installations. *cf.* https://www.docker.com
17. https://github.com/jancelin/docker-collec

implantation dans des tablettes, *Debian Linux 9* pour un déploiement de serveurs, ou *Raspbian*, pour des solutions embarquées légères [18].

4.3. Assurer la compatibilité avec d'autres dispositifs

Chaque objet est identifié par un numéro unique auto-généré (*Unique IDentifier* ou UID). Pour permettre les échanges entre les différentes bases de données de Collec-Science, ce numéro est associé à un code qui identifie de façon unique l'instance de la base de données considérée. Ces codes sont actuellement recensés dans le site web de l'application [19]. Pour permettre les échanges d'échantillons entre plusieurs instances de *Collec-Science*, nous avons rajouté l'attribut *dbuid_origin*, qui indique le numéro attribué dans l'instance initiale. Il est composé de la concaténation du code de la base de données d'origine et de son UID. Cela permet de conserver les étiquettes créées dans une autre instance de base de données, et c'est ce mécanisme qui est utilisé pour pouvoir transférer dans la base centrale les échantillons créés lors des missions sur le terrain.

Par ailleurs, pour répondre à un besoin croissant d'interconnexion avec des bases de données métiers externes, chaque objet (échantillon ou container) peut être associé à plusieurs identifiants externes. Cette extension du modèle permet par exemple d'associer l'identifiant unique international des échantillons géologiques ou *International Geo Sample Number* (*International Geo Sample Number*, 2017) aux carottes sédimentaires, que les géologues peuvent obtenir auprès du registre international SESAR (Gil *et al.*, 2016).

5. Conclusion

Cet article décrit une approche pour répondre au besoin de gestion des échantillons dans la recherche environnementale, et son implémentation sous la forme d'un système d'information, nommé *Collec-Science*. Voici en résumé, les points saillants de notre proposition et nos perspectives.

Nous avons souligné l'importance de la traçabilité des échantillons de leur subdivision, du suivi des mouvements de stocks, de la connaissance des risques associés aux produits de conservation utilisés, et d'un étiquetage largement paramétrable basé sur l'utilisation d'un QR Code. Tout ceci permet de répondre aux enjeux du stockage sur le long terme.

Nous avons privilégié la souplesse de paramétrage de *Collec-Science* pour répondre à la diversité des protocoles de collecte. Le choix de l'*open-source* et la volonté de proposer une solution facile à implanter dans les laboratoires visaient à défendre

18. Raspbian fonctionne sur l'architecture ARM des nano-ordinateurs de marque *Raspberry* – https://www.raspberrypi.org

19. https://www.collec-science.org/faq/

une science autonome et soucieuse de préserver la reproductibilité des recherches. Le recours à des containers de type *Docker* facilite le déploiement, notamment pour les solutions embarquées utilisées pour l'enregistrement des données sur le terrain. La saisie des informations « métier » associées aux échantillons permet de répondre à la plupart des besoins rencontrés par les équipes techniques, tant lors des opérations de collecte que pour l'étiquetage et le stockage.

Les mesures prises pour faciliter la dissémination de la solution (modèle *open-source*, animation de la communauté, documentation, etc.) se traduisent par un intérêt croissant de la part de nombreux laboratoires. Pour répondre à celui-ci, le projet BED [20], financé en 2018 par les Zones Ateliers, a engagé de nouvelles actions pour former les utilisateurs (avec la mise en place d'un atelier notamment). Il prévoit également la mise à disposition d'un ingénieur sur site pour des périodes d'une à trois semaines pour les aider à paramétrer et utiliser le logiciel.

En l'état actuel, les processus d'interconnexion restent assez frustres et permettent seulement d'assurer une compatibilité, et non une véritable interopérabilité avec d'autres dispositifs. La mise en place de services Web permettrait par exemple d'approvisionner automatiquement Collec-Science avec des données pré-existantes, ou d'exporter des informations vers des logiciels métiers spécialisés. La conception d'une architecture orientée *service* nécessite cependant l'implication de la communauté des utilisateurs pour définir un standard en matière de description de l'information concernant les échantillons. Elle devra intégrer une réflexion sur les niveaux de droits et d'authentification à implémenter pour garantir la sécurité de l'écosystème dans son ensemble.

Nous sommes impliqués depuis fin 2017 dans un groupe d'intérêt du consortium *Research Data Alliance* (Lehnert, 2017), pour contribuer au développement d'une norme descriptive pour un échantillon physique qui deviendrait un standard international. Ces travaux s'appuient sur la norme ISO 19156 et l'ontologie *Observation & Measurement* (Cox, 2016) [21] et une définition [22] qui peut être commentée en ligne. Les différents composants du modèle trouvent bien leur correspondance dans le modèle de données de Collec-Science. Les informations qui sont rattachées aux échantillons seront transposées dans le modèle standard qui est en cours d'élaboration, et des identifiants pérennes seront attribués. Cela devrait permettre de les référencer dans les résultats des analyses, et de pouvoir les retrouver depuis les articles qui les évoqueraient.

Notre objectif est d'aboutir à la mise en place d'une interopérabilité technique comme celle développée pour les données géographiques au sein de l'*Open Geospatial Consortium* (OGC), afin de disposer de spécifications de Services Web normalisés pour l'échange automatisé d'informations sur les échantillons.

---

20. https://www-iuem.univ-brest.fr/pops/projects/za-bancarisation-bed
21. www.opengeospatial.org/standards/om
22. https://confluence.csiro.au/pages/viewpage.action?pageId=413958301

6. Remerciements

Ce travail est issu de réflexions menées à la fois dans le cadre des Zones Ateliers, avec en particulier Emmanuelle Pelletier-Montargès au LIEC, Francis Raoul à Chrono-environnement, Isabelle Badenhausser au CEBC, qui ont alimenté le projet par la description des besoins qu'ils avaient. Ce travail a bénéficié aussi de la réflexion sur le modèle de gestion de stock de carottes géologiques menée par les membres du projet Equipex CLIMCOR (C2FN-DT INSU), notamment Arnaud Caillo (OASU) et Isabelle Billy (EPOC) à Bordeaux, et Elodie Godinho et Karim Bernardet de la DT INSU à la Seyne/Mer. Les auteurs sont vivement reconnaissants à ces personnes pour les échanges constructifs et leurs apports à la conception du système.

Bibliographie

Ancelin J., Odoux J. F., Schmit O., Caille A. (2016). Géo-Poppy, un serveur web SIG portable pour le recueil de données terrain. *Géomatique Expert*, n° 109, p. 42–48. Consulté sur https://hal.archives-ouvertes.fr/hal-01354212

Apache. (2016). *The Apache FOP Project.* Consulté sur https://xmlgraphics.apache.org/fop/

Bartunov O. (2017). *Json in postgres - the present and future.* Consulté sur http://www.sai.msu.su/~megera/postgres/talks/jsonb-pgconf.us-2017.pdf

Bendou H., Sizani L., Reid T., Swanepoel C., Ademuyiwa T., Merino-Martinez R. *et al.* (2017, avril). Baobab Laboratory Information Management System: Development of an Open-Source Laboratory Information Management System for Biobanking. *Biopreservation and Biobanking*, vol. 15, n° 2, p. 116–120. Consulté sur http://online.liebertpub.com/doi/10.1089/bio.2017.0014

Biologicial Diversity C. on. (2010). *About the nagoya protocol.* Consulté sur https://www.cbd.int/abs/about/default.shtml/

Campbell L. D., Betsou F., Garcia D. L., Giri J. G., Pitt K. E., Pugh R. S. *et al.* (2012, avril). Development of the *ISBER Best Practices for Repositories: Collection, Storage, Retrieval and Distribution of Biological Materials for Research*. *Biopreservation and Biobanking*, vol. 10, n° 2, p. 232–233. Consulté sur http://online.liebertpub.com/doi/abs/10.1089/bio.2012.1025

Chen P. P.-S. (1976, mars). The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, vol. 1, n° 1, p. 9–36. Consulté sur http://doi.acm.org/10.1145/320434.320440

Chevillot X., Pierre M., Rigaud A., Drouineau H., Chaalali A., Sautour B. *et al.* (2016). Abrupt shifts in the Gironde fish community: an indicator of ecological changes in an estuarine ecosystem. *Marine Ecology Progress Series*, vol. 549, p. 137-151. Consulté sur https://hal.archives-ouvertes.fr/hal-01411213

Cox S. J. (2016, décembre). Ontology for observations and sampling features, with alignments to existing models. *Semantic Web*, vol. 8, n° 3, p. 453–470. Consulté sur http://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-160214

Datalogic. (2016). *Fiche technique de la qbt 2400.* Consulté sur http://www-iuem.univ-brest.fr/pops/attachments/958

The dataverse project. (2018). Consulté sur https://dataverse.org

Daverat F., Tomas J., Lahaye M., Palmer M., Elie P. (2005). Tracking continental habitat shifts of eels using otolith sr/ca ratios: validation and application to the coastal, estuarine and riverine eels of the girondegaronnedordogne watershed. , vol. 56, n° 5, p. 619-627. Consulté sur http://www.publish.csiro.au/paper/MF04175

Diazgranados M., Funk V. (2013, juillet). Utility of QR codes in biological collections. *PhytoKeys*, vol. 25, p. 21–34. Consulté sur http://www.pensoft.net/journals/phytokeys/article/5175/abstract/utility-of-qr-codes-in-biological-collections

Dinu V., Nadkarni P. (2007). Guidelines for the effective use of entity?attribute?value modeling for biomedical databases. *International Journal of Medical Informatics*, vol. 76, n° 11, p. 769 - 779. Consulté sur http://www.sciencedirect.com/science/article/pii/S1386505606002371

Dondeh B. L., Lawlor R., Alteyrac L., Bongcam-Rudloff E., Labib R., Caboux E. *et al.* (2014, décembre). *Review / Evaluation of LIMS/Biobank Open source systems.* BioBanking and Molecular Resource Infrastructure of Sweden. Consulté sur www.bbmri.se/Global/Nyhetsarkiv/2015/LIMS_Evaluations_Final.pdf

Fecher B., Friesike S. (2014). Open Science: One Term, Five Schools of Thought. In S. Bartling, S. Friesike (Eds.), *Opening Science: The Evolving Guide on How the Internet is Changing Research, Collaboration and Scholarly Publishing*, p. 17–47. Cham, Springer International Publishing. Consulté sur https://doi.org/10.1007/978-3-319-00026-8_2 (DOI: 10.1007/978-3-319-00026-8_2)

Foundation N. S. (2011). *Advancing Digitization of Biodiversity Collections.* Consulté sur https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503559

Gil Y., David C. H., Demir I., Essawy B. T., Fulweiler R. W., Goodall J. L. *et al.* (2016, octobre). Toward the Geoscience Paper of the Future: Best practices for documenting and sharing research from data to software to provenance: Geoscience Paper of the Future. *Earth and Space Science*, vol. 3, n° 10, p. 388–415. Consulté sur http://doi.wiley.com/10.1002/2015EA000136

Gitana Software I. (2017). *Alpaca - easy forms for jquery.* Consulté sur http://alpacajs.org

International Geo Sample Number. (2017). Consulté sur http://www.igsn.org

Krestyaninova M., Zarins A., Viksna J., Kurbatova N., Rucevskis P., Neogi S. G. *et al.* (2009, octobre). A System for Information Management in BioMedical Studies–SIMBioMS. *Bioinformatics*, vol. 25, n° 20, p. 2768–2769. Consulté sur https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp420

Legifrance. (2014). *Politique de sécurité des systèmes d'information de l'Etat.* Consulté sur http://circulaire.legifrance.gouv.fr/index.php?action=afficherCirculaire&hit=1&retourAccueil=1&r=38641

Lehnert K. (2017, septembre). *IG Physical Samples and Collections in the Research Data Ecosystem.* Research Data Alliance. Consulté sur https://www.rd-alliance.org/system/files/documents/RDA10_IGPhysSam_BoF.pdf

List M., Schmidt S., Trojnar J., Thomas J., Thomassen M., Kruse T. A. *et al.* (2015, mai). Efficient Sample Tracking With OpenLabFramework. *Scientific Reports*, vol. 4, n° 1. Consulté sur http://www.nature.com/articles/srep04278

Lobry J., Mourand L., Rochard E., Elie P. (2003). Structure of the gironde estuarine fish assemblages: a comparison of european estuaries perspective. *Aquatic Living Resources*, vol. 16, n° 2, p. 47?58.

McNutt M., Lehnert K., Hanson B., Nosek B. A., Ellison A. M., King J. L. (2016, mars). Liberating field science samples and data. *Science*, vol. 351, n° 6277, p. 1024–1026. Consulté sur http://www.sciencemag.org/cgi/doi/10.1126/science.aad7048

Museum National d'Histoire Naturelle. (2016). *Recolnat, valorisation de 350 ans de collections d'histoire naturelle : une plateforme numérique*. Compte-rendu scientifique INFRASTRUCTURES. Museum National d'Histoire Naturelle. Consulté sur https://www.recolnat.org

Müller H., Malservet N., Quinlan P., Reihs R., Penicaud M., Chami A. *et al*. (2017, mars). From the evaluation of existing solutions to an all-inclusive package for biobanks. *Health and Technology*, vol. 7, n° 1, p. 89–95. Consulté sur http://link.springer.com/10.1007/s12553-016-0175-x

Plumejeaud-Perreau C. (2017a, janvier). *Bancarisation des données : gestion des échantillons et des protocoles.* Honfleur. Consulté sur http://www-iuem.univ-brest.fr/pops/attachments/1279

Plumejeaud-Perreau C. (2017b). *Guide d'utilisation et remarques sur collec-science.* Consulté sur http://www-iuem.univ-brest.fr/pops/attachments/1380

Plumejeaud-Perreau C., Linyer H., Pignol C., Cipière S., Quinton E., Ancelin J. *et al*. (2017, octobre). QR-CODE PROJECT : Towards better traceability of field sampling data. In *International long term ecological research network joint conference.* Nantes, France. Consulté sur https://rza.sciencesconf.org/

Prud'homme O. (2016). *Carnets de terrain électroniques: bref tour d'horizon des outils disponibles.* Sète, France. Consulté sur https://oreme.org/content/download/627/6922

Quinton E. (2017). *Logiciel Collec-Science - installation et configuration v1.2.* Consulté sur https://github.com/Irstea/collec/blob/master/database/documentation/collec_installation_configuration.pdf

Ribas S., Guillaud P., Ubeda S. (2016). *Logiciels et objets libres. animer une communauté autour d?un projet libre* (Framasoft, Ed.). Consulté sur framabook.org/logiciels-et-objets-libres/

Rougier T., Lambert P., Drouineau H., Girardin M., Castelnaud G., Carry L. *et al*. (2012). Collapse of allis shad, alosa alosa, in the gironde system (southwest france): environmental change, fishing mortality, or allee effect? *ICES Journal of Marine Science*, vol. 69, n° 10, p. 1802-1811. Consulté sur http://dx.doi.org/10.1093/icesjms/fss149

Salin G., Fève K. (2017, juin). *Présentation BARCODE.* Toulouse, France. Consulté sur http://get.genotoul.fr/wp-content/uploads/2017/05/BARCODE_Pr%C3%A9sentation_INRA_DYNAFOR_Katia_GGeral_210217.pdf

Schuh R. (2012, juillet). Integrating specimen databases and revisionary systematics. *ZooKeys*, vol. 209, p. 255–267. Consulté sur http://zookeys.pensoft.net/articles.php?id=2908

Thompson F.-C. (1994). Bar codes and specimen data management. *Insect Collection News*, vol. 9, p. 2–4.

Triplet T., Butler G. (2012). The EnzymeTracker: an open-source laboratory information management system for sample tracking. *BMC Bioinformatics*, vol. 13, n° 1, p. 15. Consulté sur http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-13-15

Union Européeene. (2008). *Règlement (ce) no 1272/2008 du parlement européen et du conseil du 16 décembre 2008 relatif à la classification, à l'étiquetage et à l'emballage des substances et des mélanges, modifiant et abrogeant les directives 67/548/cee et 1999/45/ce et modifiant le règlement (ce) n o 1907/2006 (texte présentant de l'intérêt pour l'eee).* Consulté sur http://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:32008R1272

Wilkinson M. D., Dumontier M., Aalbersberg I. J., Appleton G., Axton M., Baak A. *et al*. (2016, mars). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, vol. 3, p. 160018. Consulté sur http://www.nature.com/articles/sdata201618

Relations topologiques pour l'intégration sémantique de données et images d'observation de la Terre

Herbert Arenas, Nathalie Aussenac-Gilles, Catherine Comparot, Cassia Trojahn

IRIT, CNRS, Université de Toulouse, France
{prenom.nom}@irit.fr

RÉSUMÉ. *Les satellites d'observation de la Terre lancés récemment par l'ESA délivrent entre 8 à 10 To de données par jour, offrant de nouvelles opportunités pour la gestion de l'environnement, l'étude du climat ou de l'évolution urbaine. Les applications de ces domaines requièrent d'enrichir les méta-données d'image avec des données provenant de diverses sources (fixes ou dynamiques) pour faciliter la prise de décision sur les zones étudiées. L'intégration de données hétérogènes soulève un défi majeur. Nous présentons une approche s'appuyant sur les représentations spatio-temporelles pour enrichir des métadonnées d'images satellites avec des données ouvertes. Elle s'appuie sur un vocabulaire qui spécialise des standards (comme SOSA et GeoSPARQL) ainsi qu'un processus pour aligner et intégrer des données géospatiales hétérogènes. Le processus exploite le tuilage des images, représentant une zone fixe d'une grille associée à la surface terrestre, pour traiter les données ayant une composante spatiale fixe. Les relations temporelles, quant à elles, sont calculées à la volée à partir d'une topologie temporelle.*

ABSTRACT. *Recently launched Earth observation satellites, which deliver between 8 and 10TB of image data per day, open emerging opportunities in domains ranging from environmental monitoring to urban planning and climate studies. However, domain-oriented applications require image metadata to be enriched with data coming from various sources (either static or dynamic), in order to support decision making on the observed areas. The integration of heterogeneous data highly relying on spatio-temporal representations raises a major challenge. We present a semantic approach to support data integration thannks to spatio-temporal relations between image metadata and various open data sets. We propose a vocabulary that specializes standards (like SOSA, GeoSPARQL) as well as a process to map and integrate heterogeneous geo-spatial data sets. This process relies on image tiles, representing a fixed area of a grid associated with the Earth's surface, to handle data with a fixed spatial component. The temporal relationships are calculated on the fly based on temporal topology.*

MOTS-CLÉS : *intégration de données, vocabulaire sémantique, observations de la Terre*

KEYWORDS: *semantic vocabulary, earth observation, data integration*

1. Introduction

L'observation de la Terre offre une valeur ajoutée à un grand nombre de domaines. Récemment l'Agence Spatiale Européenne (ESA) a lancé le programme Sentinel avec deux types de satellites, Sentinel-1 and Sentinel-2, qui transmettent des images de haute qualité (entre 8 à 10 To de données quotidiennement). Ces images sont captées selon différentes technologies et libres d'accès. Cette disponibilité des données ouvre de nombreuses perspectives économiques grâce à de nouvelles applications dans des domaines aussi variés que l'agriculture, l'environnement, l'urbanisme, l'océanographie ou la climatologie. Ces applications métier ont néanmoins besoin de coupler les images avec des données sur les zones observées. Ces données sont accessibles à partir de différentes sources dans des formats hétérogènes et des temporalités différentes : elles peuvent être statiques, comme les données sur le relief ou la couverture terrestre, ou dynamiques, comme les observations météorologiques. Elles peuvent être utiles par exemple pour indiquer qu'une image contient une région touchée par un phénomène tel qu'un tremblement de terre ou une canicule, et sont alors utilisées pour décider des actions à mener dans cette zone ou conduire à des analyses à plus long terme. Plus encore, en exploitant les caractéristiques spatio-temporelles d'un phénomène (son emprise spatiale et sa date), il devient possible de savoir si une entité localisée dans l'emprise de l'image (e.g. une ville) a subi le même phénomène.

Dans ce contexte, les images étant décrites par des méta-données, une des difficultés est d'intégrer à ces méta-données, des données hétérogènes provenant de sources diverses. L'apport des technologies sémantiques pour faciliter cette tâche a été démontré dans des travaux antérieurs (Reitsma, Albrecht, 2005) (Sukhobok *et al*., 2017). En lien avec ces travaux, nous présentons une approche sémantique, basée sur un vocabulaire, pour intégrer des données en vue d'enrichir des méta-données d'images satellites avec des données provenant de sources diverses. Le vocabulaire sémantique doit être défini de manière à représenter les données et à y accéder de façon homogène. Cette approche requiert aussi des règles de transformations pour peupler le modèle avec les données de ces sources hétérogènes. Une caractéristique essentielle des observations de la Terre est qu'elles sont géo-localisées et datées. Elles peuvent donc être liées par des relations topologiques spatiales et temporelles. Le processus d'intégration des données doit gérer correctement les propriétés et relations spatiales et temporelles. Pour éviter de dupliquer des données fixes, i.e. valides pour toutes les images d'une même zone au cours du temps, il est commode d'exploiter le concept de tuile ("tile" en anglais) défini par l'ESA : la surface terrestre est associée à une grille dans laquelle une tuile représente une zone fixe de cette surface.

Nous présentons ici un cadre pour l'intégration sémantique de diverses données géographiques et des méta-données d'images satellites. Celui-ci s'appuie sur un vocabulaire que nous avons défini ; il permet d'associer les mêmes classes à ces différents types de données, et de les représenter comme des entités ayant des propriétés spatiales et temporelles. Les données proviennent d'ensembles de données géospatiales avec des formats hétérogènes (shapefile, KML, CSV, GeoJSON, TIFF). Une partie de ces données sont dites "contextuelles" et sont le résultat de mesures : nous les trai-

tons comme des données de capteurs. Ce vocabulaire spécialise ainsi des vocabulaires connus du LOD, dont SOSA [1] et GeoSPARQL (Kolas *et al*., 2013).

Une seconde contribution est le processus d'intégration basé sur la topologie des entités et les principes des données liées afin de gérer les problèmes d'hétérogénéité. Pour chaque ensemble de données à intégrer, nous avons défini des patrons et fonctions de transformation. Les propriétés temporelles contribuent à l'intégration des données dynamiques. Pour traiter la composante spatiale des données statiques et dynamiques, le processus s'appuie sur le tuilage des images qui permet de réduire le volume de données à traiter. Enfin, le processus d'intégration génère plusieurs entrepôts de données et des fichiers JSON de méta-données enrichies ou de mesures qui peuvent être réutilisés à des fins diverses. Nous illustrons notre approche par un cas d'étude qui exploite des méta-données d'images Sentinel-2 fournies par le CNES, les tuilages d'images de l'ESA et des données contextuelles : des données de météorologie fournies par Météo France, la couverture végétale terrestre et les entités administratives. Grâce à la représentation sémantique de toutes ces données [2], nous avons lié les méta-données de chaque image aux données s'appliquant à la zone terrestre définie par l'emprise (ou *footprint*) de cette image à la date de sa saisie.

Le reste de l'article est organisé comme suit. La Section 2 discute des travaux liés. La Section 3 offre un aperçu de notre approche. La Section 4 présente le modèle proposé, et la Section 5 détaille les processus de sélection, d'alignement et d'intégration de données. Nous concluons et présentons des perspectives à ce travail en Section 6.

2. Publication et mise en relation de données d'observation de la Terre

Publication de données liées d'observations de la Terre. Rendre disponibles sous forme de données ouvertes des données géo-localisées et les relier à des bases de connaissances couvrant d'autres aspects du domaine facilite le développement de services ayant une grande valeur environnementale et commerciale (Smeros, Koubarakis, 2016). Dans ce but, les principes du LOD (Linked Open Data) définissent des bonnes pratiques pour exposer, partager et intégrer des données au format RDF et identifiées par des URI déréférençables sur le Web (Heath, Bizer, 2011 ; Blázquez *et al*., 2014 ; Sukhobok *et al*., 2017). Le W3C fournit d'ailleurs des recommandations pour publier des données spatiales sous forme de LOD et gérer les relations spatiales. Il propose aussi des systèmes de référence (CRS ou *Coordinate Reference Systems*) pour leur réprésentation (Tandy *et al*., 2017). Des projets européens tels que LEO et TELEIOS ont commencé à publier des données liées au sein d'*Observatoires Virtuels* promus par l'initiative internationale IVOA (International Virtual Observatory Alliance) (Koubarakis *et al*., 2012). Grâce aux nouveaux liens identifiés entre les données et aux connaissances inférées, ces observatoires virtuels fournissent des en-

1. https://www.w3.org/2015/spatial/wiki/SOSA_Ontology
2. Les données sont publiées sur http://melodi.irit.fr/sparkindata/

sembles d'informations plus riches que les images d'observation de la Terre et leurs méta-données seules (Koubarakis *et al*., 2012).

Les données géospatiales sont souvent disponibles au format "raster", un format défini initialement pour les images. Pour représenter ces données, souvent volumineuses, dans le LOD, le W3C suggère l'ontologie QB (RDF Data Cube) (Brizhinev *et al*., 2017) combinée à d'autres ontologies standards du W3C et de l'OGC dont SSN (Semantic Sensor Network) [3], OWL-Time [4], SKOS [5], PROV-O [6] et la récente extension de DataCube pour les entités spatio-temporelles, QB4ST [7]. Pour représenter sous forme sémantique les données spatio-temporelles des CRS, les entités topographiques et leurs géométries, plusieurs modèles existent. A cette fin, Atemezing a défini quatre vocabulaires qui étendent des vocabulaires existants et offrent deux avantages supplémentaires (Atemezing, 2015) : une utilisation explicite du CRS identifié par des URI pour la géométrie, et la possibilité de décrire des géométries structurées en RDF. De même, le projet GeoKnow a tiré parti des données spatiales du LOD et mis à disposition des outils pour collecter, fusionner, agréger des données spatiales ainsi qu'une architecture pour les publier, les réutiliser et les visualiser (García-Rojas *et al*., 2013).

Mise en relation de données spatio-temporelles et découverte automatique de liens. Lier des données d'observation de la Terre signifie découvrir des liens spatiaux et temporels au sein du graphe RDF obtenu après la publication des données (Blázquez *et al*., 2012). Grâce aux propriétés spatiales, les données d'observations peuvent être associées aux tuiles et ainsi aux images d'observation de la Terre. Grâce aux propriétés temporelles, les observations temporelles peuvent aussi être liées aux images. Lorsque des entités de même nature sont collectées à partir de diverses sources, un algorithme d'association d'entités peut identifier des alignements entre des entités spatiales similaires ou identiques. On peut définir ces algorithmes de manière à ne prendre en compte que les propriétés spatiales et temporelles pour produire ces alignements.

L'OGC a introduit la notion de *données géo-liées* ("geolinked data") pour faire référence aux données liées géographiquement. Dans les premiers travaux, la géométrie était stockée dans un ensemble de données géospatiales séparé, et non directement comme valeur d'attribut. Cette option est plus contraignante lorsqu'il faut comparer la géométrie de chaque entité. C'est pourquoi les entrepôts actuels mémorisent ensemble une représentation RDF de la géométrie et une représentation RDF des entités spatiales. Suivant la source, la géométrie de chaque donnée est décrite par un ou des points, lignes ou polygones. Atemezing a identifié des outils pour construire une représentation RDF de la géométrie, comme Geometry2RDF [8] ou TripleGeo [9] (Atemezing,

3. http://purl.oclc.org/NET/ssnx/ssn
4. https://www.w3.org/TR/owl-time
5. http://www.w3.org/2004/02/skos/core
6. https://www.w3.org/TR/prov-o
7. https://www.w3.org/TR/qb4st/
8. https://github.com/boricles/geometry2rdf
9. https://github.com/GeoKnow/TripleGeo

2015). Le processus défini par Vilches-Blázquez et ses collègues compare précisément les géométries de données de telle sorte que les données spatiales puissent être retrouvées et reliées à un haut niveau de granularité (Blázquez *et al*., 2014). Pour aller plus loin et retrouver précisément tout ce qui est localisé à un endroit précis, les images de satellites peuvent être classées et enrichies de données externes dans un format sémantique qui permet de raisonner sur ces données grâce à des règles de raisonnement spatial spécifiques au domaine (Alirezaie *et al*., 2017).

Pour calculer des liens entre des ressources LOD possédant des propriétés temporelles, et donc assimilables à des événements, Georgala et ses collègues utilisent les intervalles de l'algèbre des intervalles d'Allen (Georgala *et al*., 2016). Leur proposition peut s'appliquer aux données géolocalisées ayant une dimension temporelle. Leur approche, AEGLE, réduit le nombre de relations temporelles d'Allen de 13 à 8, et les implémente de façon optimale pour effectuer plus rapidement les comparaisons de propriétés temporelles nécessaires pour calculer les relations temporelles.

Une autre facette de la mise en relation des données est traitée dans l'état de l'art par l'appariement d'entités (*entity resolution*) (Shen *et al*., 2015). Il s'agit d'associer entre elles des entités équivalentes, ce qui a un enjeu dans des domaines comme les bases de données relationnelles, la recherche d'information ou encore l'annotation de textes. Plus généralement, la découverte de liens (*entity linking*) vise à trouver des liens sémantiques entre des entités issues de différentes bases de connaissances (Auer *et al*., 2011 ; Smeros, Koubarakis, 2016). Selon (Smeros, Koubarakis, 2016), les approches de l'état de l'art se concentrent sur la recherche d'équivalence entre les entités (mêmes étiquettes, mêmes noms ou mêmes types), laissant d'autres types de relations, par exemple les relations spatiales ou temporelles, inexploitées. Ces auteurs proposent donc d'utiliser les liens spatio-temporels pour calculer plus de relations. Or la représentation spatiale de la plupart des données géo-localisées est complexe, sous forme de polygone. Le calcul des relations entre polygones au sein de très grands jeux de données est particulièrement complexe et long. Une étape de pré-traitement est nécessaire pour transformer les données (issues de vocabulaires RDF, de CRS, de sérialisations, etc.) selon un modèle unique. Ensuite, une technique de *blocking* vise à réduire la complexité du calcul. Elle consiste à découper en "blocs" (rectangles incurvés) la surface terrestre, puis à évaluer les relations topologiques entre entités en se basant sur ce découpage. De même, Sherif et ses collègues (Sherif *et al*., 2017) proposent de découvrir des liens topologiques encore plus efficacement grâce à une indexation des entités à l'aide de tuiles découpant la surface terrestres en rectangles. Cette méthode accélère le calcul de relations topologiques entre deux géométries d'entités dans le plan.

3. Une approche sémantique pour l'intégration spatiale et temporelle de données d'observations de la Terre

3.1. Principes de l'approche d'intégration

Reprenant certains des principes de cet état de l'art, nous proposons une approche sémantique pour l'intégration de données d'observations de la Terre qui s'appuie sur

leurs propriétés spatiales et temporelles. Nous nous intéressons en particulier à des données géo-localisées tirées de sources ouvertes et aux méta-données d'images satellites. Comme (Atemezing, 2015), nous proposons une ontologie, à savoir un vocabulaire formel qui étend des vocabulaires standards présents sur le LOD pour mieux représenter ces données comme des entités associées à des classes et possédant des propriétés spatiales (une géométrie) et temporelles (une datation). Pour intégrer ces données, nous nous appuyons d'abord sur leur dimension spatiale et comme (Smeros, Koubarakis, 2016), nous avons recours à la notion de tuilage pour réduire les coûts de calcul des relations spatiales entre entités représentant les données et les images. Cependant, nous avons choisi de nous limiter aux relations spatiales définies par GeoSPARQL afin d'utiliser ce langage pour interroger les données. Dans un deuxième temps, l'intégration prend en compte les propriétés temporelles des données pour associer les données pertinentes par rapport à la prise de vue d'une image.

Ce travail a été réalisé dans le cadre du projet SparkinData [10] visant à construire une plate-forme cloud de données d'observations de la Terre et à valoriser les images de la collection Sentinel-2. Nous avons évalué notre approche grâce à un cas d'étude où nous exploitons les dimensions spatiales et temporelles pour lier les méta-données d'images avec les unités administratives publiées sur le LOD de l'INSEE et des données météorologiques fournies par MétéoFrance. Nous rendons accessibles ces données via un point d'accès [11] SPARQL.

3.2. Architecture

Cette approche d'intégration est mise en oeuvre au sein d'une plate-forme dont l'architecture est modulaire (Figure 1). Ses différents niveaux permettent de découpler les étapes du processus permettant de passer des données brutes aux données sémantisées. Elle est composée des modules suivants :

– **Sélection des données** : la première étape du processus d'intégration des données est l'identification et l'accès aux sources de données à collecter. Un ensemble de données est soit un fichier, soit le résultat d'une requête d'interrogation d'un entrepôt de données. Les formats traités pour le moment sont CSV, JSON, RDF, XML, GeoTIFF, et Shapefile. Les sources de données utilisées sont décrites en Section 5.1.

– **Conversion des données** : Les données des sources sélectionnées sont dans un premier temps converties dans une représentation pivot en JSON. Pour cela, nous avons soit réutilisé des scripts dédiés soit développé nos propres scripts, selon le type de données de la source considérée. Les fichiers JSON intermédiaires sont stockés dans une base de données MongoDB comme sauvegarde de sécurité. Des exemples de conversion de données sont présentés en Section 5.2.

10. SparkInData fait partie du programme français d'Investissement d'Avenir (FUI).
11. http://melodi.irit.fr/sparkindata/

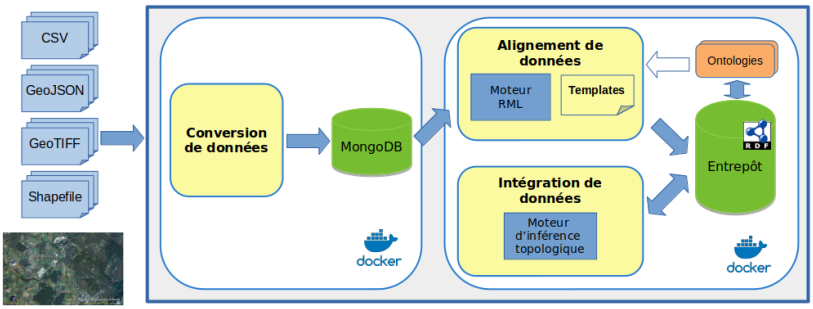


FIGURE *1. Architecture d'intégration des données issues de sources hétérogènes.*

– **Alignement des données** : Les données des fichiers JSON sont transformées en instances de classes de l'ontologie présentée en Section 4. Nous avons défini pour cela un template d'alignement (i.e. un modèle RDF de triplets à produire) et implémenté un mécanisme de traitement au sein d'un module Python. Nous fournissons dans la Section 5.2, des exemples de templates. A partir des valeurs présentes dans les documents JSON, les fonction du module Python réalisent des opérations sophistiquées qui ne sont pas possibles dans les approches alternatives telles que RML.

– **Intégration des données** : Le processus d'intégration s'appuie sur les relations topologiques, soit spatiales, soit temporelles, entre les instances des classes du modèle. A ce stade, il est possible de calculer les relations topologiques entre toutes les instances ayant une représentation spatiale, et de les stocker comme des assertions dans le triplestore. Une alternative est d'évaluer les relations topologiques à la volée, notamment les relations temporelles. Dans ce cas, pour réduire le temps et le coût de calcul, on peut opérer une sélection des relations à considérer ou des instances à lier.

4. Un modèle pour l'intégration de données d'observations de la Terre

4.1. Vocabulaires réutilisés

Notre modèle ontologique pour l'intégration des données s'appuie sur deux vocabulaires existants, l'ontologie noyau SOSA et l'ontologie GeoSPARQL. Dans un de nos précédents travaux (Arenas *et al.*, 2016), nous avions utilisé respectivement DCAT et SSN pour représenter les enregistrements de méta-données et les données météorologiques. Désormais, nous adoptons SOSA comme ontologie noyau pour ces deux types de données.

SOSA est une ontologie légère, indépendante, représentant les classes et propriétés élémentaires de SSN (Semantic Sensor Network). SOSA décrit des capteurs et leurs observations, les procédures mises en oeuvre, les éléments d'intérêt étudiés, les échantillons utilisés, et les propriétés mesurées. SOSA est pertinent pour une vaste gamme d'applications, dont l'imagerie satellite. Nous l'avons donc adoptée pour décrire les

méta-données d'image comme des *observations de la Terre* (instances de *EarthObservation*) et les observations météo comme des *observations météo* (instances de *MeteoObservation*) (Figure 2). Nous avons néanmoins spécialisé SOSA pour mieux typer les instances de ces concepts, même si la tendance dans les domaines où SOSA a été adopté, comme l'IoT, est d'éviter ce type construction et d'utiliser directement SOSA comme vocabulaire principal (Pomp *et al*., 2017).

GeoSPARQL, un standard de l'OGC, définit une petite ontologie pour représenter des caractéristiques, des relations et des fonctions spatiales (Kolas *et al*., 2013) (Battle, Kolas, 2012). Il existe des alternatives à GeoSPARQL comme GeoRDF qui permet de représenter des données simples telles que la latitude, la longitude, l'altitude, comme des propriétés de points (en utilisant WGS84 comme référentiel), ou encore GeoOWL qui permet d'exprimer des objets plus complexes (lignes, rectangles, polygones). Nous avons retenu GeoSPARQL qui permet de raisonner sur des géométries, et de proposer ainsi des relations (inclusion, recouvrement, etc.) entre des entités sur la base des relations entre leurs géométries.

4.2. Un modèle étendu pour l'intégration de données aux méta-données d'images

Le modèle ontologique que nous proposons est détaillé sur la Figure 2 et est organisé en modules. Il intègre certaines classes et propriétés (en particulier les propriétés temporelles) de SOSA (module `sosa` qui réutilise la classe `time:TemporalEntity` du vocabulaire OWL Time), ainsi que des classes et propriétés de GeoSPARQL (module `geo`) comme `SpatialObject`, `Feature` et `Geometry`. Ce modèle comporte aussi des classes et propriétés spécifiques à notre modèle. Deux modules sont dédiés à la représentation des images d'observation de la Terre : `eom` pour les méta-données d'images (qui spécialise `sosa` et `geo`) et `grid` pour représenter les tuiles (qui spécialise `geo`). Ensuite, il convient de définir des classes pour décrire chaque jeu de données que l'on souhaite intégrer, et de les relier aux classes de `geo` et si besoin de `sosa`. Dans notre cas d'étude, les classes de `mfo` (qui spécialise `sosa`) servent à représenter les données météorologiques et les stations météo, alors que `admin` permet de représenter des unités administratives.

Toute instance de la classe `sosa:Observation` possède une dimension temporelle. Nous définissons donc un enregistrement de méta-données d'image par la classe `eom:EarthObservation` qui spécialise `sosa:Observation`. Sa dimension temporelle identifie le moment où l'image a été prise. De même, les stations météo enregistrent périodiquement des mesures. Nous définissons donc la classe `mfo:MeteoObservation` comme une sous-classe de `sosa:Observation` pour représenter les données mesurées par une station météo. Nous pouvons alors lier par une relation temporelle (before, after) un enregistrement de méta-données d'image et des mesures météo ou mémoriser des périodes d'intérêt (e.g., une semaine après la prise de l'image).

Pour représenter la géo-localisation des images, le modèle s'appuie sur leur emprise (ou footprint), qui est un polygone fermé (une géométrie) correspondant à la

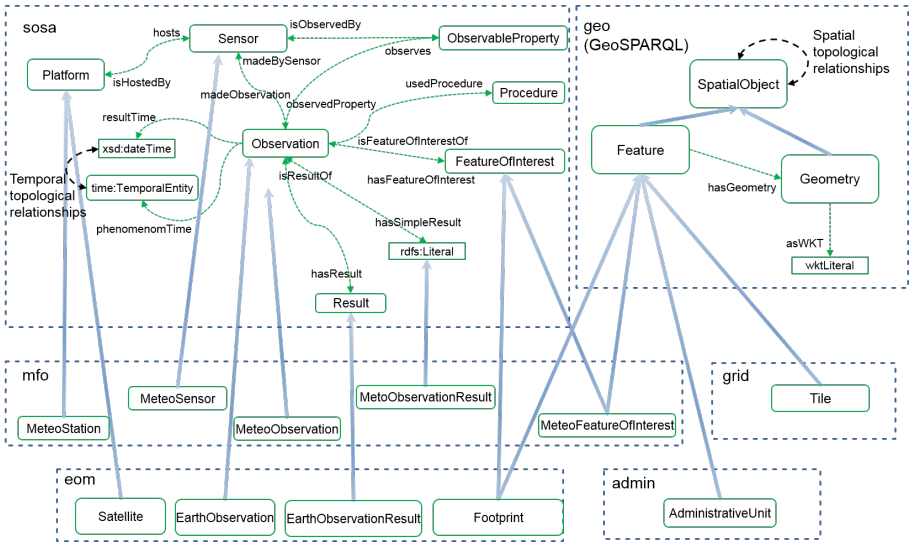


FIGURE 2. *Le modèle d'intégration. SOSA et GeoSPARQL sont spécialisés dans 4 modules dédiés à chaque source de connaissances et au tuilage des images.*

zone géographique couverte par l'image. Elle est représentée à l'aide des classes `eom:Footprint` et des tuiles, représentées comme des `grid:Tile`. Ces deux classes sont définies comme des spécialisations de `geo:Feature`. `eom:Footprint` spécialise aussi `sosa:FeatureOfInterest`.

De même, les données à intégrer sont géo-localisées, et donc définies comme des sous-classes de `geo:Feature`. C'est le cas des données météo via la classe `mfo:MeteoFeatureOfInterest` et des données sur les unités administratives `admin:AdministrativeUnit`. Images et données météo peuvent aussi être liées par des relations spatiales calculées à partir des coordonnées `geo:wktLiteral` de leur `geo:Geometry`. Pour cela, nous utilisons les relations topologiques (contient, recouvre, etc.) proposées par GeoSPARQL. Ces relations associent deux ressources (deux `geo:Geometry` ou deux `geo:Feature`) grâce aux propriétés topologiques (propriétés directes) ou à des fonctions topologiques (propriétés calculées).

Au sein du modèle `mfo`, une station météo est représentée comme une instance de la classe `mfo:MeteoStation` alors que la position géographique des stations est une propriété `hasPosition` (non mentionnée sur la figure) ayant respectivement pour domaine et co-domaine les classes `mfo:MeteoStation` et `geo:Feature`. Les capteurs fonctionnant sur une station météo sont représentés comme des instances de `mfo:MeteoSensor`, sous-classe de `sosa:Sensor`. La position géographique d'une station est représentée comme instance de la classe `mfo:MeteoFeatureOf-Interest`, une sous-classe de `sosa:FeatureOfInterest`. `mfo:MeteoFea-tureOfInterest` est aussi une sous-classe de `geo:Feature`. Ainsi, connaissant

la position d'un `mfo:MeteoFeatureOfInterest`, il est facile d'identifier les caractéristiques d'un autre type qui recouvrent les observations météo.

Afin de lier les observation de la Terre à des unités administratives françaises (régions, départements et villes) à partir de leur position géographique (point ou polygone), nous avons enrichi le modèle avec la classe `admin:AdministrativeUnit`, une sous-classe de *geo:Feature*. Enfin, pour les images Sentinel 2 Single Tile (S2ST), les tuiles correspondent aux *features of interest* des images. Un S2ST correspond à un fragment de l'image originale d'une taille approximative de 100 x 100 km. L'intérêt par rapport à une image S2 normale est que l'utilisateur peut sélectionner la surface qui l'intéresse et ne télécharger que l'information souhaitée [12]. Nous avons également enrichi le modèle avec les classes principales de *Global Land Cover* : `Artificial Surfaces`, `Cropland`, `Tree Covered Areas`, etc.

5. Sélection, conversion et alignement de données

5.1. Sélection des données

La finalité du processus étant l'intégration de données via des relations spatiales ou temporelles, les propriétés requises pour calculer ces relations (localisation, datation) doivent être accessibles. Selon le cas, ces données sont fournies avec la source à intégrer ou bien mémorisées à part. Dans ce dernier cas, il est nécessaire de recourir à des sources de données complémentaires. Nous distinguons les sources de données dynamiques, pour lesquelles la dimension temporelle est importante (comme c'est le cas des données de capteurs), des sources de données statiques, pour lesquelles seule la dimension spatiale est requise dans notre processus.

Les sources des données dynamiques. Dans le projet SparkInData nous utilisons des enregistrements de méta-données d'images Sentinel [13]. La périodicité de Sentinel-1 est de douze jours, tandis que celle de Sentinel-2 est de cinq jours. Les enregistrements de méta-données sont obtenus au format GeoJSON (format JSON pour encoder des données géospatiales) à partir de l'API RESTO, un service de données géré par le CNES (Gasperi, 2014). L'URL suivante par exemple retourne tous les enregistrements de méta-données de la collection Sentinel-2 Single Tile pour la France, réalisés entre le 19/09/2017 23:00 et le 25/09/2017 00:00 :

https://peps.cnes.fr/resto/api/collections/S2ST/search.json?q=France\&startDate=17-09-19T23:00:00\&completionDate=2017-09-25T00:00:00.

Les requêtes faites avec cette API peuvent spécifier les paramètres à retrouver, i.e. des métadonnées spécifiques comme la couverture nuageuse, l'intervalle de temps, la zone géographique d'intérêt, etc. Nous collectons ces informations toutes les nuits.

12. Un fichier S2ST est moins volumineux : il peut faire environ 500Mo alors que celui d'une image Sentinel-2 avant tuilage peut faire plus de 3Go.

13. https://sentinel.esa.int/web/sentinel/missions/ (07/2016)

Les données contextuelles que nous utilisons sont les informations météo fournies par *SYNOP Meteo France* [14] sous forme de fichiers CSV zippés. Les observations sont prises toutes les trois heures dans chacune des 62 stations françaises. Un fichier contenant la liste des stations avec leur position respective, i.e. un point fixe repéré par ses coordonnées géographiques, est fourni séparément.

Les sources des données statiques. Pour les images S2ST, des informations sur la couverture spatiale de l'image sont obtenues à partir des méta-données sous deux formes : 1) le *footprint* de l'image, 2) l'identifiant de la tuile qui lui correspond. Le fichier grid KML qui indique l'emprise de chaque tuile ainsi que son nom est fourni par l'ESA [15]. Nous avons donc traité ce fichier.

Une autre source de données que nous avons exploitée est le GLC-SHARE (Global Land Cover SHARE) produit par le FAO, qui donne des informations sur la couverture terrestre. Elle s'appuie sur une nomenclature qui classe les zones en fonction du type d'occupation des sols ou du type de surface ; 11 classes sont définies telles que surface artificielle (01), terre cultivée (02), zone forestière (03), etc. Les données du GLC-SHARE sont fournies sous forme d'image au format *GeoTIFF* (format TIFF incluant des informations de géo-référencement) dont chaque pixel correspond à une surface d'environ 1 km^2. La valeur d'un pixel est un entier indiquant la classe la plus fréquente pour la zone couverte par le pixel. Nous avons exploité cette source pour associer des données aux tuiles des images S2ST. Nous avons ainsi calculé la composition de la couverture terrestre de chacune de ces tuiles : sous forme d'un pourcentage des différentes classes GLC-SHARE sur la surface couverte par la tuile.

Finalement, nous collectons des données RDF sur les unités administratives françaises à partir de la base de connaissances de l'INSEE [16]. Ces données n'étant pas géo-localisées, il n'était pas possible de les intégrer aux méta-données d'images. Nous avons donc utilisé la plate-forme française des données publiques "data.gouv.fr" [17] pour obtenir ces informations, accessibles au format shapefile.

5.2. Transformation et alignement des données

Nous venons de présenter les données que nous exploitons, leurs diverses sources et la variété de leurs formats originaux. Afin de standardiser les traitements, nous représentons toutes ces données au format JSON puis nous les convertissons en RDF à l'aide de mécanismes d'alignement. Toutefois, certaines données, par exemple la couverture terrestre ou les données météo, sont aussi exploitées en amont du processus de transformation RDF pour produire d'autres données (moyennes, etc.) adaptées aux besoins. Nous décrivons à présent chacun de ces processus.

---

14. https://donneespubliques.meteofrance.fr/ (07/2016)
15. https://sentinel.esa.int/web/sentinel/missions/sentinel-2/news/-/article/sentinel-2-tiling-grid-updated
16. http://rdf.insee.fr/def/index.html
17. https://www.data.gouv.fr

Les unités administratives. Un langage classique pour convertir des données en RDF est RML (*RDF Mapping Language*), un langage de règles de transformation d'un format donné (e.g. JSON) en une représentation RDF. Une des limites de RML est qu'il ne permet pas de faire appel facilement à des fonctions spécialisées pour traiter des informations spécifiques. Le document JSON suivant illustre ce défaut :

```
{"wkt": "MULTIPOLYGON(((
-1.0988062299633785 45.64032288975508, ...
-1.0988062299633785 45.64032288975508))...)",
"name": "Poitou-Charentes", "geomType": 5,
"inseeInfo": {"adminType": "region", "insee": "54"}}
```

Ce code décrit une unité administrative française comme un ensemble d'attributs et de valeurs. La valeur de l'attribut `wkt` est la géométrie codée en WKT (Well known text), et la clé `name` est une chaîne de caractères donnant le nom de cette unité ("Poitou-Charentes"). La clé `inseeInfo` contient des informations en référence à l'identification INSEE de cette unité. A partir de la valeur de `inseeInfo`, on peut récupérer l'URI de l'unité administrative dans la base de connaissances de l'INSEE. Ceci nécessite tout de même de créer une requête SPARQL pour interroger la base de données de l'INSEE, ce qui ne peut pas être réalisé avec une règle RML.

Tout en conservant une approche similaire à RML, nous avons développé une solution alternative et mieux adaptée pour transformer le JSON en RDF. Cette solution comprend un template de triplets et un processeur codé en Python. Le code suivant est un exemple de template qui transforme l'extrait de document JSON montré plus haut en RDF à l'aide du vocabulaire `admin` de l'ontologie.

```
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix admin: <http://melodi.irit.fr/ontologies/administrativeUnits.owl#> .
# ce template definit la structure d'une unite administrative
<dummy> a getUrlAdministrativeUnitType(\$.inseeInfo.adminType) .
<dummy> admin:hasInseeCode stringToLiteral(\$.inseeInfo.insee) .
<dummy> admin:hasName stringToLiteral(\$.name) .
# representation spatiale de l'unite administrative
<dummy> a geo:Feature .
<dummy> geo:hasGeometry <dummy_geo> .
<dummy_geo> a geo:Geometry .
<dummy_geo> geo:asWKT valueToWktLiteral(\$.wkt) .
# l'instance est liee a l'unite administrative de l'INSEE
<dummy> owl:sameAs getInseeUrl(\$.inseeInfo) .
```

Le template est constitué de triplets dont les variables sont remplacées par les valeurs lues dans le document JSON. Pour les valeurs contenues qui nécessitent des traitements supplémentaires, nous avons développé des fonctions auxquelles nous passons en paramètres le chemin JSON vers les informations à extraire du fichier. En ce qui concerne *getInseeUrl($.inseeInfo)*, la fonction crée une requête SPARQL à partir des valeurs en paramètre, l'envoie au endpoint SPARQL, traite le résultat et retourne l'URI de l'unité administrative qui correspond à ces valeurs. Voici la requête SPARQL générée par la fonction *getInseeUrl()* pour cet exemple :

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX igeo:<http://rdf.insee.fr/def/geo#>
SELECT ?adminUnit  WHERE {
?adminUnit rdf:type igeo:Region .
?adminUnit igeo:codeINSEE "54"^^<http://www.w3.org/2001/XMLSchema#token> .}
```

Le graphe RDF résultant est fourni dans cet extrait :

```
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix admin: <http://melodi.irit.fr/ontologies/administrativeUnits.owl#> .
@prefix l_admin: <http://melodi.irit.fr/lod/administrativeUnit/> .
l_admin:region_54 a admin:Region .
l_admin:region_54 admin:hasInseeCode "54"^^xsd:String .
l_admin:region_54 admin:hasName "Poitou-Charentes"^^xsd:String .
l_admin:region_54 owl:sameAs <http://id.insee.fr/geo/region/54> .
l_admin:region_54 a geo:Feature .
l_admin:region_54 geo:hasGeometry l_admin:region_54_geo .
l_admin:region_54_geo geo:asWKT "MULTIPOLYGON(((
-1.0988062299633785 45.64032288975508, ...
-1.0988062299633785 45.64032288975508))...)"^^wkt:Literal .
```

Les observations météorologiques. Le caractère temporel des données météorologiques a une importance particulière. Les observations contenues dans l'entrepôt SYNOP ont plusieurs temporalités. Les observations codées `tminsol` représentent la plus petite température relevée durant les 12 dernières heures, alors que celles codées `t` correspondent à la température relevée au moment de la mesure. Selon notre approche il suffit d'implémenter des fonctions pour traiter la diversité temporelle de ces données. L'extrait de code JSON suivant représente une observation de type `tminsol` relevée par la station météo `07747` le `03/06/2017 à 3h`.

```
{    "temporalInfo" :
    {   "timeStamp" : 1512529200,
        "month" : "12",
        "day" : "06",
        "hour" : "03",
        "year" : "2017" },
    "tminsol" : 271.45,
    "numer_sta" : "07747" }
```

Pour traiter des observations SYNOP de Météo France nous avons écrit le template suivant qui appelle la fonction *getMFO_PhenomenonTime(doc)* :

```
<dummy> sosa:phenomenonTime getMFO_PhenomenonTime(doc) .
```

La fonction scanne le document JSON, trouve le type de l'observation à partir de sa clé (`tminsol`), et crée une instance de la classe `time:Interval` (spécialisation de `time:TemporalEntity`). Elle examine ensuite l'élément `temporalInfo` et calcule le début de l'intervalle de temps (la fin étant fournie par la clé `temporalInfo` elle-même). Début et fin de l'intervalle sont représentés comme des instances de `time:Instant`. Pour l'exemple précédent, le résultat est le suivant :

```
gmfo:Obs_07747_20171206030000_tminsol sosa:phenomenonTime
```

```
                        gmfo:TimeInterval_1512486000_1512529200 .
gmfo:TimeInterval_1512486000_1512529200 a time:TemporalEntity .
gmfo:TimeInterval_1512486000_1512529200 time:hasBeginning
                        gmfo:TimeInterval_1512486000_1512529200_beginning .
gmfo:TimeInterval_1512486000_1512529200_beginning time:inXSDDateTime
                        "2017-12-05T15:00:00+0100"^^xsd:dateTime .
gmfo:TimeInterval_1512486000_1512529200 time:hasEnd
                        gmfo:TimeInterval_1512486000_1512529200_end .
gmfo:TimeInterval_1512486000_1512529200_end time:inXSDDateTime
                        "2017-12-06T03:00:00+0100"^^xsd:dateTime .
```

La couverture terrestre. Pour intégrer cette source de données, nous avons dont l'API REST prend en entrée un polygone WKT en SRS EPSG:4326. A partir de ce polygone, il récupère les données originales dans un fichier temporaire, puis crée une table des fréquences. La réponse du serveur est un document JSON contenant le pourcentage de chaque classe de la couverture terrestre pour la zone délimitée par le polygone. Ce document JSON est ensuite transformé en RDF.

Les méta-données d'images satellites. Ces métadonnées sont obtenues sous forme de fichiers GeoJSON et transformée en RDF à l'aide d'un template particulier, selon le même principe en utilisant le vocabulaire `eom` (Section 4).

Les tuiles d'images. Les tuiles sont fournies dans le fichier `grid` de l'ESA que nous transformons en JSON, puis en RDF en utilisant le vocabulaire `grid` décrit dans la Section 4. Il est possible de calculer les relations topologiques entre les éléments spatiaux et les tuiles, puis d'extrapoler ces informations pour les images. Par exemple, sachant que les images $[img1, img2, img3]$ partagent la tuile $tile_1$, et que cette tuile recouvre $adminUnit_i$, il est possible d'inférer que $[img1, img2, img3]$ recouvrent aussi $adminUnit_i$.

5.3. Intégration des données

Intégration des données ayant une composante spatiale fixe. Les relations spatiales sont relativement stables dans le temps. Ainsi, les relations topologiques entre les grilles (SS2) et les unités administratives (l'image Y recouvre la région R), ou les informations sur la couverture terrestre associée à chaque cellule de la grille, sont calculées une fois pour toutes et mémorisées dans l'entrepôt RDF. Nous avons développé un script Python pour calculer les relations topologiques entre les instances des classes qui utilise la librairie *shapely* pour comparer les surfaces. Nous enregistrons ensuite les relations dans l'entrepôt sous forme de triplets dans lesquels le prédicat est une propriété topologique de GeoSPARQL.

Intégration des données ayant une composante temporelle. La mise en relation temporelle d'un enregistrement de méta-données d'image et de données ayant une propriété temporelle (date ou intervalle) prend en compte la période de temps qui intéresse l'utilisateur. Par exemple, il peut vouloir associer à une image à des informations météorologiques relevées une semaine après la prise de l'image. L'intervalle de temps défini par l'utilisateur joue le rôle de buffer temporel fournissant un contexte aux enregistrements de méta-données.

6. Conclusion

L'intégration de données d'observations de la Terre provenant de sources hétérogènes avec des métadonnées d'images satellites peut tirer profit des technologies du web sémantique. En publiant des ensembles de données et des méta-données d'images sous forme de LOD, l'accès aux observations de la Terre liées se trouve facilité, ce qui offre de nouvelles possibilités pour utiliser les images satellites dans une plus grande variété d'applications. De plus, pour les ensembles de données volumineux et dynamiques, en utilisant des requêtes SPARQL pour interroger conjointement des bases de données d'observations et des données liées, il est possible de créer des triplets RDF à la volée et ainsi éviter de convertir d'énormes ensembles de données en triplets RDF. Dans cet article, nous avons proposé un cadre pour intégrer des données spatiales. Nous avons conçu un vocabulaire pour représenter les données d'observations de la Terre et les méta-données d'images ; nous avons élaboré un processus de conversion RDF qui utilise des templates adaptés aux ressources et une librairie Python pour dépasser certaines limites de RML ; nous avons aussi proposé un processus d'intégration qui exploite la géométrie des données et GeoSPARQL pour lier les données géospatiales, et au final des requêtes SPARQL pour lier dynamiquement les données aux images à partir de leurs caractéristiques spatiales et temporelles. Dans la continuité de ces travaux, nous envisageons de considérer des sources propres à un domaine métier pour traiter un cas d'usage particulier (l'agriculture et des rapports bulletins agricoles) et fournir des règles et des fonctionnalités de raisonnement pour faciliter les analyses.

Bibliographie

Alirezaie M., Kiselev A., Längkvist M., Klügl F., Loutfi A. (2017). An ontology-based reasoning framework for querying satellite images for disaster monitoring. *Sensors*, vol. 17, n° 11.

Arenas H., Aussenac-Gilles N., Comparot C., Trojahn C. (2016). Semantic integration of geospatial data from earth observations. In *Knowledge engineering and knowledge management - EKAW 2016 satellite events*, p. 97–100. Bologna (It), Springer.

Atemezing G. A. (2015). *Publishing and consuming geo-spatial and government data on the semantic web*. Thèse de doctorat non publiée, Thesis. Consulté sur http://www.eurecom.fr/publication/4545

Auer S., Lehmann J., Ngonga Ngomo A.-C. (2011). Introduction to linked data and its lifecycle on the web. In *Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Ireland, August 23-27, 2011, Tutorial Lectures*, p. 1–75.

Battle R., Kolas D. (2012). Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web*, vol. 3, n° October 2012, p. 355–370.

Blázquez L. M. V., Saquicela V., Corcho Ó. (2012). Interlinking geospatial information in the web of data. In *Bridging the geographic information sciences - international agile'2012 conference*, p. 119–139. Avignon, France.

Blázquez L. M. V., Villazón-Terrazas B., Corcho Ó., Gómez-Pérez A. (2014). Integrating geographical information in the linked digital earth. *International Journal of Digital Earth*, vol. 7, n° 7, p. 554–575.

Brizhinev D., Toyer S., Taylor K., Zhang Z. (2017). *Publishing and using earth observation data with the rdf data cube and the discrete global grid system*. Rapport technique. W3C and OGC.

García-Rojas A., Athanasiou S., Lehmann J., Hladky D. (2013). Geoknow: Leveraging geospatial data in the web of data. In *Open data on the web.* Campus London, Shoreditch.

Gasperi J. (2014). Semantic Search Within Earth Observation Products Database Based on Automatic Tagging of Image Content. In *Proc. of the Conf. on Big Data from Space*, p. 4–6. ESA/ESRIN, Frascati, Italy., EU Publications.

Georgala K., Sherif M. A., Ngomo A. N. (2016). An efficient approach for the generation of allen relations. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI), Including Prestigious, Applications of Artificial Intelligence (PAIS)*, p. 948–956.

Heath T., Bizer C. (2011). *Linked data: Evolving the web into a global data space ; lectures on the semantic web: Theory and technology*. Morgan & Claypool.

Kolas D., Perry M., Herring J. (2013). *Getting started with GeoSPARQL*. Rapport technique. OGC. Consulté sur http://www.ssec.wisc.edu/meetings/geosp_sem/presentations/GeoSPARQL_Getting_Started-KolasWorkshopVersion.pdf

Koubarakis M., Karpathiotakis M., Kyzirakos K., Nikolaou C., Vassos S., Garbis G. *et al.* (2012). Building virtual earth observatories using ontologies and linked geospatial data. In *Web Reasoning and Rule Systems: 6th Int. Conf. RR, Vienna, Austria*, p. 229–233.

Pomp A., Paulus A., Jeschke S., Meisen T. (2017). ESKAPE: Platform for Enabling Semantics in the Continuously Evolving Internet of Things. In *2017 IEEE 11th International Conference on Semantic Computing*, p. 262-263.

Reitsma F., Albrecht J. (2005). Modeling with the semantic web in the geosciences. *IEEE Intelligent Systems*, vol. 20, n° 2, p. 86-88.

Shen W., Wang J., Han J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, n° 2, p. 443-460.

Sherif M. A., Dreßler K., Smeros P., Ngomo A. N. (2017). Radon - rapid discovery of topological relations. In *Proceedings of the thirty-first AAAI conference on artificial intelligence, feb. 4-9, 2017, san francisco, cal., USA.*, p. 175–181.

Smeros P., Koubarakis M. (2016). Discovering spatial and temporal links among RDF data. In *Proceedings of the workshop on linked data on the web, LDOW 2016, co-located with 25th international world wide web conference (WWW 2016).*

Sukhobok D., Sánchez H., Estrada J., Roman D. (2017). Linked data for common agriculture policy: Enabling semantic querying over sentinel-2 and lidar data. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks.*

Tandy J., Brink L. van den, Barnaghi P. (2017). *Spatial data on the web best practices, w3c working group note*. Rapport technique. W3C and OGC. Consulté sur https://www.w3.org/TR/sdw-bp/

L'influence de la gravité des données dans les architectures des lacs de données

Cédrine Madera[1], Anne Laurent[2], Thérèse Libourel[3], André Miralles[4].

1. *IBM & LIRMM, Montpellier, France*
   cedrinemadera@fr.ibm.com
2. *Université de Montpellier LIRMM, Montpellier, France*
   anne.laurent@umontpellier.fr
3. *UMR Espace-Dev (UM, IRD, UG, UA, ULR), Université de Montpellier*
   therese.libourel@umontpellier
4. *UMR Tetis/IRSTEA, Maison de la télédétection, Montpellier, France*
   andre.miralles@teledetection.fr

RESUME. La révolution digitale qui met au cœur de sa stratégie la donnée fait émerger le concept de lac de données. Celui-ci devient un composant incontournable pour la découverte de l'information potentiellement enfouie dans les données. Nombre d'industriels qui s'engagent sur cette voie recourent de plus en plus à l'intégration de lacs de données dans leur système d'information et utilisent le plus souvent une plateforme fédératrice, reposant sur la technologie open source « Apache Hadoop ». Cette approche purement industrielle mono technologie commence à trouver ses limites. Dans cet article, nous nous intéressons, d'un point de vue académique, à l'hypothèse de la remise en cause de cette mono technologie par divers facteurs, dont ceux liés à la gravité des données. Nous illustrons notre hypothèse par un cas d'usage en milieu industriel.

ABSTRACT. The digital revolution that puts the data at the heart of its strategy brings out the new concept of data lake. It becomes an essential component for the discovery of information potentially hidden in data. Many practitioners who commit to this pat rely heavily on the integration of data lakes in their information system and most often use a unifying platform, based on an open source technology "Apache Hadoop". This unique industrial approach begins to find its limits. We are interested, from an academic point of view, in the hypothesis of the questioning of this mono technology by various factors, including those related to the data gravity. We illustrate our hypothesis with a use case in industrial environment.

MOTS-CLES : lac de données, gravité des données, architecture informatique, système d'information, déplacement de données, duplication de données

KEYWORDS: data lake, data gravity, architecture, information system, data migration, data duplication

1. Introduction

L'internet des objets associé à la production, sans cesse croissante, de données émises dans les systèmes d'information traditionnels conduit à une accumulation de données disponibles sans précédent. Ces données disponibles, si elles attisent les convoitises en vue d'en tirer une richesse en termes d'information notamment, posent des questions quant à leur conservation, leur pertinence, leur mise en forme, leur gouvernance mais surtout leur capitalisation en vue de valorisation ultérieure.

La capitalisation de cette « richesse » est un enjeu majeur des systèmes d'information d'aujourd'hui mais aussi de demain. Le lac de données (*Data Lake*) est un nouveau composant du système d'information (Madera et Laurent, 2016), il met au cœur de sa conception la donnée et non l'information à délivrer, complétant les autres composants existants, tels que les systèmes décisionnels. Un de leur principal objectif est de permettre l'exploration et l'analyse de sources de données diverses afin de trouver de nouveaux « modèles » (*« pattern »*) d'information.

Dans le contexte industriel, les architectes d'information[1] ont en charge la gouvernance, la conception et l'outillage technologique de ces lacs de données. L'architecture d'information qu'ils doivent mettre en place doit prendre en compte conjointement des contraintes fonctionnelles (*cf.* section 2) mais aussi non fonctionnelles. De par l'important volume de données à traiter, la diversité des formats de ces données mais aussi leur coût considéré comme peu élevé, la technologie de type *Apache Hadoop* s'est imposée comme référente, quasi unique, occultant les discussions sur l'impact des contraintes non fonctionnelles sur ce choix. Cette technologie comme solution unique pour les lacs de données est désormais remise en cause (Russom, 2017) par le monde industriel et des architectures hybrides, avec introduction de technologie complémentaire à *Apache Hadoop*, sont désormais envisagées.

Notre intérêt académique se porte sur les facteurs pouvant influencer cette hybridation technologique mais aussi applicative. Les travaux de (McCrory, 2010 ; Alrehamy et Walker, 2015) ont introduit la notion de gravité des données qui, du point de vue de l'architecture, peut être considérée comme une contrainte non fonctionnelle. Partant de cette idée, un des objectifs de nos recherches en cours consiste à compléter la définition de (McCrory, 2010) afin d'étudier son impact sur les architectures des lacs de données. Nous vérifions nos hypothèses au travers de l'analyse d'un cas réel d'architecture d'un lac de données en milieu industriel.

La suite de cet article se présente de la manière suivante. Dans la section 2, nous rappelons les notions de lac de données et de gravité des données. La section 3 est

[1] L'architecte d'information se concentre sur les éléments requis pour structurer les aspects informationnels et données des solutions retenues et pour concevoir, construire, tester, installer, exploiter et maintenir le système d'information de la solution. Pour mener à bien sa mission, l'architecte système d'information doit en premier lieu étudier les besoins fonctionnels, établir une cartographie du système en analysant l'existant, puis proposer un modèle d'architecture et enfin la mettre en œuvre en choisissant une infrastructure matérielle et logicielle.

consacrée l'impact de la prise en compte de la gravité des données sur l'architecture des lacs de données. Dans la section 4 nous présentons notre étude expérimentale, au travers du cas d'usage industriel d'un lac de donnée dédié à la collecte de données de métrologie[2] d'un parc informatique. Nous concluons et présentons quelques perspectives dans la section 5.

2. Introduction des concepts de lac de données et de gravité des données

2.1 Les lacs de données

Nous considérons les lacs de données comme un nouveau composant du système d'information, qui se positionne en complément des systèmes décisionnels existants tels que les entrepôts de données. Leur principal objectif est de permettre la capitalisation des données d'une organisation, dans leur format le plus brut, afin d'en extraire de la valeur et permettre une valorisation du capital données de l'organisation. Dans nos précédents travaux (Madera et Laurent, 2016), nous en avons donné la définition suivante :

Le lac de données est une collection de données, non transformées, de formats non contraints (tous formats acceptés), conceptuellement rassemblées en un endroit unique mais potentiellement non matérialisé, destinées à un/des utilisateurs experts en science de données, munie d'un catalogue de méta-données ainsi que d'un ensemble de règles et méthodes de gouvernance de données.

Les lacs de données sont très souvent associés à la technologie de type *Apache Hadoop* (MarketsAndMarkets, 2016), ce qui, en termes d'architecture, peut limiter les solutions à explorer. En effet, le choix de solutions pour supporter les architectures des lacs de données si il se cantonne au champ des solutions basées sur *Apache Hadoop* peut certes simplifier le champ d'investigation mais il peut aussi occulter certaines problématiques générées par ce choix, voire oublier de répondre à certaines contraintes des lacs de données. Ces contraintes sont notamment la sensibilité des données que le lac de données souhaite collecter. La volumétrie en est une autre tout comme le coût de déplacement de ces données vers le lac. Ces contraintes peuvent remettre en cause une solution de collecte massive de données et imposer plutôt une approche où les données ne sont pas déplacées.

Sans remettre en cause la création d'environnement collecteur de données, nous souhaitons dans cet article étudier quels facteurs pourraient influencer, voire remettre en cause les architectures des lacs de données où toutes les données sont déplacées physiquement vers un ou plusieurs environnements de stockage composant les lacs de données. La gravité est un de ces facteurs.

---

[2] Dans le cadre d'un parc informatique (réseaux, serveurs, baies de stockages, etc.), l'objectif de la métrologie est de connaître et de comprendre le fonctionnement du parc informatique afin de pouvoir, non seulement intervenir dans l'urgence en cas de problème, mais aussi d'améliorer les performances, d'anticiper son évolution et sa planification.

2.2 La gravité des données

Par analogie de raisonnement entre la gravitation en sciences physiques[3], les données s'accumulent avec le temps, et peuvent être considérées comme plus denses ou avoir une plus grande masse. Lorsque la densité ou la masse croissent, l'attraction gravitationnelle des données augmente. Les services et applications ont leur propre masse et, par conséquent, ont leur propre gravité ; mais les données sont beaucoup plus volumineuses et plus denses qu'eux. Ainsi, alors que les données continuent d'augmenter, les services et les applications sont plus susceptibles d'être attirés par les données, plutôt que l'inverse. Cela ressemble, par mimétisme avec la gravité au sens physique, à l'exemple de la pomme qui tombe sur la Terre plutôt que l'inverse parce que la Terre a plus de masse que la pomme

Les travaux de McCrory (McCrory, 2010) sont les premiers à avoir exposés cette l'analogie entre la gravité de la donnée et la gravité au sens physique, en définissant la force d'attraction entre données et traitements. Dans cette analogie interviennent la masse des données, la vitesse de déplacement de ces données et les traitements/services qui y sont associés. La loi de la gravité stipule que l'attraction entre les objets est directement proportionnelle à leur masse. Dave McCrory (McCrory, 2014) a réutilisé le terme gravité des données pour décrire le phénomène dans lequel le nombre ou la quantité et la vitesse à laquelle les services, les applications, et même les clients sont attirés par les données, augmentent à mesure que la masse des données augmente. Le phénomène de gravitation peut alors être appliqué. Les données qui voient leur gravité augmenter vont attirer les traitements à elles. La force d'attraction exercée par les données sur les traitements ouvre la porte à d'autres paramètres pouvant influencer cette gravité tels que la sensibilité, le trafic du réseau, le coût, etc.

(Alrehamy et Walker, 2015), s'appuyant sur les travaux (McCrory, 2010), mettent en exergue cette gravité des données dans leur lac de données fonctionnellement dédié aux données personnelles. Cependant, leur évaluation de la gravité des données, dans ce cas d'étude où la masse des données s'avère peu importante, les amène à penser, que le paramètre le plus influent n'est pas la masse mais la sensibilité des données ; c'est elle qui va « peser » le plus dans l'évaluation de la gravité des données. Dans leur lac de données dédié aux données personnelles, celles-ci ont une sensibilité si forte que, ce lac attire à lui les traitements devant manipuler ces données. Dans ces travaux, c'est la sensibilité, un des paramètres que les auteurs ont inclus dans la gravité des donnés, qui influence l'attraction du traitement vers les données.

Ces premiers travaux intégrant la prise en compte de la gravité des données via volume (ou la masse) et sensibilité des données tendent à prouver l'influence qui peut s'exercer sur la relation donnée-traitement.

---

[3] En sciences physiques, la gravitation désigne la force qui fait que deux masses s'attirent mutuellement, comme la Terre et le Soleil. La gravité en est le résultat. C'est ce qui fait tomber les objets, comme la pomme tombée d'un arbre observée par Newton.

Il convient donc de regarder, en se basant sur ces travaux et cette analogie physique-gravitation, quelle pourrait être l'influence de la gravité des données dans les architectures des lacs de données.

Les architectures des lacs de données sont basées sur l'acquisition de données, sur lesquelles les traitements d'exploration et d'analyse (par exemple) vont s'appliquer. Il n'est pas envisagé, *a priori*, que les traitements des lacs de données se déplacent vers la donnée et que les données ne migrent pas, au sens technique du terme, vers la plateforme du lac de données.

Notre hypothèse est que lorsque l'on prend en compte la gravité des données, le traitement des données du lac peut être amené à se déplacer là où résident les données et non pas le contraire. En architecture, les paramètres qui composent cette gravité, tels que le volume (ou la masse) et la sensibilité, sont considérés comme des éléments de contraintes non fonctionnelles. Cela nous amène à considérer la gravité des données comme étant une contrainte non fonctionnelle[4] à l'évaluer lors de la conception des lacs de données. Par habitude de conception, fortement liée à des contraintes technologiques dans les systèmes d'information classiques, les données sont toujours déplacées vers les traitements qui les utilisent, ceci afin de protéger notamment les performances des systèmes opérationnels les émettant. La vision précédente de la gravité des données peut remettre en cause ce postulat. Sur cette voie, nous nous sommes donc attachés à définir quels paramètres non fonctionnels sont pertinents dans les lacs de données relativement au problème de la corrélation entre gravité et transferts données-traitements. Trois ont retenu notre attention : le volume (ou la masse), le coût et la sensibilité.

La ***masse*** des données disponibles devient de plus en plus importante, et une solution basique comme augmenter simplement la capacité de stockage ne suffit plus à répondre à cette problématique. Le ***coût***, lié la plupart du temps aux problématiques de réplication, d'acquisition, de sécurité mais aussi d'extension de capacité de stockage, doit être désormais évalué lors de la conception des architectures des lacs de données. La ***sensibilité*** des données entraîne une gestion spécifique. S'il n'existe pas de définition légale pour les données dites sensibles, les nouvelles règlementations sur la donnée personnelle RGPD[5] par exemple ou bien la cybersécurité et la Loi de Programmation Militaire (LPM) étendent celle déjà délivrée par la CNIL[6]. Chaque organisation peut, en plus de ces obligations réglementaires définir sa propre classification de données dites sensibles. Afin d'englober toutes ces notions, nous

[4] On appelle contrainte non fonctionnelle, les contraintes auxquelles sont soumises les architectures pour délivrer un fonctionnement correct, telles que la performance, le volume, la sécurité, l'évolution d'échelle, la disponibilité, la fiabilité, etc.

[5] Règlement General de la Protection des Données, une nouvelle réglementation européenne qui entrera en vigueur le 25 mai 2018

[6] http://www.cil.cnrs.fr/CIL/spip.php?rubrique300, Les données sensibles sont celles qui font apparaître, directement ou indirectement, les origines raciales ou ethniques, les opinions politiques, philosophiques ou religieuses ou l'appartenance syndicale des personnes, ou sont relatives à la santé ou à la vie sexuelle de celles-ci.

définirons qu'une donnée est dite sensible au regard du degré de sécurité criticité/précaution que nécessitent son utilisation et son traitement. Cela va donc dépendre de l'évaluation par le propriétaire de ces données.

Ces trois paramètres constituent des contraintes qui peuvent avoir des impacts très forts sur la conception des composants des systèmes d'information et peuvent réduire les choix d'architecture possibles voire remettre en cause un choix existant.

Nous proposons d'enrichir la vision des travaux de recherche précédents et d'affiner le concept de gravité de la donnée en le fondant sur les trois paramètres volume/masse, coût et sensibilité.

3. Impact de la gravité de la donnée sur les architectures des lacs de données

Dans le cadre des lacs de données, il semble que le postulat de déplacement de la donnée vers son traitement (stipulé dans la section 2.1) ait été adopté comme une pratique par défaut, ce qui se traduit par à une duplication, systématique, de toutes les données que l'on veut analyser et explorer. Or nous pensons que, dans ce postulat, la notion de gravité des données n'est pas étudiée, lors de la définition de l'architecture d'un lac de données. L'approche de duplication systématique de toutes les sources de données ne doit pas être l'approche de référence. Dans cet article, nous souhaitons étayer ce point et démontrer que la gravité des données peut jouer un rôle important dans ces architectures et doit être évaluée dès la conception.

3.1 L'impact du volume sur les lacs de données

L'augmentation du volume des données produites et donc de leur masse est l'un des paramètres de la gravité de la donnée. Si cette masse devient trop importante, d'après (McCrory, 2010), la gravité de la donnée va être telle que le traitement des données va être attiré vers elles et donc va donc devoir être déplacé. Le volume est intégré et pris en compte au niveau de la gouvernance du lac de données (cycle de vie des données) et doit donc être évalué finement lors de la conception de l'architecture fonctionnelle du lac de donnée. L'évaluation doit prendre en compte non seulement le volume des données intégrées dans le lac mais aussi prévoir une augmentation ultérieure de celui-ci. En effet, le lac de données a pour vocation de stocker les données le plus brutes possibles mais aussi, en fonctionnement courant, des données préparées, agrégées, archivées, etc. Ces différents états des données vont eux aussi influencer le volume du lac de données, et donc la masse. Les lacs de données doivent dès la conception intégrer cette notion fondamentale de cycle de vie des données qui est l'une des principales fonctionnalités de la gouvernance des données. Il peut être décidé que le volume de données généré va être tel qu'elles ne peuvent pas être déplacées physiquement ou dupliquées mais être seulement accessibles.

3.2 L'impact de la sensibilité sur les lacs de données

La notion de donnée sensible est elle aussi une contrainte qu'il faut évaluer dans les lacs de données. En effet certaines données au regard de leur « sécurité », de leur « sensibilité » ne peuvent pas être dupliquées ou déplacées. L'anonymisation ou la pseudonymisation des données sont des techniques qui permettent de manipuler et déplacer ces données en respectant leur sensibilité. Cependant ces techniques peuvent faire perdre la valeur même des données qui ne sont alors plus exploitables. L'encryptage est aussi une technique pour permettre le déplacement des données sensibles. Il permet de sécuriser le déplacement par exemple mais la donnée devra être déplacée vers un système offrant une continuité de cet encryptage. Le niveau de sensibilité va lui aussi nécessiter un niveau de protection de la plateforme qui héberge ces données, de très haut niveau, impliquant un certain coût. Déplacer la donnée pour lui faire subir un traitement sur un autre environnement peut impliquer un risque élevé et donc bloquer le déplacement de la donnée. Cette problématique est présente dans le monde industriel soumis à d'importantes normes de conformité, directives et réglementation où la protection de la donnée est exigée. La sensibilité est donc un élément crucial dans le choix du transfert données-traitements. Il doit, au même titre que le volume être intégré par conception et par défaut dans les architectures des lacs de données.

3.3 L'impact du coût sur les lacs de données

La duplication des sources de données pose la problématique non seulement au niveau de la qualité mais aussi du coût de l'extraction multiple d'une même donnée et son impact sur le système où elle est émise ou déplacée. Au niveau de la gouvernance des données, multiplier les copies d'une même donnée peut entraîner une dégradation de sa valeur, engendrer des versions différentes difficiles à gérer, rendre complexe sa traçabilité et donc impacter la qualité globale du système. La mise à disposition de données ou sources de donnée a un coût sur le système émetteur ou hébergeur de cette donnée : au niveau de son extraction, du stockage même temporaire mais aussi au niveau des capacités physiques (mémoire, processeurs, etc.). La multiplication de sollicitations trop importantes peut être un frein à la mise à disposition de copie de données. Le volume de données à dupliquer et extraire, ainsi que la fréquence de ces copies peut aussi accentuer cet impact. Un autre effet de la duplication de la donnée est le coût associé à sa traçabilité. En effet pour répondre à certaines réglementations (précédemment citées), les données doivent être tracées, leurs accès et traitements subis conservés, en vue d'un audit par exemple. Cette traçabilité fait grossir les volumes des *logs* des différents serveurs, augmentant ainsi les coûts de traitement et de stockage. La duplication de données, un des principes utilisés des lacs de données, génère un coût qu'il convient d'évaluer finement dès la conception.

Nous avons donc établi que les trois paramètres volume-masse, sensibilité et coût du déplacement des données inclus dans la gravité des données peuvent remettre en cause la relation données-traitements au sein des lacs de données.

Si nous appliquons l'analogie de la gravité des données avec la vision physique, ce sont les traitements qui utilisent ces données qui seront attirés à elles et pourront donc être déportés à l'endroit où elles se situent et non pas le contraire. C'est donc le traitement qui va aller vers la donnée et non plus la donnée que l'on va déplacer vers le traitement. La gravité des données impacte donc l'architecture applicative des lacs de données mais aussi leur architecture technique. Il faut donc explorer les possibilités techniques d'amener les traitements où résident les données désormais et envisager des solutions alternatives à la structure physique unique comme réceptacle des lacs de données. La prochaine section étudie l'impact de la gravité des données sur les architectures applicatives et techniques des lacs de données au travers d'un cas réel.

4. Etude de cas : La gravité des données sur un lac de données industriel

4.1 Description de l'étude de cas industriel

L'étude de cas industriel dans le domaine financier est celui d'un lac de données dédié à la collecte de données provenant de tout un parc informatique composé de différents types de serveurs, réseaux et baies de stockage. L'objectif de ce lac est d'améliorer la connaissance de ce parc pour en améliorer e pilotage. Le lac de données est basé sur de la technologie *Apache Hadoop* et une suite d'outils d'aide à la manipulation, l'exploration et l'administration des données : *HortonWorks Data Plateforme* (HDP). Les données émises par les serveurs et autres sont poussées en temps réel dans le lac de données HDP et explorées par les utilisateurs du lac de données.

4.2 Evaluation de la gravité des données sur le lac de données métrologie

L'observation de chaque paramètre a été effectuée sur un mois afin d'intégrer les pics d'activités de l'industriel et être représentative.

4.2.1. Le volume

Nous avons recueilli le volume moyen émis par minute et par type de serveurs (mainframe, x86, Unix.) afin d'estimer le volume journalier que le lac de données aurait à intégrer. Pour chaque type de serveurs[7] (qui représentent nos sources de données dans le lac de donnée), nous pouvons calculer le volume journalier attendu dans le lac de données métrologie, soit :

Ej (serveurs) = Nb de serveurs x 24 (heures) x 60 (minutes) x **Ev** (Petabytes)

Où **Ej** est l'estimation moyenne journalière par type de serveur (Petabyte) et **Ev** est l'estimation volume / minute / serveur (Gigabyte).

[7] Serveurs de type x86 : 18000 ; Serveurs de type Unix : 30 ; Serveurs de type Mainframe : 6 ; Baies de stockage : 50 ; Réseaux : 3 types LAN, MAN, WAN.

Cela donne pour les sources de données de notre lac de données, un volume estimé de 330 Petabytes environ par jour pour tous les serveurs à intégrer. Dans le cadre de la gouvernance de la donnée, il a été décidé une conservation des historiques de données de 30 jours, ce qui implique que la conservation des données augmente le volume dans le lac de données, soit :

Ej x 30 = 9 900 Petabytes soit environ 1 Exabyte de données, dédié à l'historique.

En y ajoutant le 1 exabyte de données produites en 30 jours, nous avons donc un volume de 2 exabytes de données , au minima dans le lac de données.

Le Tableau 1 récapitule les données moyennes mesurées sur les serveurs existants.

Tableau 1. Volume par type de serveur

	Serveur X86	**Serveurs Unix**	**Serveurs Mainframe**	**Baies de stockage**	**Réseaux**
Ev	12	20	1	20	900
Ej	311	0,86	0,00864	14	3,89

Si les 2 exabytes de données représentent un volume important ce dernier ne peut être considéré comme une contrainte assez forte pour empêcher le déplacement des données. Cependant la gestion du cycle de vie de la donnée a été imposée seulement à un mois de conservation d'historique, ce qui explique ce chiffre de 2 exabytes. Ce choix ne nous semble pas réaliste et surtout ne tient pas compte des accroissements de volume généré par les analyses et explorations faites, ni des profondeurs d'historique nécessaires lors du travail en analyse prédictive, où souvent plusieurs mois ou années sont nécessaires. Une profondeur d'historique de seulement un an, entraînerait un volume d'au moins 24 Exabytes de données, ce qui pourrait alors modifier la vision de son déplacement.

A ce stade de l'étude, le volume seul n'est cependant pas jugé assez influant pour bloquer le déplacement des données mais nous émettons une alerte sur l'estimation qui en faite.

4.2.2. La sensibilité

La sensibilité des données a été considérée comme peu influente lors de la conception de l'architecture fonctionnelle. Or l'organisme financier est soumis à la Loi de Programmation Militaire (LPM) et certaines données transitant par ses réseaux doivent être protégées car jugées sensibles. Les données de métrologie provenant notamment des serveurs de type mainframe ont été classifiées hautement sensibles, car les applications critiques de l'industriel sont opérées via ces serveurs. De plus, le cas de la métrologie n'est pas représentatif, chez cet industriel, de la réelle évaluation de la sensibilité, pour les futurs autres lacs de données, notamment celui des données clients qui va être soumis à la règlementation européenne RGPD. Ce facteur n'a donc pas été évalué correctement lors de l'architecture fonctionnelle et peut remettre en

cause le déplacement de certaines données. Le Tableau 2 réévalue la sensibilité des données selon leur provenance.

Tableau 2. Evaluation de la sensibilité des données selon leur provenance

	Serveur X86	**Serveurs Unix**	**Serveurs Mainframe**	**Baies de stockage**	**Réseaux**
Sensibilité	2	6	10	8	9
Évaluation	Faible	Moyenne	Haute	Haute	Haute

Le paramètre de sensibilité doit donc être approfondi notamment pour les données provenant des serveurs mainframe, car ils contiennent les applications et données stratégiques de l'industriel. C'est pour cela que nous avons concentré l'étude de coût du déplacement des données de ce type de serveur (mainframe).

4.2.3. Le coût

Nous avons évalué le coût de déplacement de 1 TB de données par jour du serveur mainframe vers un autre serveur. Ce coût se mesure en « *million instructions per second* » (MIPS) qui est l'unité de facturation d'un serveur. Ce coût est composé des éléments suivants :

– L'utilisation de 4 cœurs de processeurs sur un mainframe de type z13 (taux de charge 85 %) : cela correspond en unité de mesure mainframe à 519 MIPS par jour ; le coût journalier est donc de 6756,4 $ (prix moyen observé pour 519 MIPS) ; sur une année, le coût est estimé à 2 466 103 $;

– à ce premier coût, il faut ajouter ceux d'administration et de maintenance du serveur ; une étude fait état d'un coût moyen de 98 482 $ par an.

La réplication de 1 TB de données revient donc à 2,55 M$ par an. Comme le volume estimé par jour de données pour la métrologie à répliquer est estimé à 8,6 TB (cf. Tableau 1), le **coût total sur une année représente plus de 22 M$.** *Ce calcul a été validé par une étude interne à IBM réalisée en laboratoire.*

Le déplacement des données des serveurs mainframe a donc un coût très important, dont le concepteur du lac de données n'a pas tenu compte lors de sa création.

4.3 Conclusion du cas d'étude de lac de données métrologie

Nous avons évalué au travers de trois paramètres (volume-masse, sensibilité et coût) l'impact potentiel de la gravité sur l'architecture du lac de données industriel. Si le volume n'a pas eu d'impact significatif, l'évaluation du coût et de la sensibilité sur certains serveurs (les mainframes) impose que la relation données-traitement soit

revue. Un mode d'accès en fédération et non en réplication doit être mis en place pour les données provenant de ce type de serveurs.

5. Conclusions et perspectives

Le principal objectif poursuivi dans cet article est celui de répondre à la question : qu'est-ce qui peut remettre en cause le choix d'une architecture fédératrice mono-technologique des lacs de données ?

Une partie de la réponse peut être faite en prenant en compte de façon systématique la gravité des données et en évaluant les éléments que sont : le volume, la sensibilité et le coût de déplacement des données vers le lac de données.

Cela ouvre la porte à des solutions alternatives d'architectures de lac de données hybrides, composées de données dupliquées mais aussi de données seulement référencées, accédées en mode *fédération* et dans lesquelles les zones de stockage des données sont elles aussi différentes en termes de d'architectures techniques.

Dans le cadre de notre étude nous avons évalué l'impact de la gravité des données sur un lac de données « *in situ* ». Une perspective à nos travaux de recherche est d'étudier l'impact de ce facteur dans la décision de positionner un lac de données « in situ » versus dans les « nuages ». Dans ce cas, la sensibilité des données personnelles en particulier va nécessiter d'aborder les aspects de confidentialité via à vis du prestataire mais aussi du fournisseur d'accès. Cela va poser des problématiques supplémentaires et générer un coût de gestion. Le transfert des données est une opération qui peut se révéler rapidement très couteuse comme on vient de la voir en section 4.2.3.

Bibliographie

Alrehamy H. et Walker C. (2015). Personal Data Lake With Data Gravity Pull. Proceedings 2015 Ieee Fifth International Conference on Big Data and Cloud Computing Bdcloud 2015, p. 160-167.

Gartner. (2015, September 15). Gartner Says Business Intelligence and Analytics Leaders Must Focus on Mindsets and Culture to Kick Start Advanced Analytics. from http://www.gartner.com/newsroom/id/3130017

Hortonworks. (2017). Hortonworks. from https://fr.hortonworks.com

Lenovo. (2016). Lenovo Big Data Reference Architecture for Hortonworks Data Platform. 4. https://cloud.kapostcontent.net/pub/9b91ad01-2f63-4c7b-ac2d-c0b5bb2af9e5/lenovo-big-data-ra-for-hortonworks-data-platform-1.pdf ? ?kui=dk4jpyPfd3pe6YP6Adkgfg

Madera C. et Laurent A. (2016). The Next Information Architecture Evolution: The Data Lake Wave. Proceedings of the 8th International Conference on Management of Digital Ecosystems (Medes 2016), p. 174-180. doi: 10.1145/3012071.3012077

MarketsAndMarkets. (2016). Data Lakes Market worth 8.81 Billion USD by 2021. from http://www.marketsandmarkets.com/PressReleases/data-lakes.asp

(McCrory D. (2010, December 07). Data Gravity – in the Clouds. from https://blog.mccrory.me/2010/12/07/data-gravity-in-the-clouds/

McCrory D. (2014, March 1). Data Gravity. from https://datagravity.org/

Russom P. (2017). Best Practices Report | Data Lakes: Purposes, Practices, Patterns, and Platforms. March 29, 2017.

Servigne S. (2010). Conception, architecture et urbanisation des systèmes d'information. Encyclopædia Universalis, p. 1-15.

IT Glossary Gartner. (2014) . https://www.gartner.com/it-glossary/data-lake

Modèles et systèmes d’information

Modèle tensoriel pour l'entreposage et l'analyse des données des réseaux sociaux

Application à l'étude de la viralité sur Twitter

Éric Leclercq, Marinette Savonnet

Laboratoire LE2I - FRE 2005 - CNRS - Arts et Métiers
Univ. Bourgogne Franche-Comté
9, Avenue Alain Savary
F-21078 Dijon - France
prénom.nom@u-bourgogne.fr

RÉSUMÉ. *Dans cet article, nous montrons comment la notion de tenseur permet de construire un modèle multi-paradigmes pour l'entreposage des données sociales. D'un point de vue architecture, cette approche permet de lier différents systèmes de stockage (polystore) et de limiter l'impact des outils ETL réalisant les transformations de modèles pour alimenter différents algorithmes d'analyse. Ainsi, le modèle proposé permet d'assurer l'indépendance logique entre les données et les programmes implantant les algorithmes d'analyse. Avec un cas concret extrait d'une étude de la viralité sur Twitter durant la période de l'entre deux tours de l'élection présidentielle française de 2017, nous mettons en évidence les apports de notre modèle.*

ABSTRACT. *In this article, we show how the notion of tensor can be used to build a multi-paradigm model for the storage of social data. From an architectural point of view, this approach allows to link different storage systems (polystore) and thus limits the impact of ETL tools performing model transformations to feed different analysis algorithms. The proposed model allows to reach the logical independence between data and programs implementing analysis algorithms. With a concrete case study on message virality on Twitter during the period between the two rounds of the French presidential election of 2017, we highlight the contributions of our model.*

MOTS-CLÉS : *stockage polyglotte, réseau multi-relationnel, tenseur, OLAP*

KEYWORDS: *polyglot storage, multi-relational network, tensor, OLAP*

1. Introduction : contexte et problématique

Les données des réseaux sociaux, en particulier celles de Twitter, sont de plus en plus exploitées dans des projets de recherche appliquée, en sciences sociales par exemple. Ces données, riches en informations sur les interactions entre individus, permettent aux chercheurs de comprendre les modèles de communication de la société numérique et les interactions entre les réseaux sociaux numériques, les médias traditionnels et la réalité. Les résultats de ces recherches sont applicables dans de nombreux domaines comme le marketing, le journalisme, l'étude de l'impact des politiques publiques, l'étude de la communication politique, etc.

Cependant, pour éclairer leurs questions de recherche les chercheurs en sciences sociales doivent effectuer des analyse exploratoires sur les données, émettre des hypothèses, développer des modèles (Armatte, 2005) et les tester. Cette démarche nécessite généralement l'utilisation de plusieurs algorithmes de fouille de données, de *machine learning* et requiert une phase d'interprétation des résultats exploitant la connaissance du domaine. Les différentes catégories d'algorithmes permettent de détecter des communautés (Drif, Boukerram, 2014), des événements (Atefeh, Khreich, 2015), des utilisateurs influents (Riquelme, González-Cantergiani, 2016), simuler ou étudier des propagations de messages. Les algorithmes ont recours à des modèles de données variés comme des graphes, des matrices d'adjacence, des séries temporelles. De plus, les algorithmes n'utilisent pas tous de la même manière les données, par exemple un algorithme de graphe peut optimiser une fonction et/ou effectuer une marche aléatoire sur le graphe (*random walk*), ou encore détecter les arêtes d'un graphe par lesquelles passent le plus grand nombre de plus courts chemins. Les algorithmes récents d'analyse des données des réseaux sociaux sont rarement implantés dans les SGBD et encore plus rarement les opérations matricielles et les factorisations associées (LU, SVD, CUR, etc.) (Leskovec *et al*., 2014). Seuls quelques systèmes NoSQL comme Neo4j proposent des outils d'analyse de graphe de données assez avancés [1]. Cependant, Neo4j ne permet pas de gérer de grandes quantités de données attributaires comme le feraient les systèmes orientés colonnes (Hölsch *et al*., 2017).

Pour le traitement des masses de données, certains algorithmes peuvent être exécutés sur des clusters de machines. On assiste depuis quelques années à la convergence de deux domaines de recherche séparés : le calcul hautes performances (*High Performance Computing - HPC*) et les bases de données. Ainsi, une des préoccupations du *data intensive HPC* est de pouvoir alimenter rapidement les algorithmes avec les données à analyser. Pour ce faire plusieurs types de stockage peuvent être combinés (systèmes de fichiers distribués HDFS, bases de données orientées colonnes, etc.) sous la forme d'un *polystore* ou système de stockage multi-paradigmes dénommé aussi stockage polyglotte (*polyglot storage*). Dans ce type de systèmes, les données peuvent être stockées dans le modèle qui convient le mieux au type de données et aux algorithmes ; une duplication partielle peut aussi être opérée. Les *polystores* se distinguent

1. https://neo4j.com/blog/efficient-graph-algorithms-neo4j/

des *data-lake* selon deux aspects, le premier concerne la finalité, les données stockées dans un *polystore* le sont dans un objectif d'analyse à court terme, celles dans un *data-lake* sont annotées en vue d'une utilisation à moyen, long terme, le second aspect concerne les performances, un *polystore* est conçu pour exploiter au mieux les modèles de stockage les plus adaptés aux données en les combinant, les *data-lake* stockent le plus souvent les données dans leur format natif.

Les travaux qui concernent les *polystores* sont axés principalement sur l'uniformisation des systèmes par les langages mais peu proposent une approche orientée modèle. C'est ce que nous proposons d'étudier dans cet article (voir figure 1).

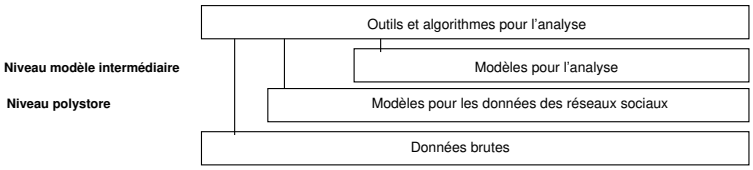


Figure 1. Relations entre les modèles et les outils d'analyse

Dans une partie état de l'art, nous discutons des modèles d'entrepôt pour les données sociales et nous présentons les principaux *polystores*. Nous décrivons ensuite notre proposition de modèle à base de tenseur et montrons comme il s'intègre dans une architecture d'entrepôt de données *polystore*. Ce modèle permet de généraliser les modèles de graphes (multi-couches, multi-relationnel), les séries temporelles et les modèles matriciels mais aussi de lier différentes données provenant des systèmes de stockage. Dans une troisième partie, nous montrons l'intérêt de notre modèle pour l'étude de l'impact des robots dans les phénomènes de viralité sur Twitter.

2. État de l'art

Dans cette section, nous présentons les différentes approches pour l'entreposage de données sociales du point de vue des modèles et de l'OLAP (*OnLine Analytical Processing*) puis du point de vue des polystores. Nous terminons cette section par une discussion sur les atouts et les faiblesses de ces solutions.

2.1. Modèles OLAP pour les données sociales

Depuis 2010, différents travaux ont cherché à étendre ou à adapter les techniques OLAP pour les données des réseaux sociaux. Les approches les plus nombreuses s'orientent vers un modèle d'entrepôt R-OLAP (*Relational-OLAP*) pour un usage particulier. Dans le modèle R-OLAP, les dimensions sont les axes avec lesquels les analyses sont effectuées et les faits sont sur quoi portent les analyses. Une façon de mettre en relation les dimensions et les faits est réalisée par le modèle en étoile ou en flocon. Par exemple, il peut y avoir une dimension temps, une dimension client, une dimension géographie pour une étude de produits.

Pour les données des réseaux sociaux, les tables de faits et de dimensions sont construites en vue d'analyses spécifiques. Par exemple, l'article de Bringay *et al*. (2011) se concentre sur la localisation, le temps et les mots importants apparaissant dans les tweets. Ces mots sont déterminés à l'aide d'une extension de TF-IDF qui déterminent les mots importants en fonction de leur place dans une hiérarchie de termes. D'autres auteurs ont construit un entrepôt dédié à l'analyse de sentiments à partir du contenu de tweets, avec le modèle R-OLAP (Moalla, Nabli, 2014) ou avec un hypercube (Moya *et al*., 2011). Bouillot *et al*. (2012) ne s'intéressent qu'à la dimension localisation des tweets en fonction du lieu d'émission du tweet, du contenu géographique, du lieu de résidence du compte.

D'autres recherches ont un caractère plus générique et proposent une modélisation en étoile des données sociales. Les travaux de Rehman *et al*. (2012) décrivent un modèle R-OLAP centré autour du compte utilisateur comme table de fait pour connaître son activité au cours du temps. Cet entrepôt a été utilisé pour analyser le comportement des utilisateurs lors d'un tremblement de terre. Une approche similaire est développée dans Mansmann *et al*. (2014). Kraiem *et al*. (2015) proposent un modèle conceptuel générique pour Twitter et le traduisent en R-OLAP : deux tables de faits sont construites une pour l'activité du tweet et l'autre pour l'activité du compte utilisateur en prenant en compte les dimensions sources, temps, lieu et utilisateurs. Kazienko *et al*. (2011) développent une modélisation hypercube des réseaux sociaux où les interactions entre groupes d'individus sont modélisées. Mansmann (2008) présente des travaux plus généraux, orientés vers une extension des technologies OLAP pour les données complexes sans réellement prendre en compte les algorithmes d'analyse.

Costa *et al*. (2012), mettent en évidence la nécessité de contextualiser les données pour aider l'analyse. Les approches décrites dans (Gallinucci *et al*., 2013) et (Rehman *et al*., 2013) élaborent un modèle intégrant un enrichissement sémantique des données collectées.

Une autre piste de recherche est apparue en 2008 avec la publication de (Chen *et al*., 2008). Le *Graph OLAP* consiste à construire un cube de graphes dans lequel il est possible de naviguer. Deux types de dimensions ont été définis : les dimensions informationnelles comme le temps, le lieu et les dimensions topologiques qui se rapportent à la modélisation des graphes dans les cellules du cube. Dans (Zhao *et al*., 2011), les auteurs étendent *Graph OLAP* et proposent le modèle *Graph Cube* pour modéliser les réseaux complexes incluant des nœuds avec des attributs multiples, et définissent un ensemble d'opérateurs spécifiques aux graphes. Le modèle est implémenté en C++ et expérimenté sur des graphes de taille moyenne à partir de jeux de données issus de DBLP [2] et IMDB [3]. Favre *et al*. (2017) étudient une autre approche qui consiste à enrichir le graphe grâce à des cubes qui valuent les nœuds et les arêtes. Loudcher *et al*. (2015) proposent un état de l'art et une étude comparative des différentes approches.

2. http://dblp.uni-trier.de/xml/
3. http://www.imdb.com/interfaces/

2.2. Entrepôt polystore

La problématique de l'accès à des sources de données hétérogènes est traitée depuis de nombreuses années dans le contexte de l'intégration de schéma et des approches multi-bases (Özsu, Valduriez, 2011). Les systèmes de stockage orientés Big Data comme HDFS et les systèmes NoSQL, matures depuis quelques années, ont fait évoluer la problématique de l'accès aux sources de données hétérogènes (Franklin *et al*., 2005). Dans ce cadre, les approches *polystore* ou *multistore* ont pour but de permettre un accès intégré à de multiples systèmes de stockage de données ne supportant pas nécessairement SQL et offrant des jeux d'opérateurs différents, au travers d'un gestionnaire de requête (Bondiombouy *et al*., 2015). Une des hypothèses fortes retenue par ces systèmes est que les accès aux sources sont principalement en lecture.

Bondiombouy et Valduriez (2016) proposent un état de l'art des principaux systèmes en les classant selon trois catégories : systèmes faiblement couplés, fortement couplés et hybrides. Les systèmes faiblement couplés et fortement couplés s'opposent principalement sur deux critères : les performances sont privilégiées dans les systèmes fortement couplés au détriment de l'autonomie et de l'accès local aux sources de données. Nous distinguerons les systèmes selon leur orientation initiale : extension d'outils d'analyse ou approches systèmes de gestion de bases de données distribuées (Özsu, Valduriez, 2011).

Le premier groupe concerne des systèmes issus de projets collaboratifs et pragmatiques. Ces systèmes utilisent un moteur de requêtes SQL comme couche de médiation. La couche SQL de la plateforme Apache Spark [4] et le projet Apache Drill [5] sont deux solutions opérationnelles de ce groupe. Ces systèmes exploitent le principe de localité des données (*data locality*) et implantent un optimiseur de requêtes.

Le second groupe concerne des systèmes issus de travaux de recherche d'équipes ayant déjà contribué à résoudre la problématique d'accès à des sources de données hétérogènes dans le contexte des systèmes de gestion de bases de données traditionnels. Le système BigDAWG (Duggan *et al*., 2015) permet d'écrire des requêtes multi-bases portant sur des îlots d'information correspondant chacun à un type de modèle de données. Le langage supporté permet des accès transparents aux différents éléments d'un même îlot. L'approche Cloud Multidatastore Query Language (CloudMdsQL) (Kolev *et al*., 2016) définit un langage fonctionnel de type SQL qui permet d'écrire des requêtes composées de sous-requêtes permettant d'interroger plusieurs systèmes de stockage hétérogènes. Chaque sous-requête cible un système particulier et contient des appels à l'interface native du système en question. Ainsi CloudMdsQL peut exploiter les spécificités et les performances des système locaux au moyen des requêtes natives comme par exemple pour une requête *breadth-first search* sur une base de données graphe. SQL++, inclus dans la plateforme FORWARD [6], est une approche de média-

4. http://spark.apache.org/sql/
5. https://drill.apache.org/
6. http://forward.ucsd.edu/

tion de schéma qui a pour objectif de définir un intergiciel permettant de créer des vues virtuelles ou matérialisées sur des systèmes supportant ou non le langage SQL (Ong *et al*., 2014 ; 2015). Le modèle de données sur lequel SQL++ repose est un surensemble du modèle relationnel et de JSON. Comparée aux systèmes BigDAWG et CloudMdsQL, l'approche SQL++ se concentre sur les aspects langage et extensibilité mais aborde peu l'architecture du système.

En conclusion, les systèmes présentés, comparés aux approches traditionnelles, combinent les avantages des systèmes faiblement couplés au niveau des accès à de multiples systèmes de stockage avec les avantages des systèmes fortement couplés au niveau de l'efficacité des accès utilisant les interfaces natives. Dans cette orientation, CoudMdsQL semble l'approche la plus développée alors que SQL++ met l'accent sur une compatibilité SQL, BigDAWG quand à lui offre une approche plus restrictive car chaque îlot d'information correspond à un seul modèle de données.

2.3. Discussion

Les limites générales des approches OLAP pour des données des réseaux sociaux sont d'une part les performances et d'autre part l'évolutivité des schémas. Du point de vue des performances, les modélisations induisent des requêtes nombreuses et coûteuses lorsqu'on applique des algorithmes de graphe nécessitant un parcours de liens (dont les marches aléatoires), la recherche de plus court chemin ou la construction de matrice d'adjacence (par nature creuse). De plus, les transformations de modèle sont coûteuses comme par exemple le passage d'un graphe à des séries temporelles pour différents échantillonnages du temps, ou l'agrégation de données (*roll-up*) avec une exploitation des relations transitives comme les citations d'utilisateurs dans des tweets. Du point de vue de l'évolutivité des schémas, le problème ne se situe pas au niveau des opérations permettant de transformer les données mais plutôt de la connaissance nécessaire afin de déterminer les dimensions pertinentes pour les analyses. Comme le faisait remarquer Byron Ruth dans une présentation intitulée "*ETL: The Dirty Little Secret of Data Science*" lors de la conférence OSCON [7], les ETL sont des processus coûteux à mettre en place et les scripts de transformation de données incluent bien souvent une connaissance implicite qui ne favorise pas leur réutilisation.

Les polystores sont peu considérés dans les approches d'entreposage qui restent encore majoritairement dans une vision traditionnelle du SGBD. Les approches décrites partagent le même principe, c'est-à-dire une unification par le langage de requête tout en préservant pour CloudMdsQL les requêtes natives. Au niveau modèle, il s'agit dans tous les cas soit du modèle relationnel avec ou sans imbrication soit d'un modèle de type JSON.

7. https://conferences.oreilly.com/oscon/oscon2014/public/schedule/detail/34578

3. L'approche TDM (*Tensor Data Model*)

Notre approche fait comme hypothèse de préserver l'autonomie locale des systèmes sans considérer les requêtes de mises à jour des données hormis celles qui consistent à matérialiser les résultats des analyses. Il s'agit donc d'une approche multi-paradigmes de l'entreposage par *polystore* dans le sens où il est possible d'utiliser soit le langage propre à chaque système soit le modèle intermédiaire tensoriel servant à faciliter les transformations de modèles vers les algorithmes d'analyse (figure 2). Le modèle tensoriel assure ainsi le découplage entre les programmes et les données (*logical data independency*) comme l'illustre la figure 3. Du point de vue des outils d'analyse nous réduisons pour l'instant notre système à deux types de langages : R et Spark [8]. Nous préciserons la notion de tenseur dans les sections suivantes. Dans un premier temps nous nous limiterons à l'analogie d'un tenseur avec un tableau multi-dimensionnel ou une hypermatrice, c'est-à-dire une famille d'éléments indexée par N ensembles, N étant l'ordre du tenseur autrement dit le nombre de dimensions.

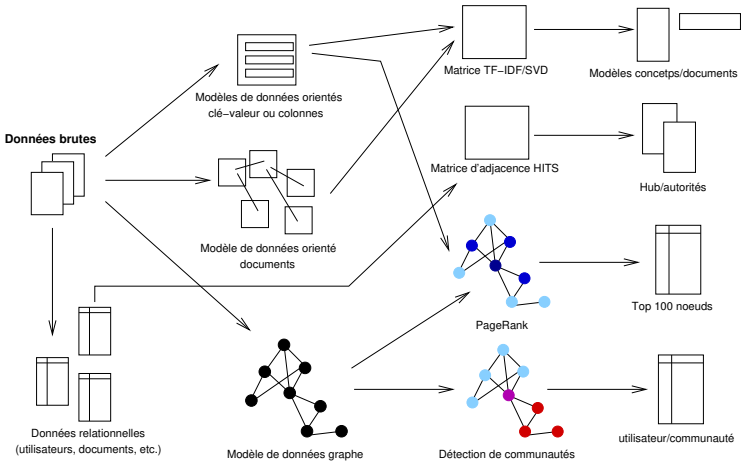


Figure 2. Modèles, transformations de modèles, algorithmes

3.1. Architecture

La figure 3 décrit comment les données qui peuplent les tenseurs du modèle intermédiaire sont obtenues avec un adaptateur (*wrapper*) qui exprime les requêtes soit dans le langage de manipulation de données du système de stockage (en R par exemple) ou en utilisant une sur-couche SQL (SparkSQL pour Spark). Dans les deux

8. Tensor flow https://www.tensorflow.org/ est comparable aux bibliothèques supportant les tenseurs dans R ou Spark, tous comme ces dernières, la bibliothèque n'a pas été construite avec une orientation modèle de représentation de données. Il s'agit plutôt d'une structure d'échange entre les processus d'un *workflow* complexe qui offre les opérateurs classiques de manipulation et décomposition des tenseurs.

cas, les données extraites transitent par une structure *data-frame* avant d'être modélisées dans un tenseur. Les requêtes de construction de tenseur soumises aux *wrappers* ont la particularité d'avoir toutes la même forme : elles renvoient $N + 1$ attributs où les N premiers attributs indiquent les dimensions et le dernier sert de valeur pour les éléments du tenseur (obtenu avec l'ajout d'une clause `GROUP BY` sur les attributs qui représentent les dimensions). Cette particularité nous permet d'avoir des *wrappers* ayant tous la même forme et ainsi simplifier les transformations de modèles. Pour réaliser les *wrappers* nous utilisons les packages R DBI, RNeo4j, RMongo, RCassandra et RHBase, pour Spark nous utilisons la couche SQL et les données sont stockées sous la forme de *data frames* et RDD (*Resilient Distributed DataSets*). Les dimensions des tenseurs sont représentées par des tableaux associatifs notés ta_i pour $i = 1, .., N$ qui contiennent les valeurs attributs identifiants comme par exemple des utilisateurs, des hashtags, des mentions d'utilisateurs et permettent d'établir les liens entre les différents systèmes de stockage.

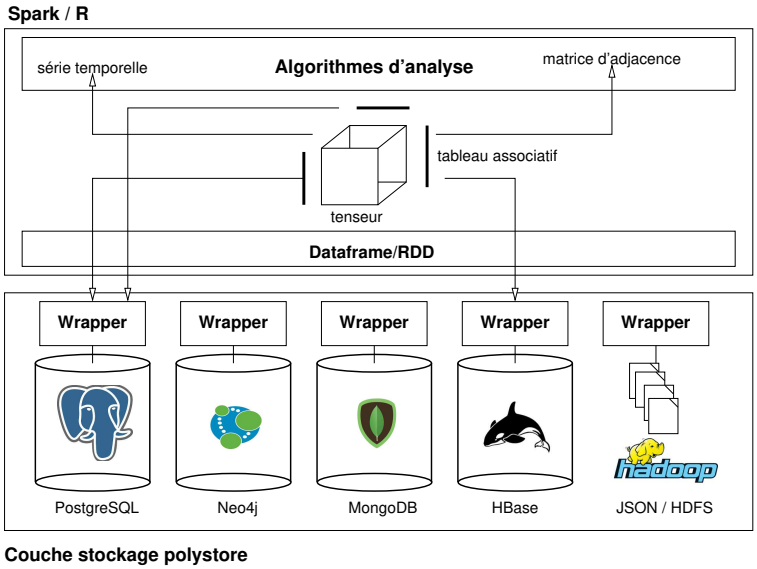


Figure 3. Architecture entrepôt polystore et modèle tensoriel

Les algorithmes d'analyse peuvent travailler directement sur le tenseur en utilisant par exemple une décomposition tensorielle pour extraire des relations cachées ou bien, utiliser le modèle tensoriel comme modèle intermédiaire pour produire par exemple une matrice d'adjacence ou une série temporelle qui seront utilisées par d'autres algorithmes.

3.2. De l'objet mathématique tenseur au modèle de données

Les tenseurs sont des objets mathématiques abstraits très généraux qui peuvent être considérés selon différents points de vue. Les tenseurs peuvent être vus comme

des applications multi-linéaires ou comme le résultat d'un produit tensoriel. Dans un contexte de bases de données, il s'agit, plutôt de tableaux multi-dimensionnels (Baumann, Holsten, 2011 ; Stonebraker *et al.*, 2013) et donc, par analogie avec la définition des matrices, d'une famille d'éléments d'un corps K indexée par N ensembles (tenseur d'ordre N). Plus formellement, un tenseur d'ordre N est un élément du produit tensoriel de N espaces vectoriels chacun ayant son propre système de coordonnées. Un tenseur d'ordre 1 est un vecteur, un tenseur d'ordre 2 est une matrice, et les tenseurs d'ordre supérieur à 3 sont regroupés sous la dénomination de tenseurs d'ordre supérieur. Dans une définition plus pragmatique, un tenseur est un élément de l'ensemble des fonctions du produit de N ensembles $I_j, j = 1, \ldots, N$ vers $\mathbb{R}$: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, N est le nombre de dimensions, ou ordre, ou encore mode. Dans la suite nous utilisons les lettres capitales en gras dans la police Euler pour désigner un tenseur $\boldsymbol{\mathcal{X}}$, pour les matrices les lettres majuscules droites en gras, les lettres minuscules pour désigner un élément du tenseur par exemple x_{ijk} est le ijk-ième élément du tenseur $\boldsymbol{\mathcal{X}}$ d'ordre 3.

L'objectif est de munir l'objet mathématique tenseur d'opérations de manipulation, d'analyse et de lui adjoindre la notion de schéma afin d'en faire un modèle de données (figure 4).

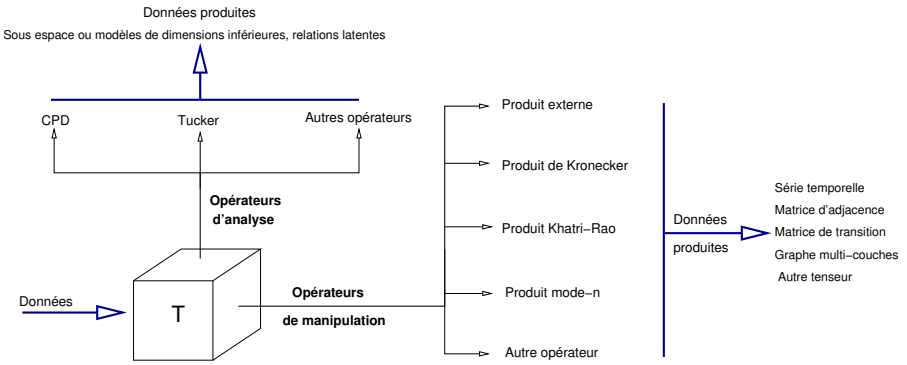


Figure 4. Modèle et opérateurs

Parmi les opérations courantes sur les tenseurs et par analogie avec les opérations sur les matrices et les vecteurs, nous retrouvons des opérateurs de multiplication ainsi que les dépliements et le matriçage (*matricization* ou *matricification*) (Kolda, Bader, 2009). Les produits tensoriels les plus utilisés sont le produit de Kronecker noté $\otimes$, de Khatri-Rao noté $\odot$, d'Hadamard noté $\circledast$, externe noté $\circ$ et mode-n noté $\times_n$.

3.3. Opérateurs de projection

Une fibre est un fragment mono-dimensionnel d'un tenseur analogue à la notion de vecteur, et aux lignes ou colonnes d'une matrice. Ainsi, pour un tenseur d'ordre 3 les fibres peuvent être des colonnes, des lignes ou des tubes notées respectivement : $\boldsymbol{\mathcal{X}}_{:jk}$, $\boldsymbol{\mathcal{X}}_{i:k}$ et $\boldsymbol{\mathcal{X}}_{ij:}$. Une tranche de tenseur est un fragment d'ordre 2. Par exemple

pour un tenseur d'ordre 3 on obtient des tranches horizontales, latérales et frontales notées respectivement : $\boldsymbol{\mathcal{X}}_{i::}$, $\boldsymbol{\mathcal{X}}_{:j:}$ et $\boldsymbol{\mathcal{X}}_{::k}$. Que ce soit pour les fibres ou les tranches il est courant de les nommer en fonction de leur dimension d'origine (mode-1, mode-2 ou mode-3). Tranches ou fibres horizontales sont obtenues par le produit mode-3 $\times_3$, latérales par le produit mode-2 $\times_2$, frontales par le produit mode-1 $\times_1$.

Les opérations de projection peuvent être généralisées par le produit d'Hadamard d'un tenseur d'ordre N avec un tenseur booléen de même ordre qui contient des valeurs 1 pour les éléments à sélectionner : $\boldsymbol{\mathcal{X}} \circledast \boldsymbol{\mathcal{B}}$. Par exemple pour un tenseur d'ordre 3, $\boldsymbol{\mathcal{T}}_1$, représentant les utilisateurs, les hashtags utilisés (leur nombre d'occurrences dans les tweets de l'utilisateur) et le temps, sélectionner tous les hashtags utilisés par un utilisateur i se traduit par un tenseur d'ordre 2 : $\boldsymbol{\mathcal{T}}_2 = \boldsymbol{\mathcal{T}}_1 \circledast \boldsymbol{\mathcal{B}}_1$ avec $\boldsymbol{\mathcal{B}}_{1_{i::}} = 1$. Pour obtenir une série temporelle traduisant l'utilisation d'un hashtag on effectue la somme des colonnes du tenseur d'ordre 2 obtenu.

3.4. Opérateurs de sélection

Les opérateurs de sélection peuvent agir à deux niveaux : 1) sur les valeurs contenues dans le tenseur (équivalent à une sélection portant sur un seul attribut en relationnel) ou bien 2) sur les valeurs des dimensions qui sont dans notre cas les tableaux associatifs notés ta_i pour $i = 1, .., N$. Ces derniers peuvent être considérés comme des relations à deux attributs du modèle relationnel associant un identifiant externe à une valeur d'index. L'opérateur de sélection σ s'écrit avec deux conditions la première portant sur les dimensions, l'autre sur les valeurs : $\sigma[cdt\ dim][cdt\ val]\boldsymbol{\mathcal{X}}$. La composante de l'opérateur de sélection portant sur les dimensions est réalisée par le produit d'Hadamard par tenseur booléen dont les éléments de valeur 1 correspondent aux éléments des ta_i sélectionnés.

Par exemple, pour sélectionner tous les hashtags contenus dans les tweets d'un utilisateur u_1 entre les temps $t = 10$ et $t = 40$ et dont le nombre d'occurrences est supérieur à 25, à partir du tenseur $\boldsymbol{\mathcal{T}}_1$ dont les dimensions sont U, H et T on effectue une sélection : $\sigma[U =' u'_1 \wedge T \geq 10 \wedge T \leq 40][> 25]\boldsymbol{\mathcal{T}}_1$

3.5. Opérateurs analytiques de décomposition

Les décompositions tensorielles telles que CANDECOMP/PARAFAC (CP), Tucker, HOSVD sont utilisées pour effectuer des réductions de dimensions et extraire des relations latentes (Kolda, Bader, 2009). Comme les représentations des données par tenseur sont multiples et leur sémantique non explicite, les résultats des décompositions tensorielles sont complexes à interpréter. Par analogie avec les décompositions matricielles il est possible de déterminer la décomposition associée à un objectif. Par exemple, pour exprimer chaque espace engendré par une des dimensions du tenseur indépendamment des autres mais en fonction de l'espace global on peut utiliser une décomposition CP (figure 5). Pour déterminer des modèles d'utilisateurs en fonction de hashtag ou des motifs récurrents de comportement on préférera utiliser une décom-

position HOSVD. Les modèles produits peuvent alors être utilisés dans des systèmes de recommandation.

4. Retours d'expériences

Le jeu de données sur lequel nous travaillons est un corpus de 50M de tweets collectés dans le cadre du projet pluridisciplinaire TEP2017 dont l'objectif est l'analyse de la communication politique sur Twitter lors de l'élection présidentielle de 2017. Les données brutes au format JSON représentent 720Go, les attributs les plus courants ont été sélectionnés pour constituer une base de données relationnelle non normalisée (50Go) à partir de laquelle une seconde base de données relationnelle en 3FN a été créée autour des entités utilisateur, hashtag, tweet (55Go incluant les index). Dans la suite, nous présentons la modélisation du réseau social Twitter à l'aide du modèle TDM et nous décrirons nos expérimentations.

4.1. Modélisation avec TDM

Le réseau social Twitter est un réseau complexe dans lequel les nœuds sont hétérogènes (utilisateurs, tweets, hashtags, etc.) et les liens également (retweet, émet, follows, cite, etc.). On considère un ensemble d'entités E incluant les utilisateurs et les ressources (tweets, hashtags, URL) sous la forme d'une partition, et un ensemble de relations R entre les entités. Ces deux ensembles sont définis comme suit : $E = \bigcup_{i=1}^{n} E_i$ et $R = \{R_i, i = 1, \ldots, m\}$ avec $R_i : E_k \times E_l \rightarrow \mathbb{N},\ k, l \in \{1, \ldots, n\}$. Les relations R_i sont des applications, elles peuvent être représentées par des matrices. Par exemple, un utilisateur est décrit par m vecteurs, un pour chacune des relations. Les relations n'ont pas la même signature comme par exemple :

– utilisateur/hashtag pour l'utilisation d'un hashtag dans un de ses tweets ;

– utilisateur/utilisateur pour une relation suivre, ou une mention dans un tweet ou encore le relais d'un tweet d'un utilisateur ;

– utilisateur/tweet pour l'émission d'un tweet, ou le retweet, etc.

L'ensemble des utilisateurs pour toutes les relations considérées peuvent être représentés par un tenseur d'ordre 3, si toutes les relations à représenter sont homogènes c'est-à-dire si les applications associées ont la même signature. Par exemple, deux des modes du tenseur seront E_1 et E_k avec $k \in 1, n$ et le troisième représentera les différentes relations R_i. Il peut s'agir par exemple d'un tenseur représentant des relations entre utilisateurs comme les abonnements, les retweets, la citation, etc. De plus une dimension supplémentaire peut être ajoutée pour représenter le temps et permettre une analyse de la dynamique du réseau. Kivelä *et al.* (2014), et De Domenico *et al.* (2013) montrent comment les tenseurs peuvent représenter des relation n-aires, des hypergraphes, des réseaux multi-couches et multi-relationnels sans pour autant les considérer comme un véritable modèle de données.

Nous présentons dans la suite nos expérimentations concernant l'étude de l'impact des robots dans la propagation de tweets supposés viraux. Nous n'insisterons pas sur l'interprétation des résultats faite par les chercheurs en sciences sociales.

4.2. Robots et viralité

La viralité peut être définie par les trois paramètres résumés dans Beauvisage *et al*. (2011) : la concentration temporelle de l'attention sur un contenu, la circulation de ces contenus et les mécanismes de la contagion d'un individu à l'autre. Contrairement aux tweets qui font du *buzz*, le démarrage de leur diffusion est plus lent et celle-ci dure plus longtemps dans le temps. Des métriques d'ordre lexical (présence d'URL et de hashtag, construction du hashtag à partir de plusieurs mots, etc.) et d'ordre contextuel (activité du compte, sa communauté, etc.) sont prises en compte pour déterminer la viralité d'un tweet (Hoang *et al*., 2011 ; Ma *et al*., 2013 ; Weng *et al*., 2014). Leur nombre et leur variété rendent difficile l'interprétation des résultats obtenus.

Nous considérons que dans un premier temps, détecter la viralité potentielle d'un tweet peut être effectué en mesurant sa popularité c'est-à-dire par le nombre de retweets engendrés. Nous sommes partis du corpus global puis nous avons réduit la période d'étude à l'entre-deux tours (du 24 avril 2017 au 7 mai 2017) et calculé le nombre de retweets pour chaque tweet. Un échantillon des tweets les plus populaires c'est-à-dire dans notre cas les plus propagés par retweets a été obtenu en sélectionnant les tweets retweetés au moins 1 000 fois sur la période. Cet échantillon comporte 1 123 tweets dont certains ont été retweetés plus de 20 000 fois [9]. Les requêtes qui ont produit la liste des tweets à étudier sont exprimées dans le langage natif du système de stockage, ici SQL et lancées depuis R sur la base de données PostgreSQL en 3FN. Elles produisent les données en moins d'une minute. Mettre en avant ces tweets nous a permis de réduire le corpus de 50M de tweets à un millier, donnant la possibilité à nos collègues en sciences de la communication de valider nos expérimentations par une étude qualitative. Leur interprétation a révélé que les tweets les plus populaires et ceux qui avaient une période de présence longue contenaient des URLs référençant des vidéos ou des photos.

Afin de comprendre les mécanismes de diffusion des tweets supposés viraux, nous cherchons à étudier la part d'activité liée à des robots ou plutôt des comptes dont le comportement ne ressemble pas à celui d'un humain. D'après Bessi et Ferrara (2016), les robots ont émis 19% des tweets relatifs à l'élection présidentielle américaine de 2016. Différentes méthodes ont été proposées pour détecter les robots (Varol *et al*., 2017), elles agrègent le plus souvent un grand nombre de caractéristiques pour produire un modèle prédictif s'appuyant sur un algorithme d'apprentissage comme les *random-forest*. Une expérience en utilisant l'API OSoMe [10] pour obtenir une proba-

9. La liste des id de tweets est disponible sur GitHub (https://github.com/EricLeclercq/TEE-2017-Virality) pour une reproductibilité des résultats.

10. https://botometer.iuni.iu.edu/

bilité de comportement automatisé (de type robot) nous a conduit à constater que les valeurs des probabilités n'étaient pas assez significatives pour détecter les robots durant la période. Une des hypothèses est qu'il s'agit de comptes hybrides d'utilisateurs assistés par des algorithmes.

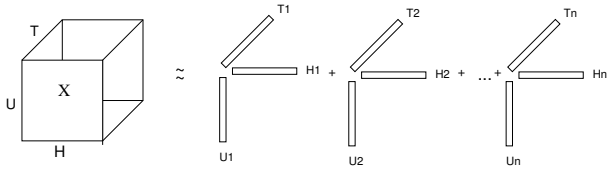


Figure 5. Illustration de la décomposition CP

À partir du comportement des utilisateurs et du contenu des tweets en utilisant les hashtags, nous avons construit un tenseur d'ordre 3 contenant les utilisateurs U ayant retweeté un tweet supposé viral, les hashtags H contenus dans ces tweets et le temps T (14 jours d'observation). Nous avons obtenu un tenseur de dimension $1077 \times 568 \times 336$; les valeurs du tenseur représentent par conséquent le nombre d'occurrence de chaque hashtag par utilisateur et par heure en considérant uniquement les retweets des 1 123 tweets supposés viraux. L'espace de recherche étant très important, nous le réduisons en effectuant une décomposition CP (figure 5) pour étudier l'espace des utilisateurs en fonction de leur comportement. La décomposition CP produit n groupes de trois tenseurs d'ordre 1 (ici les vecteurs U H T). Nous appliquons ensuite l'algorithme de clustering k-means pour identifier des groupes d'utilisateurs. La valeur de n retenue est celle à partir de laquelle il n'y a plus de modification des clusters. Expérimentalement, nous obtenons $n = 8$ et par conséquent un utilisateur est décrit par un point dans un espace à 8 dimensions. L'algorithme k-means appliqué sur ces données permet de déterminer 4 groupes d'utilisateurs : un groupe d'un compte déjà détecté comme un robot, un groupe de deux comptes, un groupe d'une trentaine de comptes comportant plus de la moitié d'utilisateurs ayant une probabilité d'être un robot supérieur à 0.6 (avec OSoMe) et un dernier groupe contenant les autres utilisateurs. Le groupe de deux comptes, se révèle, après étude manuelle, être lié (même comportement et hashtags) et assisté par un algorithme qui retweete des messages contre le candidat Macron. Ces comptes avaient échappé aux autres techniques d'analyse. Les tableaux associatifs et le tenseur sont produits à partir des données en quelques secondes, la décomposition tensorielle en R est effectuée en moins de 5 minutes. Avec Spark nous avons seulement testé la construction du tenseur et n'avons pas noté de différence de performance. Une étude approfondie des performances et du passage à l'échelle sera nécessaire.

5. Conclusion

Dans cet article nous avons proposé une nouvelle architecture pour l'entreposage de données des réseaux sociaux reposant sur un système de stockage *polystore* et un modèle intermédiaire tensoriel. Le modèle de tenseur permet de généraliser les

représentations matricielles (matrice d'adjacence, séries temporelles etc.) ainsi que les graphes dont les multigraphes et les hypergraphes. De plus, le modèle permet de prendre en compte des modélisations de réseaux complexes (réseaux multi-couches, multi-relationnels, etc.). Nous avons aussi présenté quelques opérateurs de manipulation et d'analyse de données sur le modèle de tenseur. Le travail a été expérimenté avec des données issues des réseaux sociaux. L'expérimentation sert de preuve de concept pour valider la faisabilité d'une implémentation d'un entrepôt *polystore* et de l'utilité de l'approche TDM pour des analyses de données de Twitter. Nous nous sommes intéressés à la détection de robots à partir d'un ensemble de tweets supposés viraux. Nos résultats ont été validés par des collègues en sciences de la communication à travers une étude qualitative des résultats. Cette expérimentation a démontré les capacités de modélisation des tenseurs et la pertinence de l'architecture du point de vue de la rapidité de mise en œuvre des transformations de données pour les analyses.

Les perspectives concernent la formalisation des opérateurs ainsi que l'étude des propriétés de la structure algébrique qu'ils engendrent (semi-anneau par exemple). En parallèle, nous souhaitons développer un véritable prototype d'architecture afin de pouvoir étudier l'optimisation des requêtes incluant des opérateurs tensoriels. Néanmoins, les structures de semi-anneaux confèrent aux opérateurs des bonnes propriétés pour une implémentation distribuée ce qui laisse entrevoir un bon potentiel de passage à l'échelle.

Bibliographie

Armatte M. (2005). La notion de modèle dans les sciences sociales: anciennes et nouvelles significations. *Mathématiques et sciences humaines. Mathematics and social sciences*, n° 172.

Atefeh F., Khreich W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence*, vol. 31, n° 1, p. 132–164.

Baumann P., Holsten S. (2011). A comparative analysis of array models for databases. In *Database theory and application, bio-science and bio-technology*, p. 80–89. Springer.

Beauvisage T., Beuscart J.-S., Couronné T., Mellet K. (2011). Le succès sur Internet repose-t-il sur la contagion ? Une analyse des recherches sur la viralité. *Tracés. Revue de sciences humaines*, n° 21, p. 151–166.

Bessi A., Ferrara E. (2016). Social bots distort the 2016 U.S. Presidential election online discussion. *First Monday*, vol. 21, n° 11.

Bondiombouy C., Kolev B., Levchenko O., Valduriez P. (2015). Integrating big data and relational data with a functional sql-like query language. In *Database and expert systems applications*, p. 170–185.

Bondiombouy C., Valduriez P. (2016). *Query processing in multistore systems: an overview*. Rapport technique. INRIA Sophia Antipolis-Méditerranée.

Bouillot F., Poncelet P., Roche M. (2012, août). How and why exploit tweet's location information? In *AGILE'2012: 15th International Conference on Geographic Information Science*, p. N/A. Avignon, France.

Bringay S., Béchet N., Bouillot F., Poncelet P., Roche M., Teisseire M. (2011). Towards an on-line analysis of tweets processing. In *Database and expert systems applications*, p. 154–161.

Chen C., Yan X., Zhu F., Han J., Philip S. Y. (2008). Graph OLAP: Towards online analytical processing on graphs. In *ICDM*, p. 103–112.

Costa P., Souza F. F., Times V. C., Benevenuto F. (2012). Towards integrating online social networks and business intelligence. In *Iadis international conference on web based communities and social media*, vol. 2012.

De Domenico M., Solé-Ribalta A., Cozzo E., Kivelä M., Moreno Y., Porter M. A. *et al.* (2013). Mathematical formulation of multilayer networks. *Physical Review X*, vol. 3, n° 4, p. 041022.

Drif A., Boukerram A. (2014). Taxonomy and survey of community discovery methods in complex networks. *International Journal of Computer Science and Engineering Survey*, vol. 5, n° 4, p. 1.

Duggan J., Elmore A. J., Stonebraker M., Balazinska M., Howe B., Kepner J. *et al.* (2015). The bigdawg polystore system. *ACM SIGMOD Record*, vol. 44, n° 2, p. 11–16.

Favre C., Jakawat W., Loudcher S. (2017). Graphes enrichis par des Cubes (GreC) : une approche innovante pour l'OLAP sur des réseaux d'information. In *Actes du xxxvème congrès inforsid, toulouse, france, may 30 - june 2, 2017*, p. 293–308.

Franklin M., Halevy A., Maier D. (2005). From databases to dataspaces: a new abstraction for information management. *ACM Sigmod Record*, vol. 34, n° 4, p. 27–33.

Gallinucci E., Golfarelli M., Rizzi S. (2013). Meta-stars: multidimensional modeling for social business intelligence. In *Proceedings of the sixteenth international workshop on data warehousing and olap*, p. 11–18.

Hoang T.-A., Lim E.-P., Achananuparp P., Jiang J., Zhu F. (2011). On modeling virality of twitter content. In *International conference on asian digital libraries*, p. 212–221.

Hölsch J., Schmidt T., Grossniklaus M. (2017). On the performance of analytical and pattern matching graph queries in neo4j and a relational database. In *EDBT/ICDT 2017 Joint Conference: 6th International Workshop on Querying Graph Structured Data (GraphQ).*

Kazienko P., Kukla E., Musial K., Kajdanowicz T., Bródka P., Gaworecki J. (2011). A generic model for a multidimensional temporal social network. In *e-technologies and networks for development*, p. 1–14. Springer.

Kivelä M., Arenas A., Barthelemy M., Gleeson J. P., Moreno Y., Porter M. A. (2014). Multilayer networks. *Journal of Complex Networks*, vol. 2, n° 3, p. 203–271.

Kolda T. G., Bader B. W. (2009). Tensor decompositions and applications. *SIAM review*, vol. 51, n° 3, p. 455–500.

Kolev B., Bondiombouy C., Valduriez P., Jiménez-Peris R., Pau R., Pereira J. (2016). The cloudmdsql multistore system. In *Sigmod.*

Kraiem M. B., Feki J., Khrouf K., Ravat F., Teste O. (2015). Modeling and olaping social media: the case of twitter. *Social Network Analysis and Mining*, vol. 5, n° 1, p. 47.

Leskovec J., Rajaraman A., Ullman J. D. (2014). *Mining of massive datasets*. Cambridge university press.

Loudcher S., Jakawat W., Morales E. P. S., Favre C. (2015, May). Combining OLAP and information networks for bibliographic data analysis: a survey. *Scientometrics*, vol. 103, n° 2, p. 471–487.

Ma Z., Sun A., Cong G. (2013). On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, vol. 64, n° 7, p. 1399–1410.

Mansmann S. (2008). Extending the OLAP technology to handle non-conventional and complex data (Thèse de doctorat, University of Konstanz). *KOPS*.

Mansmann S., Rehman N. U., Weiler A., Scholl M. H. (2014). Discovering OLAP dimensions in semi-structured data. *Information Systems*, vol. 44, p. 120–133.

Moalla I., Nabli A. (2014). Towards Data Mart Building from Social Network for Opinion Analysis. In *Intelligent data engineering and automated learning - IDEAL 2014 - 15th international conference, salamanca, spain, september 10-12, 2014. proceedings*, p. 295–302.

Moya L. G., Kudama S., Cabo M. J. A., Llavori R. B. (2011). Integrating web feed opinions into a corporate data warehouse. In *Proceedings of the 2nd international workshop on business intelligence and the web*, p. 20–27.

Ong K. W., Papakonstantinou Y., Vernoux R. (2014). *The sql++ unifying semi-structured query language, and an expressiveness benchmark of sql-on-hadoop, nosql and newsql databases*. Rapport technique. UCSD.

Ong K. W., Papakonstantinou Y., Vernoux R. (2015). *The sql++ query language: Configurable, unifying and semi-structured*. Rapport technique. UCSD.

Özsu M. T., Valduriez P. (2011). *Principles of distributed database systems*. Springer Science & Business Media.

Rehman N. U., Mansmann S., Weiler A., Scholl M. H. (2012). Building a data warehouse for twitter stream exploration. In *Proceedings of the 2012 international conference on advances in social networks analysis and mining (asonam 2012)*, p. 1341–1348.

Rehman N. U., Weiler A., Scholl M. H. (2013). Olaping social media: the case of twitter. In *Proceedings of the 2013 ieee/acm international conference on advances in social networks analysis and mining*, p. 1139–1146.

Riquelme F., González-Cantergiani P. (2016). Measuring user influence on twitter: A survey. *Information Processing & Management*, vol. 52, n° 5, p. 949–975.

Stonebraker M., Brown P., Zhang D., Becla J. (2013). Scidb: A database management system for applications with complex analytics. *Computing in Science & Engineering*, vol. 15, n° 3, p. 54–62.

Varol O., Ferrara E., Davis C. A., Menczer F., Flammini A. (2017). Online Human-Bot Interactions: Detection, Estimation, and Characterization. *CoRR*, vol. abs/1703.03107. Consulté sur http://arxiv.org/abs/1703.03107

Weng L., Menczer F., Ahn Y. (2014). Predicting Successful Memes using Network and Community Structure. *CoRR*, vol. abs/1403.6199.

Zhao P., Li X., Xin D., Han J. (2011). Graph cube: on warehousing and olap multidimensional networks. In *Proceedings of the 2011 acm sigmod international conference on management of data*, p. 853–864.

Une sémantique pour les patrons de justification

Clément Duffau, Thomas Polacsek, Mireille Blay-Fornarino

AXONIC, Sophia Antipolis, France
I3S, Université Côte d'Azur, CNRS, Sophia Antipolis, France
ONERA, 2 avenue Edouard Belin BP74025, 31055 TOULOUSE Cedex 4, France

RÉSUMÉ. *La création d'un produit, que cela soit un objet matériel ou un service, s'accompagne de la production de justifications qui peuvent être, suivant les cas, des éléments de conformité dans le cadre de la qualité, des documents de traçabilité, des rapports d'expérimentations, des rapports d'experts, etc. Dans des contextes critiques, comme le médical, le ferroviaire ou l'aéronautique, il est obligatoire de convaincre une autorité certificatrice, que le développement d'un produit a été correctement réalisé. Cette obligation entraîne une inflation des documents justificatifs, inflation qui rend la lecture et la compréhension de cet ensemble de justifications difficile. Pour structurer ces justifications, il peut être utile d'utiliser des diagrammes de justification. Cependant, ces diagrammes, bien qu'utiles, ne sont qu'une notation graphique informelle. Dans cet article, nous définissons une sémantique formelle du diagramme de justification et nous donnons les premières pistes de ce que pourrait être un logiciel d'aide à la conception de tels diagrammes.*

ABSTRACT. *The creation of a product, whether it is an object or a service, is accompanied by the production of justifications which may be, depending on the case, elements of conformity in the context of quality, traceability documents, experimental reports, expert reports, etc. In critical contexts, such as medical, railway or aeronautics, it is mandatory to convince a certifying authority that the development of a product has been carried out correctly. This obligation leads to an inflation of justification documents, which makes it difficult to read and to understand this set of justifications. To structure these justifications, it may be useful to use Justification Diagrams. However, these diagrams, while useful, are only an informal graphical notation. In this article, we define a formal semantics of the Justification Diagram and we give the first hints of what could be a software to support the design of such diagrams.*

MOTS-CLÉS : *Justification, Certification, Ingénierie des exigences, Exigences de qualité, Ingénierie dirigée par les modèles*

KEYWORDS: *Justification, Certification, Requirements engineering, Quality requirements, Model-driven engineering*

1. Introduction

Dans de nombreux domaines où il existe des risques pour l'homme, comme la médecine, le nucléaire ou l'avionique, il est nécessaire de passer par une phase de certification visant à garantir le bon fonctionnement d'un système ou d'un produit. Cette certification est délivrée par une autorité, qui est généralement une tierce partie, mais pas toujours. Parfois une entreprise peut avoir l'autorité d'effectuer elle-même les activités de certification. Il est important de comprendre que les activités de certification ne se concentrent pas seulement sur le produit final, mais concernent tous les aspects du processus de production. Dans la pratique, la certification se fait en fonction de documents normatifs qui expriment les exigences auxquelles le produit et le processus de développement doivent se conformer. Par exemple, la norme applicable aux instruments médicaux (ISO 13485) décrit le processus de haut niveau qui doit être suivi à chaque étape de développement. Dans le cadre de cette norme, un audit de certification consiste, pour l'industriel, à produire une documentation selon laquelle le processus suivi est conforme à la norme. Ces documents peuvent être des explications logicielles, des résultats d'essais ou des protocoles d'essais cliniques suivis pendant les expériences. Dans les systèmes logiciels avioniques, l'autorité de certification suit le processus de développement de bout en bout et elle doit être convaincue que le processus et le produit sont conformes à la directive DO 178 et à l'ARP4754. Ainsi, une grande partie des activités d'accréditation est liée à la production de documents justificatifs, à une argumentation pour convaincre une autorité.

Pour répondre à cette nécessité de justification, il peut-être tentant de considérer comme éléments de justification tous les documents produits, peu importe leur but premier. Dès lors, une partie du travail de certification consiste à appréhender cette masse de justifications, qui est non structurée, parfois sans liens clairs avec les exigences du produit final et sans liens entre les documents eux-mêmes. Pour faire face à ce tsunami documentaire, Polacsek(Polacsek, 2016) propose un nouveau type de diagramme, le diagramme de justification, qui vise à structurer une argumentation de certification. Le but de tels diagrammes est de donner à l'autorité de certification une vue de l'ensemble de la justification explicitant clairement les liens de raisonnement entre les différents éléments et les arguments avancés pour prouver la conformité du produit avec les exigences propres à la certification. De plus, le diagramme de justification peut permettre d'exprimer, dans une vue synthétique, la liste des éléments de preuves et des moyens de conformité que doivent fournir les équipes de développement pour être conformes à une norme de certification.

Dans la pratique, les diagrammes de justification sont utilisés à plusieurs niveaux. Ils servent certes à organiser les documents de justification, mais ils servent aussi à représenter des patrons d'argumentation (Duffau *et al.*, 2017a), (Polacsek, 2017). Ces patrons consistent en des diagrammes génériques qui listent, pour une solution donnée, les éléments de preuves et la conclusion attendue. Les patrons d'argumentation sont ainsi conçus par des experts du domaine, dans la même veine que les patrons de conception en ingénierie des modèles (Alexander *et al.*, 1977), (Gamma *et al.*, 1995). Ici, pour chaque patron de justification, les experts définissent, pour un objectif donné,

quelles sont les preuves nécessaires et quelles sont les restrictions et les conditions d'utilisation dans la mise en œuvre d'un moyen de conformité.

De plus, un même élément de justification présent dans plusieurs patrons, peut être "*instancié*" avec plus ou moins de détail. Par exemple, dans le domaine médical, l'élément de justification que sont les documents d'analyse des risques servira à justifier la conformité aux normes de gestion des risques, mais servira aussi, dans une version beaucoup plus détaillée et spécifique, à justifier la gestion des risques dans toutes les parties techniques du projet et leurs normes associées. Dès lors, la distinction entre ces différents niveaux d'abstraction et les différents points de vue sur les justifications doit être soigneusement pris en compte.

Aujourd'hui, les diagrammes de justification ne posent pas clairement la distinction entre patron et réalisation de patron et le lien entre un élément générique et ses réalisations, instanciations, n'existe pas. Il apparaît donc comme nécessaire d'exprimer clairement, et sans ambiguïté, ces différents concepts. De plus, disposer d'une telle sémantique, permettrait de disposer d'outils de gestion de diagrammes de justification et de patrons qui s'appuient sur cette sémantique.

Le but de cet article est donc de définir une sémantique formelle pour les diagrammes de justification et plus précisément sur la relation de raffinement entre les patrons et les diagrammes finaux. Par ailleurs, nous présenterons une première implémentation d'un prototype de gestion de patrons de justification pour une aide à la certification, basée sur cette sémantique formelle. Dans cet article, en Section 2, nous présentons les concepts autour des diagrammes de justification ainsi que des travaux connexes autour des modèles de justification. En Section 3, nous introduisons la sémantique formelle pour les diagrammes de justification, sémantique qui nous permettra, en Section 4, de poser les pistes d'un outil pour l'aide à la création de diagrammes de justification. Pour finir, nous présentons les conclusions et perspectives de nos travaux en Section 5.

2. Justifier pour certifier

2.1. *Diagramme de Justification*

Les diagrammes de justification sont des diagrammes conceptuels permettant d'exprimer comment à partir d'un ensemble de faits on peut déduire une conclusion. Nous ne sommes pas ici dans le monde de la logique formelle, mais dans celui de l'explication. En effet, l'usage des diagrammes de justification est motivé par l'impossibilité d'être dans un monde formel. Ils capturent de bonnes pratiques, un pas de raisonnement accepté entre faits et conséquences, il relève d'une *"bonne rhétorique"* comme qualifiée par (Perelman et Olbrechts-Tyteca, 1958). Dérivée des travaux de l'argumentation légale de (Toulmin, 2003), leur représentation graphique s'inspire de la Goal Structured Notation (Kelly et Weaver, 2004), voir Figure 1. Les principaux concepts des diagrammes de justification sont :

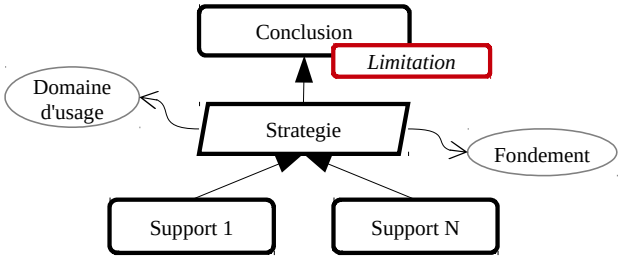


Figure 1. *Notation graphique d'un pas de justification inspirée de GSN*

– *Conclusion* explicite ce qui est soutenu pas la justification. C'est la conclusion du raisonnement, ce que nous cherchons à justifier.

– *Support* est un fait, une donnée, une hypothèse, etc. Ce sont les élément sur lesquels s'appuie la justification de la conclusion. Un support peut également être une *sous-conclusion* i.e. une conclusion d'une autre justification qui vient ici contribuer en tant que support.

– *Strategie* correspond à la méthode utilisée pour établir la connexion entre les supports et la conclusion.

– *Limitation* est une restriction à la conclusion. Elles sont séparées de la conclusion car elles n'ont vocation qu'à disparaître soit dans une justification de plus haut niveau, soit plus tard si les diagrammes s'inscrivent dans un processus temporel.

– *Fondement* est une explication de pourquoi une Stratégie est applicable. Si une stratégie consiste à suivre un processus défini par une norme, alors la Stratégie est l'application de la norme et le Fondement est l'explication de pourquoi la norme est applicable.

– *Domaine d'usage* donne les conditions précises d'usage et les limitations de la Stratégie. Si nous prenons l'exemple de l'application d'une norme, alors le domaine d'usage décrira dans quel contexte et pour quel besoin cette norme est applicable.

Même si, dans le cadre de la certification, l'usage de diagrammes de justification est prometteur, il faut tout de même souligner que, pour le moment, ce n'est qu'un système de notation. S'ils permettent de clarifier les différents concepts nécessaires à une justification, ils ne sont associés à aucun outil informatique et, par conséquent, toutes les opérations de vérification doivent être réalisées par un être humain. Avoir une définition formelle de propriétés souhaitables permettrait de disposer d'un outil automatique d'aide à la création et à la relecture de diagrammes de justification.

De plus, les différentes expérimentations d'utilisation, que cela soit dans le logiciel embarqué comme dans le médical, ont fait apparaître le besoin de patrons de justification. Cependant, les liens entre patron et réalisation de ces patrons, ainsi que les liens

possibles entre les patrons ne sont pas explicités et sont laissés à interprétation. Il est donc nécessaire de clarifier ces différents liens.

2.2. *Travaux connexes*

2.2.1. *Un besoin d'explication*

Le problème de fournir une explication pour convaincre qu'un résultat est bien fondé ne se pose pas uniquement dans le contexte de la certification. Au début des années quatre-vingt-dix, avec la montée des systèmes intelligents tels que les systèmes experts, les systèmes d'aide à la décision et les systèmes de prévision, de nombreux travaux ont insisté sur la nécessité de fournir une explication pour qu'un humain accepte un résultat automatique (Ass, 1992) (Int, 1993).

Dans ce domaine, des travaux comme ceux de (Chandrasekaran *et al.*, 1989) et (Southwick, 1991) soulignent le fait que les explications fournies par un système intelligent peuvent être classées en trois catégories. Tout d'abord, une trace des informations, les règles, ainsi que les étapes qui ont permis au système de produire le résultat correspondent à des explications, appelées *trace explications*. Deuxièmement, les *explications stratégiques* expliquent pourquoi une information est utilisée avant une autre, comment les différentes étapes du raisonnement sont choisies et comment elles contribuent à atteindre les objectifs principaux. Les explications stratégiques fournissent une explication du fonctionnement général du système. Troisièmement, les *explications profondes* justifient les fondements du système. Ces explications fournissent les théories à partir desquelles le résultat a été généré, les logiques sous-jacentes et les justifications de la base de connaissances.

Dans le cadre de la certification, nous sommes typiquement dans les explications profondes et, partiellement, dans les explications stratégiques. Ce qui est conforme avec (Ye et Johnson, 1995) qui explique que ce sont les explications profondes qui induisent la meilleure acceptation du système par des utilisateurs et la certification vise justement à convaincre une autorité. Notons ici qu'il existe des liens forts entre explication, et plus précisément explication profonde, et le schéma de Toulmin, comme le montre (Ye, 1995), schéma de Toulmin qui est le fondement des diagrammes de justification comme montré en Figure 1.

2.2.2. *Des modèles de justification*

Un des standards de modélisation pour la justification, appelé SACM (OMG, 2013), est un norme portée par l'OMG qui cherche à établir le méta-modèle de différents modèles d'ingénierie des exigences pour la qualité : les cas d'assurance structurés. Un cas d'assurance structuré est un argument structuré, supporté par un ensemble d'évidences, permettant de donner un cadre compréhensif et valide de sûreté et d'applicabilité d'un système dans un environnement donné (Weinstock et Goodenough, 2009). Différentes modélisations concrètes co-existent comme Goal-oriented Structured Notation (GSN) (Kelly et Weaver, 2004) ou Claim-Argument-Evidence (CAE)

(Emmet et Cleland, 2002). Ces deux modèles visent à expliciter la satisfaction de propriétés sur la sûreté d'un système. GSN et CAE bien que basés sur le modèle de Toulmin abordent celui-ci en rendant optionnel l'expression de la stratégie qui explicite le passage des supports de l'argumentation à la conclusion. Une partie du raisonnement est ainsi perdue. De plus, il est difficile d'utiliser ces modèles dans des cadres différents des propriétés de sûreté. C'est pour ces raisons qu'il est apparu crucial pour la communauté de pouvoir monter d'un niveau d'abstraction supplémentaire pour capturer un cadre générique pour les cas d'assurance structurés à travers SACM. Le métamodèle proposé par SACM repose sur 3 aspects distincts : le modèle argumentaire, le modèle d'artefacts et le modèle de cas d'assurance. Le modèle argumentaire définit une logique d'assertions permettant de représenter des arguments liés entre eux par des liens typés permettant ainsi d'établir des inférences entre arguments comme l'ajout d'un contexte applicatif à un argument. Le modèle d'artefacts permet de formaliser des données factuelles représentées comme des artefacts (évènement, participant, technique, ressource, etc). Ces artefacts sont les éléments basiques sur lesquels des raisonnements peuvent être faits et ainsi mener à une argumentation. Pour finir le modèle de cas d'assurance permet de décorer, enrichir le modèle argumentaire en ajoutant des notes, des contraintes, la gestion de la langue. Ce canevas permet ainsi de définir toutes les couches pour un cadriciel d'exigences pour la qualité, des faits jusqu'au raisonnement menant à une revendication en passant par la couche de présentation de ce raisonnement. Ce meta-modèle a été utilisé dans diverses approches par la communauté pour en définir un langage utilisable à travers Eclipse pour editer et visualiser des diagrammes basés sur GSN (Matsuno, 2014) ou concevoir des exigences pour la certification (de la Vara et Panesar-Walawege, 2013).

Dans le domaine de l'ingénierie des exigences, i* est un langage qui propose de capturer les besoins très en amont depuis l'intérêt des parties prenantes jusqu'aux différentes variantes possibles du système à réaliser (Yu, 1997). Dans la nouvelle version i* (Dalpiaz *et al.*, 2016), un des changements notables est le remplacement des *"softgoal"*, qui permettaient de capturer des exigences non fonctionnelles, par le concept de *"qualité"*. Cette évolution ouvre la porte à une utilisation de i* dans le domaine des exigences qualité. Néanmoins, le langage reste très orienté processus, puisqu'il manipule principalement les notions d'acteurs, de ressources et de tâches. I* permet d'exprimer ce qui contribue ou empêche l'atteinte d'une qualité, mais ne permet pas de justifier cette contribution en s'appuyant sur du raisonnement et des artefacts de qualité. Cependant, pour palier cette limitation, (Van Zee *et al.*, 2016) proposent d'attacher de l'argumentation pour expliciter le raisonnement d'un diagramme i*. Plus précisément, il propose de rajouter une représentation des arguments et contre-arguments proposés par les différentes parties prenantes qui se sont prononcées pour, ou contre, un élément du modèle exprimé en i*.

Par rapport à ces travaux, les diagrammes de justification se situent a posteriori, il n'y a plus de pour et de contre, mais une explication rationnelle du choix final en vue de le certifier ou a minima de le justifier. Si nous prenons pour exemple la réservation d'un déplacement dans un cadre professionnel, dans un diagramme i*, nous trouverons des éléments relatifs aux tâches (acheter des billets, réserver une chambre), des

ressources (budget), des acteurs (agences de voyage), des objectifs (billets réservés) et des qualités (réservation rapide). Un diagramme de justification va lui se focaliser sur la preuve que des objectifs ont été atteints, par exemple, que les billets d'avion et la réservation de l'hôtel correspondent, ou que le déplacement est accepté (validé par le supérieur hiérarchique, conforme au budget, etc.).

3. Une sémantique formelle pour les diagrammes de justification

Dans cette section, nous allons présenter une sémantique formelle pour les diagrammes de justification. Nous nous sommes concentrés sur les éléments clés du schéma de justification et nous ne détaillons pas ici la *Limitation*, le *Domaine d'usage* et le *Fondement*. Concernant le *Domaine d'usage* et le *Fondement*, ils peuvent être vus comme un ajout à la *Stratégie* et donc être contenus, embarqués, dans la représentation que nous allons en donner. Concernant la *Limitation*, il est aussi possible de la voir comme une commodité d'écriture, du sucre syntaxique, et pas comme un élément constitutif du langage.

3.1. *Concepts de base*

Les diagrammes de justification manipulent des assertions, des allégations, du type "*le système résiste à une panne*" ou "*les tests sont tous positifs*". Ces assertions correspondent aussi bien à la conclusion, une sous-conclusion, qu'à un élément de preuve. Elles regroupent donc les supports présentés en Section 2. Elles peuvent être exprimées avec une simple phrase en langage naturel ou être exprimées en langage plus formel, comme la logique. Une assertion peut même être un élément plus complexe, comme l'intégralité d'un rapport ou la sortie numérique d'un logiciel. Nous noterons l'ensemble des assertions $\mathcal{A}$.

Afin de pouvoir comparer certaines assertions entres elles, nous définissons une relation, notée $\mathcal{R}$, que nous qualifierons de relation de conformité. L'idée sous-jacente à cette notion de conformité est celle d'un "*raffinement*". Ainsi, si deux assertions sont des formules logiques, $\mathcal{R}$ pourra être vu comme "*est un modèle de*", si ce sont des classes comme "*est une spécialisation de*" et si nous sommes dans l'utilisation du langage naturel, alors la signification de la relation de conformité $\mathcal{R}$ devra être donnée.

Définition 1 (Relation de conformité)
Soit $\mathcal{A}$ l'ensemble des assertions, $\mathcal{R}$ est une relation de conformité ssi $\forall a_1, a_2, a_3 \in \mathcal{A}$:

- $a_1 \mathcal{R} a_1$
- *if* $a_1 \mathcal{R} a_2$ *and* $a_2 \mathcal{R} a_1$ *then* $a_1 = a_2$
- *if* $a_1 \mathcal{R} a_2$ *and* $a_2 \mathcal{R} a_3$ *then* $a_1 \mathcal{R} a_3$

Notons qu'une relation de conformité définie un ordre partiel sur l'ensemble des assertions, elle est réflexive, transitive et antisymétrique. D'après cette définition, plusieurs assertions peuvent être conformes à une même assertion. Prenons pour exemple

deux rapports de tests différents, rt_1 et rt_2, qui sont conformes à une norme définissant les rapports de tests $nt : rt_1 \mathcal{R} nt$ et $rt_2 \mathcal{R} nt$. De façon duale, une assertion peut aussi être conforme à plusieurs assertions. Pour exemple, la parole d'un expert, notée a_e, peut être conforme à l'assertion de très haut niveau "*jugement d'expert*", a_{je}, et être aussi conforme à l'assertion "*hypothèse de travail*", $a_h : a_e \mathcal{R} a_{je}$ et $a_e \mathcal{R} a_h$.

Nous définissons un type d'assertions particulières, les assertions terminales. Nous dirons qu'une assertion est un terminal si aucune assertion n'est conforme avec elle. Généralement, un terminal est un fait concret comme un résultat bibliographique, une bonne pratique établie, un point de la spécification ou le résultat de test.

Définition 2 (Terminal)
$a \in \mathcal{A}$ *est un terminal ssi* $\forall a' \in \mathcal{A}, a' \mathcal{R} a \rightarrow (a' = a)$.

Comme chez Toulmin, la strategie est la pierre angulaire des diagrammes de justification. C'est elle qui explicite, comment, à partir d'éléments de preuve, il est possible de déduire une conclusion. Cette déduction s'appuie sur un domaine d'usage et des fondements, mais n'est pas de l'ordre d'une déduction formelle, sinon il serait inutile d'utiliser les diagrammes de justification. Dès lors, nous avons choisi d'encapsuler tous les éléments de cette déduction dans un concept *Stratégie*. Nous notons l'ensemble des stratégies $\mathcal{S}$.

Pour finir, nous allons définir un pas de justification (*pdj*), en d'autres termes donner une sémantique à la notation graphique présentée Figure 1. Nous noterons l'ensemble des *pdj* : $\mathcal{P}$.

Définition 3 (Pas de justification (*pdj*))
Un pdj *(pas de justification)* p *est un tuple* $\langle supports, strategie, conclusion \rangle$ *où :*

– $supports$ *est un ensemble d'assertions* $\subset \mathcal{A}$,
– $strategie \in \mathcal{S}$;
– $conclusion \in \mathcal{A}$

3.2. *Patron*

A l'aide de la relation de conformité $\mathcal{R}$, il est possible d'étendre le concept de conformité aux pas de justification. Nous donnons ci-après une définition de la conformité entre *pdj* .

Définition 4 (Conformité *pdj*)
Un pdj $s = \langle supp, strat, c \rangle$ *est dit conforme à un* pdj $s' = \langle supp', strat', c' \rangle$ *ssi :*

– $\forall a \in supp, \exists a' \in supp'$, $a \mathcal{R} a'$,
– $\forall a' \in supp', \exists a \in supp$, $a \mathcal{R} a'$,
– $\forall a, b \in supp, \forall a' \in supp'$, *si* $a \mathcal{R} a'$ *et* $b \mathcal{R} a'$ *alors* $a = b$,
– $strat = strat'$,

– $c\mathcal{R}c'$.

La définition que nous donnons ici privilégie la préservation des concepts encapsulés dans les supports d'un pas de justification plutôt que la cardinalité. En effet, si nous considérons les *pdj* $p_1 =< \{s\}, w, c >$ et $p_2 =< \{s_1, s_2\}, w, c >$, et $s\mathcal{R}s_1$ et $s\mathcal{R}s_2$ d'après notre définition p_1 est conforme à p_2.

Prenons pour exemple un pas de justification pdj_1 qui permet de faire une déduction sur une simulation numérique avec les deux supports suivants : "*la température est inférieure à 90 degrés*" et "*la température est supérieure à 0 degrés*". Considérons maintenant un autre pas de justification pdj_2 avec la même stratégie et la même conclusion, mais un seul support qui est un document validé attestant que la température est bien comprise entre 10 et 50 degrés. Si l'on considère ce support comme une spécialisation des deux autres, comme relié par R aux deux autres, alors ce *pdj* est bien conforme au premier. Le pas de justification pdj_3 qui prend en support un document d_1 qui atteste que la température est comprise entre 10 et 25 pourra alors être également conforme à pdj_2 et pdj_1.

Si nous voulons prendre en compte la cardinalité, c'est-à-dire ne considérer comme conforme entre eux que des *pdj* qui ont le même nombre de supports, nous devons considérer que la relation $\mathcal{R}$ est, pour l'ensemble des supports des deux *pdj* , une relation bijective, en d'autres termes, rajouter la condition suivante : $\forall a \in supp, \forall a', b' \in supp'$, si $a\mathcal{R}a'$ et $a\mathcal{R}b'$ alors $a' = b'$.

Notons que la relation de conformité entre les *pdj* , quelle que soit la définition choisie, est, elle aussi, une relation d'ordre partiel. La démonstration est triviale puisque cette nouvelle conformité est simplement basée sur la relation $\mathcal{R}$ entre les assertions. Par commodité nous noterons $s'\mathcal{R}s$, le fait que le $pdjs'$ est conforme au $pdjs$.

Propriété 1 *La relation de conformité entre pas de justification est réflexive, transitive et antisymétrique.*

Maintenant que nous disposons d'une relation entre les *pdj*, il nous est possible de définir formellement ce que sont un patron de justification, un raffinement de patron et une concrétisation de patron.

Définition 5

– $\forall s \in \mathcal{P}$, *s est un patron de justification ssi* $\exists s' \in \mathcal{P}, s'\mathcal{R}s$ *et* $s \neq s'$.

– $\forall s, s' \in \mathcal{P}$, *s raffine* s' *ssi* $s\mathcal{R}s'$.

– $\forall s \in \mathcal{P}$, *s est une concrétisation de patron de justification ssi* $\forall s' \in \mathcal{A}, s'\mathcal{R}s \rightarrow (s' = s)$.

Relativement à l'exemple donné précédemment : pd_1 et pd_2 sont des patrons. pd_2 raffine pd_1. pd_3 peut être appréhendé comme une concrétisation si aucun support ne peut raffiner le document d_1.

4. Vers un outil d'aide à la certification

4.1. *Utiliser des patrons d'experts*

Notre idée est de pouvoir disposer d'un ensemble de patrons de justification dédiés à un domaine. Cet ensemble de patrons forme une bibliothèque dans laquelle il est possible de venir piocher pour pouvoir guider et structurer la justification de conformité, dans le cadre de la réalisation d'un produit. Les patrons peuvent donc être vu comme des guides, listant les éléments nécessaires à la justification d'un objectif de certification.

La réalisation de cette bibliothèque ne peut être faite que sur la base d'experts qui vont définir les patrons de justification en fonction de leur connaissances, de bonnes pratiques établies, de normes, d'exigences de qualité, etc.

Dès lors, à la réalisation du produit, construire une justification pour un objectif fixé consiste à construire un diagramme de justification composé de *pdj* terminaux conformes aux patrons de la bibliothèque.

Notre but est donc de faciliter la construction des diagrammes de justifications. Cette construction peut être manuelle et/ou programmative en fonction du domaine ciblé, de l'environnement de développement et de la justification elle-même. Ainsi lorsque les justifications portent sur des tâches qui échappent au contrôle du système informatique, il appartient aux experts de construire les pas de justification. Inversement, dans un contexte de justification d'un processus pris en charge par le système informatique, il doit être possible de construire les pas de justification de manière automatique. En utilisant les patrons comme un guide, les outils doivent alors faciliter la construction incrémentale des pas de justification, permettre l'identification des assertions ou *pdj* manquants, forcer la vérification des conformités.

4.2. *La relation $\mathcal{R}$ dans la pratique*

La relation de conformité $\mathcal{R}$ est un élément clé de notre sémantique formelle. C'est pour pouvoir rester générique et accepter tout type d'assertion, que nous avons choisi de ne pas caractériser son comportement. Cependant, dans le cadre d'un logiciel, il nous faut faire des choix d'implémentation de $\mathcal{R}$. Comme nous l'avons précédemment évoqué, la relation $\mathcal{R}$ peut être vue comme un raffinement, une spécialisation, etc. Nous avons choisi de représenter ce lien au travers d'une procédure de décision qui permet de savoir si deux assertions sont reliées ensemble. De façon pratique, cela consiste en une méthode qui implémente la sémantique de $\mathcal{R}$. Cette sémantique est forcément dépendante du format choisi pour représenter les assertions. Ainsi, si les assertions sont représentées en logique, il est possible de considérer la conformité comme : "*est conséquence logique*", si les assertions sont dans un langage contrôlé (boilerplate), la conformité peut être une relation d'instanciation et dans le cadre d'assertion en langue naturelle, la conformité sera un algorithme réalisant du traitement automatique de la langue. C'est donc pour rester aussi générique que possible que nous avons décidé d'encapsuler la sémantique de $\mathcal{R}$ dans une méthode : $isConform$.

4.3. *Une aide automatique*

Le but de notre démarche est de proposer un outil d'assistance à la création de diagrammes de justification. Nous allons présenter ici un ensemble de fonctionnalités que devrait fournir un tel outil et, pour chaque fonctionnalité, exprimer ce qu'elle fait à l'aide de notre sémantique formelle.

4.3.1. *Comment justifier une conclusion ?*

Considérons une assertion, l'outil renvoie tous les patrons qui permettent de justifier cette assertion. D'un point de vue de la sémantique cela revient, pour une assertion $goal$ donnée, à retourner tous les $pdj \in \mathcal{P}$ avec $pdj =< S, w, c >$ tel que : $goal \mathcal{R} c$.

4.3.2. *Que peut-on justifier ?*

A partir d'un ensemble d'assertions, il s'agit de déterminer tous les *pdj* qui peuvent être appliqués. En d'autres termes, pour un ensemble d'assertions A, il faut trouver tous les $pdj \in \mathcal{P}$ avec $pdj =< S, w, c >$ tel que : $\forall s \in S, \exists a \in A, a \mathcal{R} s$.

4.3.3. *Le diagramme est-il constitué de concrétisations de patron ?*

Il peut être intéressant de considérer qu'un diagramme de justification a été construit sans assistance automatique. Dès lors, un outil automatique devrait pouvoir vérifier que (1) chaque pas de justification du diagramme est conforme à un patron et (2) que tous les pas de justifications du diagramme ne sont pas des patrons mais bien des concrétisations de patron. Sémantiquement, cela correspond à vérifier que pour tout pdj du diagramme :

(1) $\exists p \in \mathcal{P}, pdj \mathcal{R} p$ et $pdj \neq p$;

(2) pdj est une concrétisation de patron.

4.3.4. *Quels sont les pas de justification non conformes à des patrons ?*

Inversement, il peut être intéressant de déterminer quels sont les *pdj* qui ne sont pas conformes à des patrons. Sémantiquement, cela correspond à déterminer quels sont les pdj du diagramme tels que $\nexists p \in \mathcal{P}, pdj \mathcal{R} p$.
L'ensemble des *pdj* non conformes à des patrons nécessite une étude approfondie pour confirmer la confiance qui leur est attribuée ou pour les éliminer comme non utiles à la justification. Les *pdj* utiles sont les *pdj* conformes à des patrons ou ceux qui conduisent à une sous-conclusion utilisée directement ou indirectement comme support dans un *pdj* conforme à un patron. Par expérience lors de deux études de cas industriels (Duffau *et al.*, 2018), ces *pdj* apparaissent lorsque les attentes des patrons nécessitent des supports qui ne peuvent pas être obtenus directement et qu'il convient donc de justifier.

Sémantiquement, déterminer les pas *utiles* revient à trouver tous les *pdj* dans le diagramme tels que

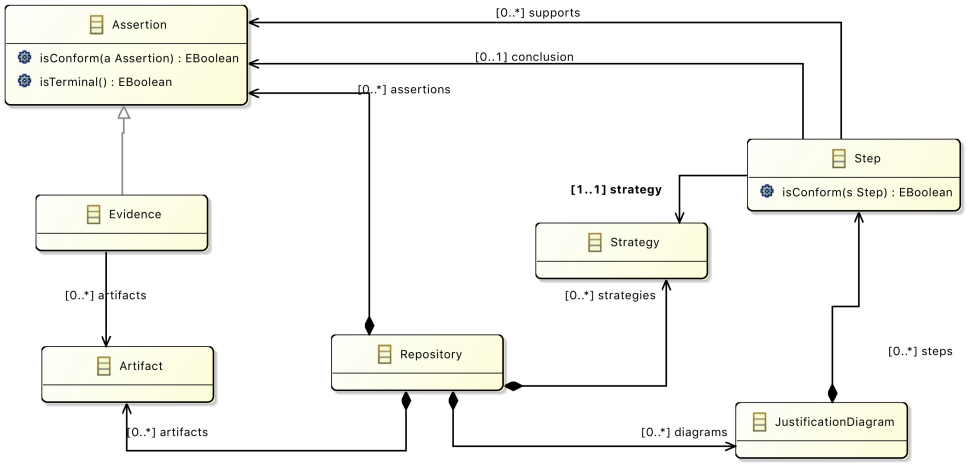


Figure 2. *Extrait du méta-modèle pour les diagrammes de justifications*

(1) $\exists p \in \mathcal{P}, pdj \mathcal{R} p, pdj \neq p$, ou
(2) $(pdj =< S, w, c >, \exists pdj\prime \in P, pdj\prime =< S\prime, w\prime, c\prime >, c \in S\prime)$ et pdj' est utile.

4.3.5. *Quels sont les supports réutilisables pour justifier d'une conclusion ?*

A partir d'un patron permettant d'atteindre une conclusion particulière, il peut être intéressant de savoir s'il existe des supports déjà présents dans un diagramme qui sont des raffinements des supports du patron. Ces supports sont soit des éléments de preuves, des faits, des données, etc. soit des conclusions de *pdj*, en d'autres termes ce sont des terminaux présents dans le diagramme de justification.

Dès lors, déterminer les supports réutilisables pour justifier d'une conclusion présente dans un patron, revient à rechercher toutes les assertions présentes dans le diagramme qui raffinent les supports du patron.

Un exemple d'une telle utilisation, pourrait être la justification de l'exécution des tests d'intégration qui a besoin que les tests unitaires aient bien été exécutés préalablement, point qui se trouve être déjà justifié par un pdj qui a été construit automatiquement comme l'illustre la Figure 3.

4.4. *Vers un outil*

Dans le méta-modèle pour les diagrammes de justification en Figure 2, nous retrouvons les concepts de base définis par le formalisme. Nous retrouvons la méthode $isConform$ qui matérialise la relation $\mathcal{R}$ entre *Assertion* ou entre *Step*. De manière identique, le concept d'assertion terminale a été implémenté par une méthode

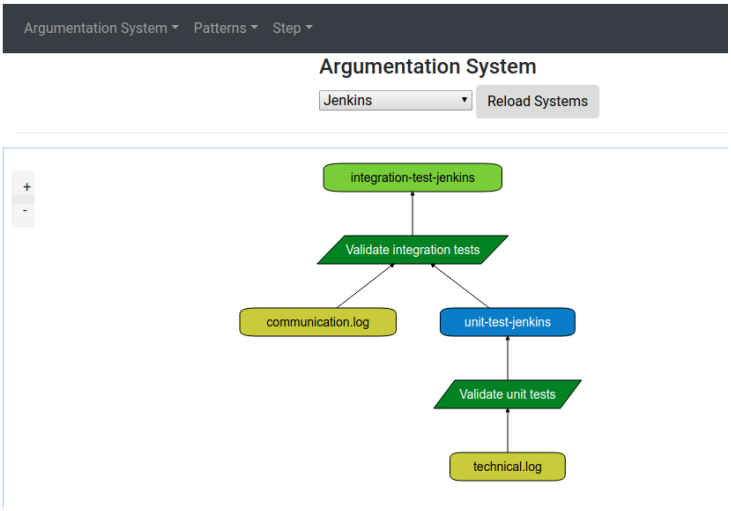


Figure 3. *Exemple d'un patron de diagrammes, qui justifie la bonne exécution de tests logiciels à travers Jenkins, conçu avec le logiciel ADEV*

$isTerminal$. Dans la pratique, les *Assertion*, *Strategy*, *Artifact*, *JustificationDiagram* sont conservés dans un *Repository* pour permettre de supporter le flot de construction, d'un pas dans un patron jusqu'à la concrétisation d'un diagramme de justification. Un *pdj* (dans la figure *Step*) est bien composé d'un tuple de *supports*, *strategy*, *conclusion*. Ce modèle permet bien de capturer la notion de *sous-conclusion* à travers l'utilisation d'assertions pouvant être utilisées comme une *conclusion* ou un *support*.

Ce modèle est au coeur du moteur d'argumentation intégré dans une usine nommée *Argumentation Factory* (Duffau *et al.*, 2017a). Cette usine expose trois services [1] permettant d'enregistrer des patrons, de construire des diagrammes et de vérifier des propriétés sur ces diagrammes. Afin de faciliter la création des patrons et pas de justification, un outil *What You See Is What You Get* a été développé, présenté en Figure 3. Cet outil interagit avec l'*Argumentation Factory* à travers les services qu'elle expose et est donc l'interface graphique proposée pour la visualisation et édition des diagrammes de justifications. Suite à ceci, grâce à la vision service mise en place, la réalisation d'un diagramme ainsi que la vérification de propriétés avec l'Argumentation Factory sont facilitées et peuvent être facilement intégrés à d'autres outils. Duffau (Duffau *et al.*, 2017b) propose, par exemple, son utilisation à travers un plugin pour Jenkins [2].

1. au niveau technologique, ces services sont des services web REST et le moteur sous-jacent est développé en Java
2. Jenkins est la plateforme d'intégration continue communément utilisée en industrie permettant de compiler, tester, déployer automatiquement des produits logiciels à partir des code sources.

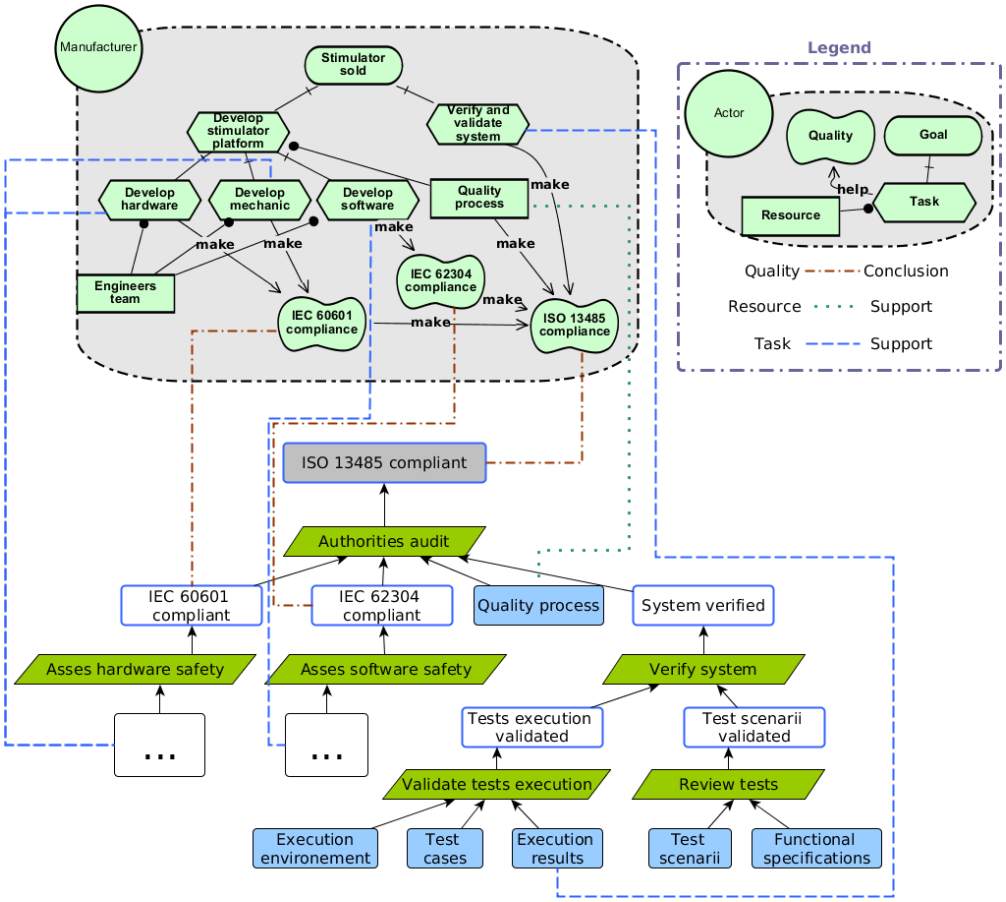


Figure 4. *Exemple d'interaction possible entre i* et les diagrammes de justifications*

5. Conclusion et perspectives

Le formalisme des diagrammes de justification a déjà été mis en oeuvre à travers un outil. Nous avons utilisé les diagrammes de justification, sur un cas industriel dans le domaine des technologies médicales. Dans cette étude, nous avons outillé un moteur d'intégration continue par appel aux services dédiés de l'*Argumentation Factory* permettant ainsi de justifier l'ensemble des processus de développement : d'une spécification à travers un ticket jusqu'à la validation de celle-ci par des tests exécutés par la plateforme d'intégration continue (pour plus de détails voir (Duffau *et al.*, 2017b)).

Dans la suite de nos travaux, nous comptons compléter les aspects théoriques des diagrammes de justification et dans le même temps étendre leurs usages sur d'autres applications industrielles. Ainsi parmi les applications possibles, nous avons com-

mencé à étudier l'utilisation des diagrammes de justification pour la certification de processeurs pluri-cœurs dans le cadre avionique (Bieber *et al.*, 2018). Là encore, nous cherchons à définir des patrons génériques qui permettent in fine de justifier des objectifs haut niveau, objectifs étant définis par les documents de certification.

Concernant la partie plus théorique, nous pouvons désormais axer la suite de nos travaux sur la collaboration avec d'autres modèles. Par exemple, le langage i* définit des types de contribution (crée, aide, nuit, casse) qui permet de définir des liens de contribution entre une qualité et une autre qualité, une tâche ou un objectif. Puisque le langage propose ce mécanisme, nous proposons de nous brancher sur i* pour permettre de justifier ces qualités notamment à travers ces types de contributions. Un exemple d'une telle interaction est donnée en Figure 4. Le diagramme i* présente les exigences permettant d'atteindre l'objectif de vendre des neurostimulateurs en prenant en compte des qualités comme la conformité avec la norme ISO 13485. L'atteinte de cette qualité est justifiée par le diagramme de justification du dessous. Les liens en pointillés entre les deux diagrammes matérialisent les points d'interaction possibles entre ces deux domaines. A travers cet exemple, nous illustrons la complémentarité de ces deux approches qui permet ainsi d'avoir une plus grande confiance dans l'atteinte des qualités souhaitées.

L'établissement de liens entre les diagrammes de justification et SACM nous permettent déjà d'envisager de vérifier les propriétés que nous avons présentées ici sur ces modèles. Modulo une difficulté liée à la multiplicité des conclusions dans SACM, nous pensons pouvoir définir une relation $\mathcal{R}$ propre aux modèles SACM, et ainsi introduire sémantiquement la notion de patron qui est aujourd'hui diffuse dans plusieurs méta-éléments. C'est une de nos perspectives à court terme.

6. Bibliographie

Alexander C., Ishikawa S., Silverstein M., i Ramió J. R., Jacobson M., Fiksdahl-King I., *A pattern language*, Gustavo Gili, 1977.

Association for the Advancement of Artificial Intelligence, *Proceedings of the the AAAI Spring Symposium on producing cooperative explanations*, 1992.

Bieber P., Boniol F., Bouchebaba Y., Brunel J., Pagetti C., Poitou O., Polacsek T., Santinelli L., Sensfelder N., « A model-based certification approach for multi/many-core embedded systems », *ERTS*, 2018.

Chandrasekaran B., Tanner M. C., Josephson J. R., « Explaining Control Strategies in Problem Solving », *IEEE Intelligent Systems*, vol. 4, 1989, p. 9-15, 19-24, IEEE Computer Society.

Dalpiaz F., Franch X., Horkoff J., « iStar 2.0 Language Guide », *CoRR*, vol. abs/1605.07767, 2016.

de la Vara J. L., Panesar-Walawege R. K., « Safetymet : A metamodel for safety standards », *International Conference on Model Driven Engineering Languages and Systems*, Springer, 2013, p. 69–86.

Duffau C., Camillieri C., Blay-Fornarino M., « Improving Confidence in Experimental Systems through Automated Construction of Argumentation Diagrams », *ICEIS 2017*, vol. 1, 2017, p. 495–500.

Duffau C., Grabiec B., Blay-Fornarino M., « Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation : Application in a Healthcare Company », *ISSREW 2017*, 2017.

Duffau C., Polacsek T., Blay-Fornarino M., « Support of Justification Elicitation : Two Industrial Reports », *Advanced Information Systems Engineering - 30th International Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018. Proceedings*, Lecture Notes in Computer Science, Springer, 2018.

Emmet L., Cleland G., « Graphical notations, narratives and persuasion : a Pliant Systems approach to Hypertext Tool Design », Blustein J., Allen R. B., Anderson K. M., Moulthrop S., Eds., *HYPERTEXT 2002, Proceedings of the 13th ACM Conference on Hypertext and Hypermedia*, ACM, 2002, p. 55–64.

Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns : Elements of Reusable Object-oriented Software*, Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1995.

International Joint Conferences on Artificial Intelligence, *Proceedings of the IJCAI'93 Workshop on Explanation and Problem Solving*, 1993.

Kelly T., Weaver R., « The Goal Structuring Notation /- A Safety Argument Notation », *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004.

Matsuno Y., « A design and implementation of an assurance case language », *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, IEEE, 2014, p. 630–641.

OMG, « Structured Assurance Case Meta-model », 2013.

Perelman C., Olbrechts-Tyteca L., *Traité de l'argumentation : La nouvelle rhétorique*, Presses universitaires de France, 1958.

Polacsek T., « Validation, accreditation or certification : a new kind of diagram to provide confidence », *Research Challenges in Information Science*, IEEE, 2016.

Polacsek T., « Diagramme de justification. Un outil pour la validation, la certification et l'accréditation », *Ingénierie des Systèmes d'Information*, vol. 22, n° 2, 2017, p. 95–119.

Southwick R. W., « Explaining reasoning : an overview of explanation in knowledge-based systems », *Knowledge Eng. Review*, vol. 6, n° 1, 1991, p. 1–19.

Toulmin S. E., *The uses of argument*, Cambridge University Press, 2003.

Van Zee M., Marosin D., Bex F., Ghanavati S., « RationalGRL : A Framework for Rationalizing Goal Models Using Argument Diagrams », *Conceptual Modeling : 35th International Conference, ER 2016*, Springer, 2016, p. 553–560.

Weinstock C. B., Goodenough J., « Towards an assurance case practice for medical devices », rapport, 2009, Software Engineering Institute.

Ye L. R., Johnson P. E., « The Impact of Explanation Facilities in User Acceptance of Expert System Advice », *MIS Quarterly*, vol. 19, n° 2, 1995, p. 157–172.

Ye L. R., « The value of explanation in expert systems for auditing : An experimental investigation », *Expert Systems with Applications*, vol. 9, n° 4, 1995, p. 543–556, Elsevier.

Yu E. S., « Towards modelling and reasoning support for early-phase requirements engineering », *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, IEEE, 1997, p. 226–235.

Alignement des points de vue du système d'information

Une approche pragmatique

Jonathan Pepin [1,2], **Pascal André** [1], **Christian Attiogbé** [1], **Erwann Breton** [2]

1. *LS2N CNRS UMR 6004 Université de Nantes 2, rue de la Houssinière, BP 92208, F-44322 Nantes Cedex 3*
   Prenom.Nom@univ-nantes.fr
2. *Mia-Software - Nantes 11, rue Nina Simone 44009 Nantes Cedex 1*
   ebreton@sodifrance.fr

RÉSUMÉ. *La maintenance de systèmes d'informations implique de mettre en correspondance la vision stratégique du métier et la vision informatique, parfois via le prisme de l'urbanisation. La distance sémantique entre les points de vue rend difficile la mesure de l'impact de l'évolution du parc applicatif vis-à-vis des processus métiers ou des technologies. Nous proposons une approche pragmatique pour rapprocher les points de vue et aider à évaluer l'impact de restructurations sur l'évolution du parc applicatif. Une fois alignés les modèles des deux points de vue, des mesures estiment la qualité de l'alignement. L'approche présentée est mise en œuvre par des transformations de modèles et expérimentée sur des cas concrets.*

ABSTRACT. *The maintenance of Information Systems involves to fit the business vision with the Information Technology vision, possibly with the enterprise architecture. The semantic distance between viewpoints makes it difficult to evaluate the impact of the application portfolio evolution with respect to business processes or technologies. We propose a pragmatic approach to align the different viewpoints and help to estimate the impact of restructuring actions on the legacy application portfolio. Once models are aligned from both points of view, we propose measurements that estimate the quality of the alignment. This approach has been implemented by model transformations and experimented on concrete cases.*

MOTS-CLÉS : *Systèmes d'information - Architecture d'entreprise - Rétro-ingénierie - Alignement Métier - Ingénierie des modèles - Mesure*

KEYWORDS: *Information Systems, Enterprise Architecture, Reverse Engineering, Business-IT Alignment, Model Driven Engineering*

1. Introduction

Dans un système d'information (SI), la cohérence entre l'organisation et le système informatique qui implante ses processus automatisés est fondamentale du point de vue de la qualité du SI. On parle d'*alignement business/IT* lorsque les moyens mis en œuvre (pas seulement l'informatique) sont cohérents avec la stratégie d'entreprise et contribuent efficacement à sa compétitivité (Henderson, Venkatraman, 1993). Cet alignement contribue à la performance de l'organisation (Thevenet, Salinesi, 2007 ; Ullah, Lai, 2013 ; Roelens *et al.*, 2017). Face à la concurrence, les entreprises raccourcissent le cycle de décision et exigent une forte réactivité du SI alors que le cycle de maintenance et de renouvellement du parc applicatif, souvent hétérogène, est plus long du fait de contraintes budgétaires ou organisationnelles. Cet alignement est un défi lancé depuis plus de 20 ans avec l'article fondateur (Henderson, Venkatraman, 1993) et qui reste d'(actualité (Coltman *et al.*, 2015). Il a évolué au cours du temps notamment avec l'émergence de l'*architecture d'entreprise* (ou *urbanisation*).

La démarche d'architecture d'entreprise permet de faire évoluer le système d'information d'une entreprise au rythme des stratégies à appliquer pour faire progresser l'activité de l'entreprise. Des plans de transitions (ou plans de migrations) se mettent en place selon un cycle itératif permettant l'évolution de sous-ensembles ciblés du système d'information (Lankhorst, 2013). La cartographie vise à obtenir une image fidèle de l'état actuel des sous-ensembles du système d'information. Elle est constituée de *points de vue* du système d'information, notamment les points de vue des domaines *métier* et *informatique*. Les points de vue ont des objectifs, des méthodes, des acteurs, des représentations ou des pratiques variées. Les points de vue sont complémentaires, couvrant des aspects différents (composition), ou similaires mais avec des niveaux de détail différents (abstraction). Par exemple, il est nécessaire de structurer les différentes sources, support de la cartographie, hétérogènes par nature. D'une part, la stratégie et le fonctionnement de l'entreprise sont renseignés à travers des documents le plus souvent informels et dans le meilleur des cas à travers une modélisation métier décrite dans un langage plus ou moins formalisé et normé. D'autre part, le patrimoine applicatif est composé de programmes et d'infrastructures physiques. Sans cartographie applicative, la documentation est constituée au mieux de modèles d'architecture. Le cas échéant le code source est un point d'entrée possible, mais sa densité nécessite une abstraction méthodique des concepts techniques par détection ou filtre. Il peut être difficile de garder ces points de vue cohérents au fil des **maintenances**, des *changements technologiques* (mobilité, géolocalisation, objets connectés…) et des *évolutions stratégiques et organisationnelles* (lois, marché concurrentiel, actionnariat, économie collaborative, préoccupations environnementales…). L'évolution des points de vue du SI est un problème complexe et un enjeu important pour l'entreprise. Elle permet à l'entreprise d'être à la fois *performante et flexible*. Pour suivre et piloter l'évolution du SI, la démarche d'architecture d'entreprise est un projet qui mobilise un temps conséquent et du personnel en nombre ce qui peut *engendrer des coûts* importants.

Les travaux présentés ici s'inscrivent dans les étapes de cartographie et d'identification de la trajectoire du SI. Aligner de bout en bout chaque élément du SI, du code

binaire déployé jusqu'à la stratégie d'entreprise, reste encore une gageure. Notre objectif est de proposer une *méthode pragmatique d'alignement par les modèles* pour les systèmes d'information existants qui nécessitent des évolutions. L'alignement *business/IT* n'est qu'une étape de la démarche d'architecture d'entreprise et d'urbanisation (Ullah, Lai, 2013) mais elle est cruciale du point de vue de l'évolution du SI. Dans le cycle de la méthode de développement d'architecture (*Architecture Development Method, ADM*) du cadre d'architecture TOGAF (*The Open Group Architecture Framework*) présenté par la FIGURE 1, nos travaux assisteraient à l'accomplissement des étapes notées B, C, D et E. Notre méthode contribue à l'alignement entre les

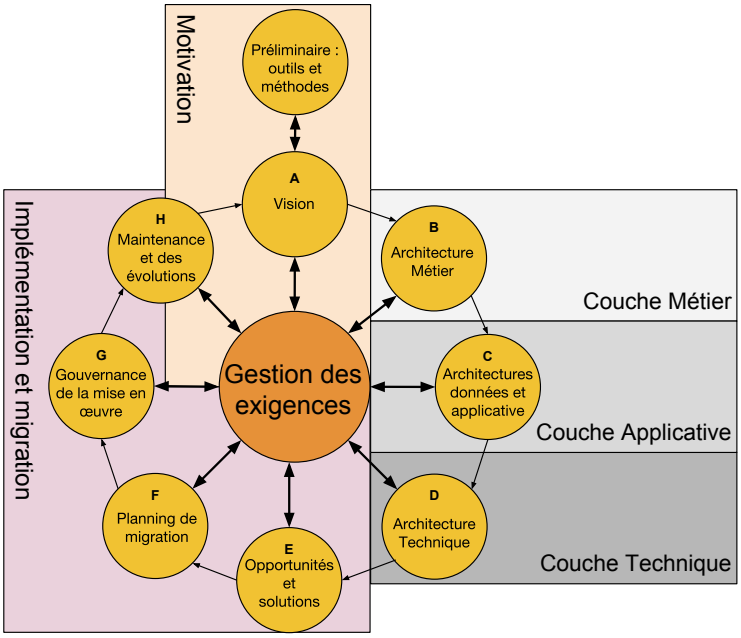


Figure 1. Le cycle ADM du cadre d'architecture d'entreprise TOGAF

visions métier et technique (*business/IT*) pour maintenir le patrimoine applicatif en phase avec l'évolution des métiers. Nous avions exposé la méthode dans (Pepin *et al.*, 2015). Nous en présentons ici l'évaluation et l'extension aux couches hautes et basses de l'architecture d'entreprise.

Cet article est organisé comme suit. Dans la section 2 nous exposons notre approche opérationnelle de l'alignement des visions métier et technique ; cette approche est orientée modèles et le cœur du problème concerne les vues métier, applicative et fonctionnelle. Nous proposons une solution non-intrusive par tissage. L'évaluation de l'alignement est discutée dans la section 3. Notre approche est outillée et a été expérimentée sur plusieurs cas concrets relatés dans la section 4. Le cœur opérationnel est étendu dans sa partie basse à l'infra-structure et dans sa partie haute à la stratégie; nous en discutons la vision dans la section 5. La section 6 résume l'état des travaux et trace des perspectives.

2. Une approche pragmatique au cœur de l'alignement

Notre problème est de définir une méthode d'alignement qui s'applique aux systèmes d'information existants (*legacy systems*) qui nécessitent des évolutions. Nous ne considérons que la dimension alignement *business-IT* de (Henderson, Venkatraman, 1993). Dans une vision classique, cet alignement est interprété comme une ligne de traçabilité traversant les couches (partie gauche de la FIGURE 2) mais comme indiqué en introduction aligner de bout en bout reste encore une gageure. L'expérimentation et la pratique des cas industriels nous ont montré que cette vision classique du SI est *idéaliste* : les entreprises n'ont pas toutes le même niveau de **maturité** ni la capacité à s'investir dans un projet d'architecture du SI. Ainsi, certaines entreprises n'abordent que l'essentiel du SI avec les vues applicative et fonctionnelle, tandis que d'autres poussent l'exercice jusqu'aux couches de processus métier et technique. De plus un SI est rarement cartographié en entier, le travail d'architecture commence lorsque surviennent des problèmes : techniques, de maintenance ou de coût. Lorsque les problèmes deviennent majeurs et ne peuvent plus être réparés par l'application de pansement (*Patch*), un audit complet est nécessaire pour redéfinir une vision stratégique du SI par la cartographie de l'existant. Ainsi, l'architecture va s'attaquer à *une ou plusieurs applications* mises en cause dans l'entreprise et non à l'intégralité du SI.

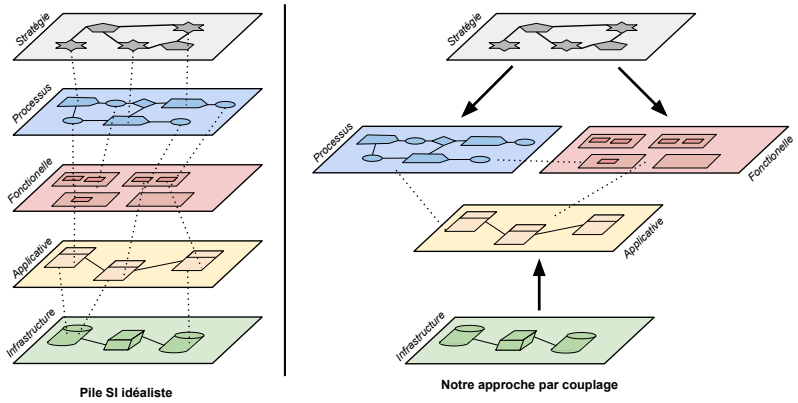


Figure 2. Les couches du SI : approche idéale vs. approche pragmatique

En pratique le but est surtout de rapprocher les modèles et de mettre en évidence les forces et faiblesses pour des scénarios d'évolution. Dans nos travaux précédents (Pepin *et al.*, 2016) nous avons présenté une vision opérationnelle, illustrée à droite sur la FIGURE 2. Le cœur de l'alignement est formé d'un triplet de points de vue : métier, fonctionnel et applicatif. La traçabilité est mise en œuvre par des liens sémantiques variés entre ces points de vue (ou couche). Le point de vue métier décrit les processus métier du SI. Le point de vue application représente les applications du SI sous forme de composants et services logiciels. Le point de vue fonctionnel représente l'urbanisation du système d'information *e.g.* en zone/îlots/quartiers. Selon le niveau de maturité, la préoccupation, ou l'importance de l'entreprise, on disposera

ou pas des points de vue métier et fonctionnel. L'approche est *flexible* (agile) car elle s'adapte aux éléments disponibles. Il sera toujours possible d'ajouter des points de vue à l'alignement. Les liens sémantiques expriment des relations de raffinement ou de correspondance pour les données et les traitements. La FIGURE 3 est un extrait des méta-modèles de ces points de vue et de leur alignement détaillés dans (Pepin, 2016).

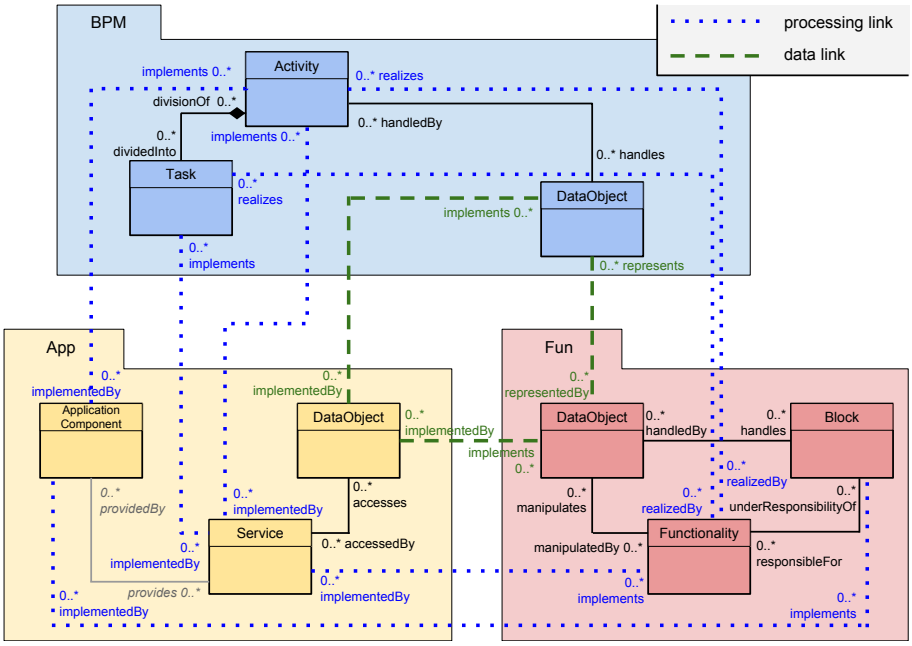


Figure 3. Définition de l'alignement au niveau méta-modèle

Le *rapprochement des points de vue* se fait en concrétisant la stratégie par des modèles de processus métier (*top-down*) et en masquant les détails d'implantation (*bottom-up*) jusqu'à un niveau acceptable pour un "langage commun". Le langage commun est alors défini comme un tissage entre les langages *i.e.* une correspondance entre leurs concepts. A ce stade, nous faisons abstraction de la vision stratégique , nous y reviendrons en section 5.2. A l'autre bout de la chaîne, le code déployé implique une maîtrise de l'infrastructure pas toujours disponible, nous y reviendrons en section 5.1.

L'alignement s'inscrit dans la première étape du travail de l'architecte dans le cycle ADM (FIGURE 1) : établir la cartographie du SI (B et C). Le *processus d'alignement* vise à alimenter les modèles *BPM*, *App* et *Fun* en fonction des données disponibles dans l'organisation puis à les aligner concrètement. Pour obtenir le modèle applicatif, nous avons proposé un processus de remontée en *abstraction* depuis le code source des applications (rétro-ingénierie). Une série de transformations permet d'abstraire les concepts techniques pour ne conserver que les éléments architecturaux. Pour obtenir les modèles métiers, c'est l'inverse, on doit matérialiser la stratégie d'entreprise, on parle de *concrétisation* ou de raffinement. Si la documentation métier est inexistante ou partielle, les différents acteurs (analystes métier, analystes fonctionnels et utilisa-

teurs finaux) doivent définir les modèles fonctionnels et processus à partir de leurs savoirs et connaissances du fonctionnement et de l'organisation de l'entreprise. Les processus ont été implantés sous Eclipse avec Modisco et Mia-transformation. Les détails sont fournis dans le chapitre 5 de (Pepin, 2016). Pour aligner des modèles obtenus, nous proposons une solution non-intrusive de *tissage* d'implémentation de facettes permettant l'extension virtuelle de méta-modèles. Nos propositions forment une méthode *outillée* présentée dans (Pepin *et al*., 2018) et qui utilise des standards de l'ingénierie des modèles. Nous avons notamment contribué au framework EMF Facet.

En résumé, nous proposons une méthode qui consiste à outiller la démarche d'architecture d'entreprise dans les étapes de cartographie, d'alignement et de prise de décision à la transformation du SI. Notre méthode est **générique** et **compatible** avec les principaux cadres d'architecture abordés précédemment. Nous avons vu qu'il n'est pas nécessaire d'adopter un cadre unique, mais qu'il est possible, voire recommandé de prendre les **bonnes pratiques** adaptées à la situation dans différents cadres. Nos travaux n'ont aucune adhérence forte avec l'un des cadres d'architecture d'entreprise. Les modèles sont construits et alignés pour être analysés et répondre aux interrogations des décideurs. Nous y répondons dans la section 3.

3. Maîtriser les évolutions du SI par analyse de dépendances

La cartographie et l'alignement des points de vue sont les étapes préliminaires d'un processus d'évolution du SI tel que celui de la FIGURE 1. L'architecte d'entreprise doit être capable d'identifier les éléments du SI à faire évoluer et de mesurer les impacts sur les différentes couches (ou points de vue). Les points de vue étant transverses (partie droite de la FIGURE 2), l'alignement que nous avons proposé à partir de liens sémantiques permet à l'architecte de *d'interroger le modèle i.e.* quels composants applicatifs implémentent telle fonctionnalité ? Quels processus métiers sont impactées par des modifications de composants applicatifs ou techniques ? L'objectif est de fournir aux urbanistes des outils d'assistance à l'évolution.

L'évaluation de l'alignement vise à détecter des problèmes de cohérences ou de complétude entre couches. Nous avons dans un premier temps travaillé sur une mesure de la qualité de l'alignement avec un indicateur global permettant d'étalonner la cohérence globale du système d'information. Nous n'avons pas trouvé de calcul d'un agrégat pertinent. L'analyse détaillée nous a convaincu que ce n'était ni la voie, ni le besoin. Ce n'est pas la voie car trouver un référentiel de bonne qualité de l'alignement se révèle ardu et finalement trop complexe. Ce n'est pas le besoin car l'objectif des architectes n'est pas d'aboutir après plusieurs itérations à un "super" alignement final, mais plutôt d'établir un constat à un moment donné avec des pistes d'améliorations puisque le système d'information évolue en permanence. L'évaluation peut se faire par observation/modification, par une analyse structurelle ou par des requêtes spécifiques. Dans la suite nous présentons trois techniques : l'outil de tissage, l'analyse par requêtes et l'analyse de dépendances.

3.1. Observer et modifier l'alignement du SI par tissage

L'alignement de modèles réalisé à l'aide de l'approche de la section 2 permet une navigation bi-directionnelle, *montante et descendante* entre les points de vue du SI. Un éditeur arborescent compatible avec le framework Eclipse EMF permet de charger les modèles et les liens sémantiques d'alignement permettent la navigation. Nous proposons un éditeur de tissage pour pour créer, consulter, modifier ou supprimer les liens d'alignement entre les trois points de vue exhibés dans la section 2. Cet éditeur est donc un navigateur dans l'alignement. La FIGURE 4) illustre cet éditeur sur un cas de simulation bancaire. A droite, le point de vue applicatif comprend différents services et fonctions et le point de vue fonctionnel met en évidence le bloc Adhérent d'un contrat IARD. A gauche le tissage relie le bloc fonctionnel *Simulation contrat* avec les services du modèle applicatif et notamment le service *validerSimulationContrat*.

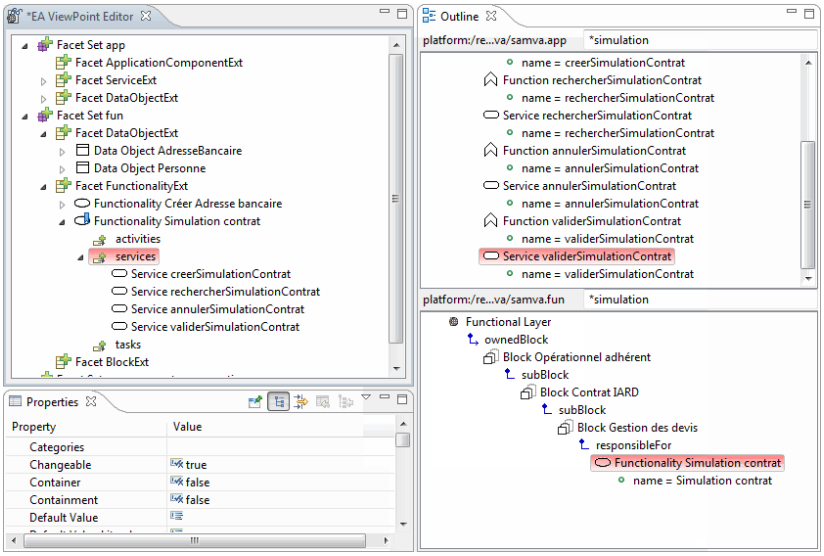


Figure 4. Editeur de tissage pour la navigation entre modèles

Naviguer et modifier le tissage ne suffit pas pour évaluer. D'une part, la navigation arborescente même assistée par des filtres de recherche est peu pratique pour les modèles volumineux. D'autre part, elle ne répond qu'aux seuls types de questions posées précédemment. Nous souhaitons en poser d'autres comme "quels sont les éléments qui ne sont pas alignés correctement?"... Afin de pallier ce problème, nous proposons deux types d'analyses dans les sections suivantes.

3.2. Mesurer l'alignement du SI par requêtes

Le travail d'alignement est proportionnel au volume d'informations cartographiant les points de vue du SI. L'architecte doit pouvoir suivre son avancement et vérifier sa

cohérence, mais aussi mesurer l'alignement pour fournir aux décideurs des indicateurs guidant à la fois l'analyse de l'état actuel, la détection d'anomalies et la valorisation de scénario de projections futures.

La mesure de l'alignement reste inexplorée (Aversano *et al.*, 2016). Les travaux de Simonin (Simonin, 2009) se focalisent sur la couche fonctionnelle (urbanisation). D'autres traitent surtout de la couche métier (voire stratégique) avec le logiciel. Ils incluent la notion d'acteurs, qui n'a pas forcément de sens du point de vue logiciel. Aversano et al. (Aversano *et al.*, 2010) traitent de la couverture des processus métiers par le logiciel, mais ne présentent pas les modèles du logiciel. Ils ont proposé des métriques pour l'*alignement fonctionnel* dans (Aversano *et al.*, 2016) mais pour des modèles de bas niveaux (activités et classes UML). Nous avions défini des métriques d'indice de couplage et de cohérence similaires mais leur intérêt restait difficile à appréhender car l'alignement est multiforme. Rolland et Etien (Etien, Rolland, 2005) alignent les processus métiers avec des modèles UML via des ontologies mais sans parc applicatif. Thévenet et al. (Thevenet, Salinesi, 2007) s'intéressent à l'alignement stratégique et l'évolution, sans lien direct avec le code.

Nos indicateurs sont classés en quatre catégories : couverture, cohérence, densité d'alignement concret (entre les modèles métiers, applicatifs et fonctionnels) et couverture du code final. Le *framework* Eclipse EMF permet d'interroger les modèles par requêtes écrites dans le langage OCL. Nous avons étendu OCL pour que les requêtes soient compatibles avec EMF Facet, utilisé pour le tissage entre modèles. Les requêtes servent à mesurer les indicateurs d'alignement. Ainsi, une première mesure possible est la complétude de l'alignement illustré par la FIGURE 5 : le nombre d'éléments alignés sur le nom d'éléments alignables des points de vue processus et applicatif, selon la définition de l'alignement (FIGURE 3) entre points de vue au niveau méta-modèle.

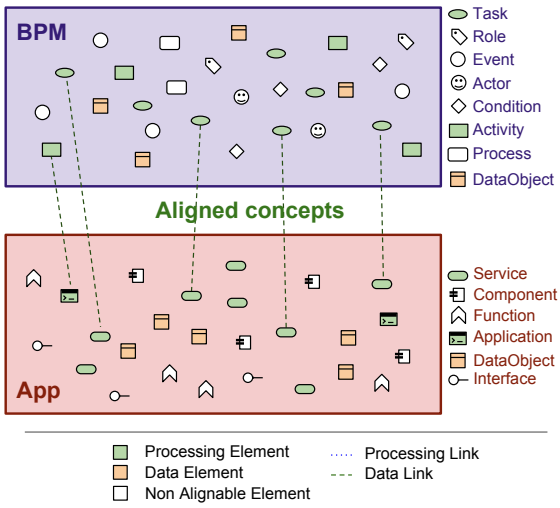


Figure 5. Complétude Activités de Processus/Service applicatif

Les concepts sont typés ce qui autorise des analyses par préoccupation (*concerns*) *e.g.* données et traitements sont distingués. Les requêtes permettent la vérification de cohérence entre traitements et données. Si un élément de traitement d'un point de vue A qui manipule des éléments de données, est aligné avec un élément de traitement d'un autre point de vue B, alors les éléments de données du point de vue A doivent être alignés aux éléments de données du point de vue B et réciproquement. La requête 1 liste les blocs ayant un alignement incohérent de données avec des composants applicatifs.

Listing 1 – Oublis entre Bloc Fonctionnels et Composants applicatifs

```
BlockAppComponentNonConsistent:
context FunctionnalLayer
fun::Block.allInstances()->select(applicationComponents->notEmpty())->
    symmetricDifference(
fun::Block.allInstances()->select(blk | blk.applicationComponents.
    provides.accesses.dataObjectsFun.handles->includes(blk)))
```

L'analyse par requête de l'alignement détecte des oublis et incohérences et de les corriger à chaque itération permettant un réel suivi qualité en continu du SI. Des règles d'urbanisme spécifiques au SI peuvent aisément être implémentées en OCL et compatibles avec les liens d'alignement.

3.3. Identifier les dépendances entre points de vue

Un SI doit avoir des qualités de cohérence forte et couplage faible, bien connues en conception logicielle (Parnas, 1972). Le SI au fil des évolutions peut devenir intriqué. L'analyse de dépendances répond ainsi à des problématiques très diverses. Du point de vue applicatif, elle permet à l'architecte de mettre en évidence (i) les zones de couplage fort candidates à une restructuration (*refactoring*) (ii) les impacts en cas de débranchement ou de remplacement de composants applicatifs. Du point de vue processus métiers, elle permet d'identifier les dépendances entre activités mais aussi déterminer les composants applicatifs impactés par une réorganisation métier. Outre le calcul des dépendances, l'architecte a besoin de représentations visuelles des dépendances (à grande échelle) pour se projeter dans l'évolution.

Techniquement, les modèles alignés représentent un graphe composé de nœuds (les instances des types) et d'arcs (les relations entre instances). Ce qui permet d'appliquer des algorithmes issus de la théorie des graphes. Les graphes peuvent se représenter par une matrice, et plus précisément une matrice d'adjacence. Les nœuds sont représentés en-tête de chaque ligne (l) et colonne (c), une intersection entre une ligne et une colonne est la présence d'une dépendance entre deux nœuds. Cette matrice est appelé *Matrice de structure de dépendance* (DSM) (Eppinger, Browning, 2012). Il existe donc $l \times c$ possibilités de représenter le graphe en matrice, selon l'ordre de lecture de chaque nœuds. De ce constat quantitatif, nous nous sommes posés la question du meilleur ordre pour représenter notre graphe d'alignement et ainsi de faciliter la lecture de l'architecte pour visualiser les dépendances et notamment identifier les groupes de dépendance. Nous avons étudié les algorithmes de regroupements (*clus-*

tering) (Schaeffer, 2007) pour choisir le plus adapté à la nature de nos travaux. Nous avons retenu l'algorithme Markov Cluster (MCL) (Dongen, 2000) qui présente l'avantage de ne pas nécessiter de connaître à l'avance le nombre de clusters mais l'inconvénient de s'appliquer naturellement à un seul point de vue à la fois.

La matrice correspondant à un alignement des points de vues (processus, fonctionnel, applicatif) est multi-domaine, Multiple-Domain Matrix (MDM) (Bartolomei *et al.*, 2010). L'application d'un algorithme de regroupement sur une matrice MDM est plus délicate. Notre contribution est de proposer d'appliquer l'algorithme de regroupement de trois manières différentes sur l'alignement. Un regroupement global qui s'applique sur toute la matrice mélangeant les domaines. Un regroupement intra-domaine qui ne s'applique qu'à l'intérieur de chaque domaine. Un regroupement inter-domaine qui s'applique à l'intersection entre deux domaines. Cette innovation permet à l'architecte de visualiser précisément par quels liens les points de vue sont en dépendance.

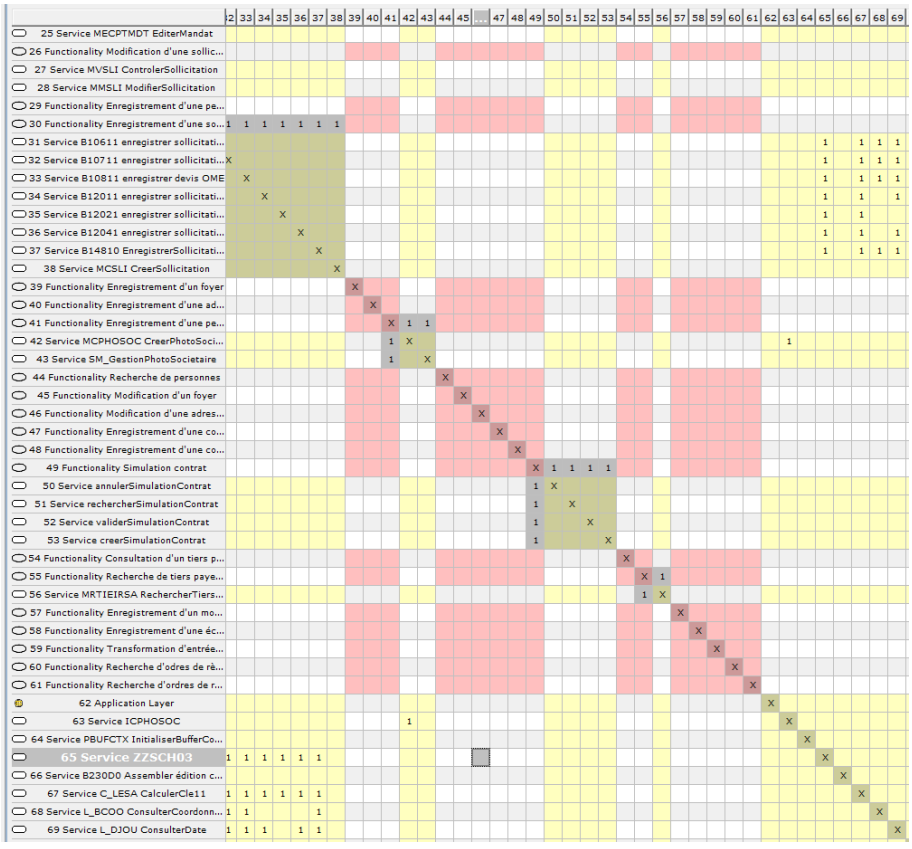


Figure 6. Matrice de dépendances entre modèles

La FIGURE 6 illustre notre outil de calcul de regroupement MCL inter-domaine sur un alignement entre un modèle d'un point de vue fonctionnel et un modèle d'un point

de vue applicatif du même exemple (jouet) que la FIGURE 4. Les lignes et colonnes représentent les concepts du modèle applicatif (en jaune) ou fonctionnel (en rose). La ligne diagonale montre que chaque concept est lié à lui-même. Les points intéressants sont les sous-matrices plus pleines comme celui de la ligne 20 et suivantes qui met en évidence un cluster de services, concernant ici les sollicitations clients. En ligne 41-43 on voit un couplage applicatif-fonctionnel sur la gestion des photos. Noter que nos algorithmes de regroupement sont déterministes et fournissent donc un résultat identique quelle que soit l'exécution d'un jeu de valeurs. Noter aussi qu'il s'agit d'un prototype que nous avons développer et que de nombreux points sont à améliorer.

4. Expérimentations

Dans un premier temps, nous avons testé la méthode (principes, démarche, outils) sur des cas simples et imaginaires, comme celui qui a servi à illustrer nos propos dans les sections précédentes. Nous avons eu ensuite l'opportunité de tester notre démarche sur trois cas d'études concrets, provenant de sociétés d'Assurance Mutuelle françaises que nous nommerons SAMM, SAMI et SAMUT. Chaque étude possède ses spécificités et couvre tout ou partie des modèles du processus de la section2. Il ne s'agissait pas de réaliser un cas de bout en bout et le tissage reste partiel. L'expérimentation a pour but de vérifier la faisabilité de notre démarche : pertinence des modèles vis-à-vis de la pratique en entreprise, adéquation de la démarche, automatisation des transformations et de leur enchaînement, pertinence du tissage et de sa mise en œuvre, application à des modèles volumineux. Nous résumons ci-après les expériences.

Le cas SAMM est composé *i)* d'un code source complet écrit en Java avec 33 400 classes (3 400 000 lignes de codes) et *ii)* d'un référentiel d'entreprise sous la forme d'un portail HTML exporté depuis le logiciel MEGA EA [1]. Le but de ce scénario était de valider que notre méthode permet d'exécuter l'analyse par clustering d'un alignement réel à l'aide de notre matrice de dépendance. Le référentiel d'entreprise contient 360 diagrammes de processus métier couvrant la totalité du SI. Le volume de l'application est conséquent et représente un véritable défi à traiter. Ce cas d'étude nous a permis de tester notre approche sur un source code important, le chargement des modèles obtenus par rétro-ingénierie a été un défi pour nos outils. La transformation a été facilitée par la présence d'une nomenclature des concepts qui se retrouve à différents niveaux et par là même montre de bonnes pratiques de codage, à quelques exceptions près. Néanmoins, nous regrettons ne pas avoir eu accès au source original du référentiel MEGA pour réaliser un tissage complet.

Le cas SAMI est composé uniquement d'un référentiel MEGA dont le source est disponible. Le but de ce scénario était de valider que notre méthode permettait bien d'exécuter une remontée en abstraction d'un code applicatif réel et conséquent. Nous avons extrait les différents concepts pour peupler à l'aide d'une transformation nos différents modèles (App, BPM, Fun) afin de vérifier la couverture et la compatibilité

1. http://www.mega.com/fr/solution/business-architecture

des concepts. Le cas présente 625 composants, 11894 services, 18 blocs fonctionnels, 131 fonctionnalités, 167 processus et 268 activités.

Le cas SAMUT n'avait aucune représentation métier. Le but de ce scénario était de valider que notre méthode permet de construire un alignement à partir de zéro en constituant les modèles et d'aider l'architecte d'entreprise à visualiser les dépendances. Les objets de données ont été extraits par rétro-ingénierie d'une base de données hiérarchique et les composants applicatifs de procédures stockées. Un architecte à ensuite modélisé et identifié des blocs fonctionnels et nous avons alors pu réaliser le tissage entre les blocs et les composants applicatifs. Le SI comporte 12 blocs, 1045 composants et 669 objets de données. Ce cas d'étude a permis de tester notre méthode de tissage, et isoler des composants qui étaient orphelins (rangés dans aucun bloc).

Ces trois cas ont des propriétés particulières, les supports sources sont hétérogènes et représentatifs de la disparité de maturité des SI. Ils ont mis en évidence la souplesse de notre méthode qui peut s'adapter à chaque étape. La contrepartie est d'enrichir l'écosystème des modèles et transformations pour chaque nouveau cas rencontré.

5. Extensions de l'alignement opérationnel

Dans les sections précédentes, nous nous sommes attachés à réduire le fossé entre domaine métier et informatique. Nous discutons ici de l'extension de l'alignement aux couches infrastructure et stratégie en vue de couvrir l'ensemble des vues du SI.

5.1. Alignement d'infrastructure

Dans nos cas d'études, nous avons constaté que les entreprises maintenaient une cartographie technologique composée des informations sur les serveurs, systèmes, réseaux et caractéristiques physiques. Cette cartographie permet à la direction des systèmes d'information (DSI) de réaliser une intervention de maintenance ciblée. Par exemple, en cas de panne d'un service applicatif, il est nécessaire de retrouver rapidement les coordonnées du logiciel incriminé : IP, droits d'accès, site physique ou cloud, etc. Conserver une cartographie technologique actualisée est une forte préoccupation. Une panne entraînant la cessation d'une partie de l'activité de l'entreprise ayant un coût lié au temps d'immobilisation, la résolution doit-être la plus rapide possible pour limiter les pertes. Nous avons établi un premier état de l'art qui met en évidence que certaines cartographies technologiques mélangent tout les aspects, tandis que d'autres distinguent les concepts de déploiement et des concepts d'infrastructure.

Déploiement ce point de vue a pour but de décrire comment les applications sont réparties au niveau virtuel. La vue applicative décrit les composants, fonctions ou objets de donnée, mais pas leur nature technologique. Par exemple, un ou plusieurs composants écrits en Java peuvent être encapsulés dans des archives *jar*, *war* ou *ear*et déployés dans un serveur d'application J2EE (Glassfish, JBoss, Apache Tomcat...) ; un ou plusieurs objets de données sont stockés dans une table de base de données (MySQL, Postgre, Oracle, Neo4j, MongoDB...).

Infrastructure ce point de vue a pour but de décrire le matériel où sont installés les logiciels, ainsi que le réseau et les interfaces d'échanges d'informations. Ce point de vue modélise la situation géographique du matériel (pays > ville > site > bâtiment > salle > baie > unité) et peut couvrir plusieurs sites distants.

Archimate® [2] distingue le niveau technologique (intergiciel, composants et services des applications) et le niveau physique (équipements matériels, de réseaux physiques...) de la FIGURE 7. Le niveau technologique peut être qualifié de virtuel.

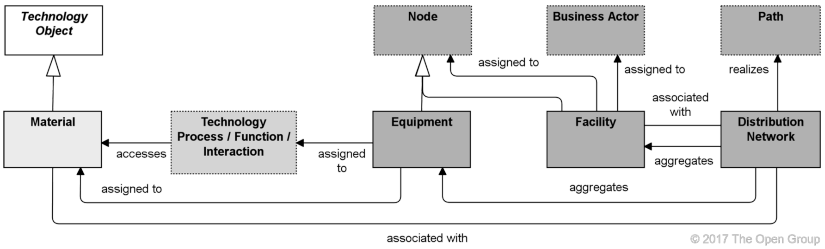


Figure 7. Le méta-modèle physique de ArchiMate 3.0.1

La Direction Interministérielle des Systèmes d'Information et de Communication (DISIC) [3] de l'État français propose un seul point de vue infrastructure (FIGURE 8) qui couvre le niveau physique.

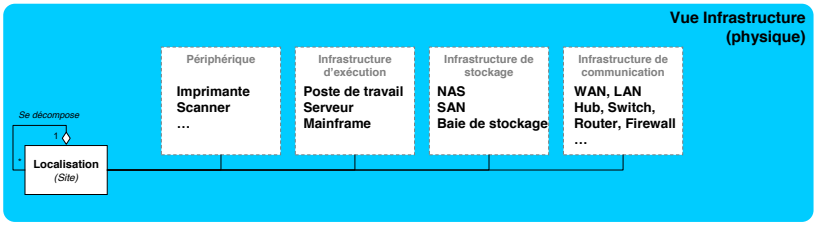


Figure 8. Les concepts simplifiés de la Vue Infrastructure selon DISIC

D'autres études de méta-modèles d'infrastructure doivent étoffer notre état de l'art et nous permettre de créer une extension générique de notre définition de l'alignement au niveau méta-modèle qui doit rester indépendante d'une solution spécifique.

5.2. Alignement stratégique

L'alignement stratégique est une thématique qui a généré une grande production scientifique depuis l'article fondateur d'Henderson et Venkatraman (Henderson, Venkatraman, 1993). Cela s'explique non seulement par l'intérêt croissant du sujet mais aussi plus prosaïquement par le fait que le problème relève à la fois des domaines de

2. http://pubs.opengroup.org/architecture/archimate3-doc/
3. https://references.modernisation.gouv.fr

la gestion et de l'informatique. Nous restreignons notre domaine d'investigation aux approches incluant des modèles, notamment autour de l'architecture d'entreprise. Le problème se résume alors à disposer de modèles pour la couche stratégie et la couche processus métiers et de modèles pour l'alignement entre ces couches conformément à l'approche de la FIGURE 2. Nous disposons déjà de langages pour la couche métier.

L'étude bibliographique met en évidence les éléments suivants :

– Il existe de nombreux langages et techniques de modélisation de la stratégie ou de maturité (Ullah, Lai, 2011 ; Aversano *et al*., 2016 ; Roelens *et al*., 2017), ce qui ne facilite pas la mise en pratique pour les entreprises et la conception d'outils d'aide.

– Certaines approches s'appliquent à la conception (*design time*) (Thevenet, Salinesi, 2007) en établissant des ponts entre couches (traçabilité), l'ingénierie des exigences y joue un rôle prépondérant (Thevenet, Salinesi, 2007 ; Ullah, Lai, 2011). D'autres visent la rétro-conception en annotant les modèles métiers d'indicateurs exploités pour relier à des concepts de la couche stratégie.

– La couche stratégique peut elle-même être décomposée en plusieurs couches notamment en séparant par exemple une approche basée sur les *buts* (*goal modeling languages* (Ullah, Lai, 2011 ; Doumi *et al*., 2011), i* (Pijpers *et al*., 2008)) et celle basée sur la *valeur* (VMDL (Roelens, Poels, 2013), e^3-value (Pijpers *et al*., 2008)). Une approche telle que *Process Goal Alignment* (PGA) permet de relier les deux niveaux (Roelens *et al*., 2017).

– La jonction entre les couches métier et stratégique peut se faire en annotant les processus métiers d'informations utiles à l'alignement avec la stratégie. Morrisson et al. proposent ainsi un calcul d'alignement sur les buts basés sur ces annotations (Morrison *et al*., 2012). Ullah et Lai proposent de relier les processus métiers aux buts (Ullah, Lai, 2011).

– Plus récemment, le *Business Motivation Model* (BMM) de l'OMG contient des éléments pour spécifier la stratégie et d'autres éléments pour relier le niveau stratégie au niveau métier (Bhattacharya, 2017 ; Hinkelmann, Pasquini, 2014).

Ne disposant pas de modèles de stratégie pour nos études de cas, nous proposons dans un premier temps un système de pondération des activités des processus métiers pour évaluer l'impact des changements sur le système futur. Ces pondérations servent à la mesure d'impact des différents scénarios d'évolution du SI. Par la suite, nous envisageons une mise en œuvre d'une version générique de BMM qui a l'avantage de s'intégrer avec nos standards de modélisation que ce soit TOGAF, UML ou EMF.

6. Conclusion

Nous avons proposé une méthode pragmatique au problème d'alignement des points de vue métier et applicatif s'insérant dans la démarche d'urbanisation des architectures d'entreprise. Notre méthode est basée sur une proposition de modèles intermédiaires génériques, un rapprochement des points de vue et un alignement par tissage de concepts comparables. Le rapprochement est rendu possible par une abs-

traction progressive du code en architecture applicative à base de composants et services. Notre approche est outillée dans le cadre d'Eclipse EMF et a été expérimentée sur des cas réels d'entreprise, permettant d'éprouver la viabilité de notre approche. L'application à un source code de taille importante fut un défi mais les modèles obtenus par rétro-ingénierie ont pu être chargés par nos outils.

Certains points sont encore à améliorer dans l'automatisation du processus comme la découverte des composants logiciels ou la détection des correspondances du tissage. Le travail en cours sur les métriques doit permettre la mise en place de tableaux de bord pour l'évaluation du tissage et les scénarios d'évolution. Une autre perspective est l'enrichissement du tissage avec plus de concepts pour mieux prendre en compte les points d'évolution du système d'information, pas seulement la structure (Saat *et al.*, 2010). En particulier, nous souhaitons pouvoir représenter le couplage entre parties de modèles. Dans cette lignée, nous devons mettre à l'épreuve nos méta-modèles vis-à-vis de pratiques (non automatisées) d'urbanisation et d'alignement.

Bibliographie

Aversano L., Grasso C., Tortorella M. (2010). Measuring the alignment between business processes and software systems: A case study. In *Proceedings of the 2010 acm symposium on applied computing*, p. 2330–2336. New York, NY, USA, ACM.

Aversano L., Grasso C., Tortorella M. (2016). Managing the alignment between business processes and software systems. *Information and Software Technology*, vol. 72, p. 171 - 188.

Bartolomei J., Cokus M., Dahlgren J., Neufville R. de, Maldonado D., Wilds J. (2010). Analysis and applications of design structure matrix, domain mapping matrix, and engineering system matrix frameworks. Consulté sur http://ardent.mit.edu/real_options/Real_opts_papers/Jennifer%20mini%20thesis.pdf

Bhattacharya P. (2017). Modelling Strategic Alignment of Business and IT through Enterprise Architecture: Augmenting Archimate with BMM. *Procedia Computer Science*, vol. 121, p. 80 - 88. (CENTERIS 2017)

Coltman T., Tallon P., Sharma R., Queiroz M. (2015, 01 Jun). Strategic it alignment: twenty-five years on. *Journal of Information Technology*, vol. 30, n° 2, p. 91–100.

Dongen S. van. (2000). *Graph Clustering by Flow Simulation*. Phd thesis, University of Utrecht.

Doumi K., Baïna S., Baïna K. (2011). Modeling approach for business IT alignment. In *Proceedings of ICEIS 2011, volume 4, beijing, china*, p. 457–464.

Eppinger S. D., Browning T. R. (2012). *Design Structure Matrix Methods and Applications*. Cambridge, Mass, MIT Press.

Etien A., Rolland C. (2005). Measuring the fitness relationship. *Requirements Engineering*, vol. 10, n° 3, p. 184-197. Consulté sur http://dx.doi.org/10.1007/s00766-005-0003-8

Henderson J. C., Venkatraman N. (1993, janvier). Strategic alignment: Leveraging information technology for transforming organizations. *IBM Syst. J.*, vol. 32, n° 1, p. 4–16.

Hinkelmann K., Pasquini A. (2014). Supporting business and IT alignment by modeling business and IT strategy and its relations to enterprise architecture. In *Enterprise systems conference, ES 2014, shangai, china, august 2-3, 2014*, p. 149–154. IEEE.

Lankhorst M. M. (2013). *Enterprise architecture at work - modelling, communication and analysis (3. ed.)*. Springer.

Morrison E. D., Ghose A. K., Dam H. K., Hinge K. G., Hoesch-Klohe K. (2012). Strategic alignment of business processes. In G. Pallis *et al.* (Eds.), *Service-oriented computing - icsoc 2011 workshops*, p. 9–21. Berlin, Heidelberg, Springer Berlin Heidelberg.

Parnas D. L. (1972, décembre). On the criteria to be used in decomposing systems into modules. *Commun. ACM*, vol. 15, n° 12, p. 1053–1058. Consulté sur http://doi.acm.org/10.1145/361598.361623

Pepin J. (2016). *Architecture d'entreprise : alignement des cartographies métiers et applicatives du système d'information*. Thèse, Université Bretagne Loire.

Pepin J., André P., Attiogbé C., Breton E. (2016). Using ontologies for enterprise architecture integration and analysis. *CSIMQ*, vol. 9.

Pepin J., André P., Attiogbé C., Breton E. (2018). Virtual extension of meta-models with facet tools. In *Proceedings of the 6th international conference on model-driven engineering and software development - volume 1: Modelsward,*, p. 59-70. SciTePress.

Pepin J., André P., Attiogbé C., Breton E. (2015). A method for business-it alignment of legacy systems. In *Proceedings of ICEIS 2015, volume 3, barcelona, spain, 27-30 april, 2015.*

Pijpers V., Gordijn J., Akkermans H. (2008). Business strategy-it alignment in a multi-actor setting: A mobile e-service case. In D. Fensel, H. Werthner (Eds.), *Proceedings of the 10th international conference on electronic commerce 2008*, vol. 342. ACM.

Roelens B., Poels G. (2013). Towards an integrative component framework for business models: Identifying the common elements between the current business model views. In *Proceedings of the caise'13 forum*, vol. 998, p. 114–121. Valencia, Spain, CEUR-WS.org.

Roelens B., Steenacker W., Poels G. (2017, 13 Jan). Realizing strategic fit within the business architecture: the design of a process-goal alignment modeling and analysis technique. *Software & Systems Modeling*.

Saat J., Franke U., Lagerstrom R., Ekstedt M. (2010). Enterprise architecture meta models for IT/Business alignment situations. In *Enterprise distributed object computing conference (EDOC), 2010 14th IEEE international*, p. 14–23.

Schaeffer S. E. (2007). Graph clustering. *Computer science review*, vol. 1, n° 1, p. 27–64.

Simonin J. (2009). *Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel*. Thèse de doctorat non publiée, Université de Rennes 1.

Thevenet L., Salinesi C. (2007). Aligning IS to organization's strategy: The instal method. In *Advanced information systems engineering, 19th international conference, caise 2007, trondheim, norway, june 11-15, 2007, proceedings*, p. 203–217.

Ullah A., Lai R. (2011). Modeling business goal for business/it alignment using requirements engineering. *Journal of Computer Information Systems*, vol. 51, n° 3, p. 21-28.

Ullah A., Lai R. (2013, avril). A systematic review of business and information technology alignment. *ACM Trans. Manage. Inf. Syst.*, vol. 4, n° 1, p. 4:1–4:30.

Session commune INFORSID-RCIS

Emergence d'un nouveau type de Système de Systèmes : observations et propositions à partir du système d'alerte national français

Maude Arru[1], Elsa Negre[1], Camille Rosenthal-Sabroux[1]

1. Université Paris-Dauphine
PSL Research Universities,
CNRS UMR 7243, LAMSADE
75016 Paris, France
{maude.arru,elsa.negre,camille.rosenthal-sabroux}@dauphine.fr

RÉSUMÉ. Les alertes permettent de prévenir ou de limiter des dommages humains et matériels si elles sont délivrées à temps et si elles permettent aux intervenants et à la population concernée de se préparer de manière adéquate à la crise à venir. Aujourd'hui, il existe de nombreux indicateurs et systèmes de capteurs conçus pour produire des alertes et limiter les conséquences des crises. Ces systèmes ont prouvé leur efficacité mais ils demeurent complexes, incluant différentes organisations expertes, difficiles à gérer. Nous étudions dans ce document le cas du système national d'alerte précoce en France (appelé SAIP), qui peut être représenté comme un Système de Systèmes. Beaucoup de Systèmes de Systèmes existent, ils peuvent être dirigés, collaboratifs, virtuels ou « reconnus » (acknowledged). Nous étudions ici à quel type de système correspond le système d'alerte précoce français et quelles ouvertures de recherche peuvent raisonnablement être envisagées pour ce système.

ABSTRACT. Warnings can help to prevent damage and harm if they are issued timely and provide information that helps respondents and population to adequately prepare for the disaster to come. Today, many indicators and sensor systems are designed to produce alert and reduce disaster risks. These systems have proved to be effective but they remain complex, include different expertise components, and are difficult to manage. We study in this paper the case of the National Early-Warning System in France (called SAIP), which can be seen as a System of Systems (SoS). A lot of SoSs exist. They can be directed, collaborative, virtual or even acknowledged systems. We study here what type of system corresponds to the French Early-Warning System, which openings may reasonably be considered for this system and we introduce a new category of SoSs: "delimited system".

Mots-clés : Systèmes d'alerte précoce, Systèmes de Systèmes, Gestion de crise, Systèmes d'Information

KEYWORDS: Early-Warning Systems, Systems of Systems, Crisis management, Information Systems

DOI:10.3166/RCMA.25.1-n © 2016 Lavoisier

Les alertes permettent de prévenir ou de limiter des dommages humains et matériels si elles sont délivrées à temps et si elles permettent aux intervenants et à la population concernée de se préparer de manière adéquate à la crise à venir. Aujourd'hui, il existe de nombreux indicateurs et systèmes de capteurs conçus pour produire des alertes et limiter les conséquences des crises, appelés systèmes d'alerte précoce (EWS). Ils peuvent être définis comme des systèmes permettant d'alerter et d'informer des populations sensibilisées et des organisations préparées, afin de prendre les mesures nécessaires pour anticiper, éviter ou réduire les conséquences matérielles et humaines d'une potentielle crise à venir (Waidyanatha, 2009). Ils sont composés de quatre éléments (Basher, 2006) : connaissance du risque, surveillance des indicateurs et services d'alerte, diffusion des alertes et capacité de réponse. Ces EWS ont prouvé leur efficacité mais ils demeurent complexes, incluant différentes organisations expertes, et difficiles à gérer. En France, le système national d'alerte (appelé SAIP) est ainsi conçu et s'inscrit sous l'autorité de la Direction Générale de la Sécurité Civile et de la Gestion des Crises. Il repose sur différentes organisations de sécurité civile (pompiers, gendarmes…) et d'expertise (Météo-France, Vigicrue…). Toutes ces organisations sont indépendantes les unes des autres et selon l'évènement, elles peuvent avoir à collaborer ensemble dans un objectif commun.

Cette collaboration entre organisations est un des objectifs des systèmes de systèmes (SoSs), qui permettent d'aider à unifier les informations partagées entre les collaborateurs et leur permettre de développer un cadre commun. Nous les approchons sous l'angle de (ISO/IEC/IEEE, 2015) qui définit un SoS comme « un ensemble de systèmes réunis pour réaliser une tâche qu'aucun système ne peut accomplir seul. Chaque système constituant maintient sa propre gestion, ses objectifs et ses ressources tout en se coordonnant au sein du SoS et en s'adaptant pour atteindre les objectifs du SoS ». Beaucoup de SoSs existent, ils peuvent être de type dirigés, collaboratifs, virtuels ou « reconnus » (*acknowledged*).

Le SAIP correspond bien à la définition d'un SoS et en partage toutes les propriétés (Maier, 1998), mais il ne correspond à aucun des types de SoS précités. Nous proposons ici d'ouvrir la catégorisation des SoSs à un nouveau type, appelé « systèmes délimités ». De tels SoSs ont une « autorité de gestion » centrale. Les systèmes constituants et « l'autorité centrale » partagent les informations dont ils ont besoin pour prendre des décisions pertinentes. La nature et la quantité d'information peuvent évoluer dans le temps, selon les situations. Les systèmes constituants ont une autonomie parfaite dans leurs propres propriétés et objectifs. Nous considérons que beaucoup de systèmes récemment créés tels que *Moovit* ou le Programme *Waze Citoyen* entrent dans cette nouvelle catégorie. Enfin, nous considérons que dans les SoSs de manière générale, la centralisation des données est un enjeu majeur qui pourrait être optimisé avec des solutions de travail collaboratives numériques pour gérer les connaissances et des données acquises de manière collaborative à différents niveaux. Pour conclure, nous posons la question de l'adaptation de l'architecture du SAIP aux nouvelles technologies et usages, et à sa capacité en tant que SoS à intégrer de nouveaux systèmes constituants.

L'intégralité de cet article est publié dans les actes de la conférence RCIS 2018.

Bibliographie

Basher R. (2006). Global early warning systems for natural hazards: systematic and people-centred. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 364, no 1845, p. 2167–2182.

ISO/IEC/IEEE. (2015). *Systems and software engineering – system life cycle processes*. Rapport technique, ISO/IEC/IEEE 15288:2015 éd. Geneva, Switzerland: International Organisation for Standardisation / International Electrotechnical Commissions / Institute of Electrical and Electronics Engineers.

Maier M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, vol. 1, no 4, p. 267–284.

Waidyanatha N. (2009). Towards a typology of integrated functional early warning systems. *International journal of critical infrastructures*, vol. 6, no 1, p. 31–51.

Aide à la démarche experimentale en recherche en Système d'Information

Le processus de recherche THEDRE et son arbre de décision MATUI

Nadine Mandran[1], Sophie Dupuy-Chessa[1]

1. *Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG,*
38000 Grenoble, France
Prenom.Nom@univ-grenoble-alpes.fr

RESUME. Les Systèmes d'Information (SI) sont des systèmes socio-techniques qui doivent, par nature, prendre en compte les individus et leur environnement de travail. Ainsi la recherche en SI doit inclure l'Humain dans le processus de construction et d'évaluation de la connaissance scientifique. Mais prendre en compte l'Humain peut être difficile pour plusieurs raisons : 1) les chercheurs en informatique sont rarement formés aux méthodes de sciences humaines et sociales, 2) les spécialistes en sciences sociales qui connaissent les méthodes expérimentales, connaissent peu les modèles et les concepts des SI, 3) les SI s'appuient sur des concepts complexes et abstraits, 4) les humains au coeur des SI sont inconstants : leurs opinions et perceptions peuvent évoluer éventuellement même de manière contradictoire. Ces problèmes sont avant tout relatifs à la réalisation d'expérimentations avec des Humains dans un but de recherche. Dans ce cadre, notre question de recherche concerne le guidage des chercheurs dans la construction de leurs expérimentations c'est-à-dire dans l'aide à la sélection de méthodes expérimentales centrées utilisateur appropriée et à la définition des expérimentations.

Pour répondre à cette question, deux approches ont été proposées. La première est une approche méthodologique comme par exemple le Design Science. Elle permet de guider les chercheurs mais pas à un niveau suffisamment opérationnel. La deuxième approche repose l'analyse de méthodes expérimentales telles que les entretiens ou les questionnaires, pour définir leur contexte d'utilisation. Cependant l'utilisation de ces méthodes ne s'inscrit pas dans un processus global de recherche.

Le manque de liens entre les méthodologies de la recherche de haut niveau et les méthodes expérimentales très pratiques nous ont mené à formaliser un processus de recherche nommé THEDRE, qui inclut en particulier une étape de prise de décision dédiée à la sélection des méthodes expérimentales.

L'article intégral publié à RCIS'2018 présente globalement THEDRE et ses étapes liées à l'expérimentation. Il décrit un arbre de décision nommé MATUI qui aide les chercheurs dans leur travail expérimental. MATUI est basé sur une catégorisation des méthodes centrées utilisateurs et des critères de sélection de méthodes du point de vue des chercheurs. THEDRE et MATUI sont supportés par un outil de guidage en ligne. Ils sont le résultat de 10 ans de travail en support expérimental à des chercheurs en informatique. MATUI a en particulier été évalué grâce à deux groupes de discussion.

DOI:10.3166/RCMA.25.1-n © 2016 Lavoisier AR_DOI

Une approche centrée sur l'utilisateur pour intégrer les acteurs sociaux dans des communautés d'intérêt

Nadia Chouchani[1], Mourad Abed[2]

1. LAMIH, UMR, CNRS 8021
Valenciennes, France
Nadia.chouchani@univ-valenciennes.fr

2. LAMIH, UMR, CNRS 8021
Valenciennes, France
Mourad.abed@univ-valenciennes.fr

RESUME. La détection des communautés d'intérêt est un problème complexe qui a été abordé sous différents angles. Dans ce travail, nous proposons une approche centrée sur l'utilisateur intégrant des profils utilisateurs sociaux dans la détection des communautés dans les réseaux sociaux en ligne. Dans notre approche, nous calculons d'abord l'acquisition explicite des connaissances. En explorant les réseaux égocentriques des utilisateurs, nous pouvons déduire des similitudes implicites des intérêts. Ces similitudes sont estimées en référence à l'homophilie et à l'influence sociale. Cette dernière est utilisée pour améliorer l'analyse des sentiments au sein des communautés. Enfin, nous menons des expériences sur des ensembles de données extraits de réseaux sociaux réels.
Cet article est publié dans les actes de la 12ème conférence internationale Research Challenges in Information Science (29-31 Mai 2018), IEEE, Nantes, France.

Mots-clés : Réseaux Sociaux, Communauté d'intérêt, Profil utilisateur, Ontologie, Analyse des sentiments

Bases de données

Métriques structurelles pour l'analyse de bases orientées documents

Paola Gómez [1], **Claudia Roncancio** [1], **Rubby Casallas** [2]

1. *Univ. Grenoble Alpes, CNRS, Grenoble INP* [*], *LIG, 38000 Grenoble, France*
   {paola.gomez-barreto,claudia.roncancio}@univ-grenoble-alpes.fr
2. *TICSw, Universidad de los Andes,Bogotá - Colombia*
   rcasalla@uniandes.edu.co

RÉSUMÉ. *La flexibilité dans la structuration des données dans les bases orientées documents est appréciée pour permettre un développement initial rapide. Cependant, les possibilités de structuration des données sont nombreuses et le choix de structuration adopté reste assez crucial par son impact potentiel sur plusieurs aspects de la qualité des applications. En effet, chaque structuration peut présenter des avantages et des inconvénients notamment en matière d'empreinte mémoire, redondance de données engendrée, coût de navigation dans les structures et accès à certaines données, lisibilité des programmes. Dans cet article nous proposons un ensemble de métriques structurelles pour des "schémas" de documents JSON. Ces métriques permettent de refléter la complexité des schémas et des critères de qualité tels que leur lisibilité et maintenabilité. La définition de ces métriques s'appuie, entre autres, sur des expérimentations avec MongoDB, des travaux liés à XML et les métriques utilisées en Génie logiciel pour la qualité du code. La définition des métriques est complétée par un scénario de validation.*

ABSTRACT. *Document oriented bases allow high flexibility in data representation. This facilitates a rapid development of applications and allows many possibilities for data structuration. Nevertheless, the structuration choices remain crucial because they impact several aspects of the document base and application quality, e.g, memory print, data redundancy, querying and navigation facility and performances, readability and maintainability. It is therefore important to be able to analyse and to compare several data structuration alternatives. In this paper, we propose a set of structural metrics of JSON documents. These metrics work on the structure (not the data) considered as a schema. They mesure several aspects of the complexity of the structure in order to be used in criteria helping in the schema design process. This work capitalises on experiences with MongoDB so as proposals for XML and software quality. This paper presents the definition ot the metrics together with a validation scenario.*

MOTS-CLÉS : *NoSQL, métriques structurelles, systèmes orientés document, MongoDB*
KEYWORDS: *NoSQL, structural metrics, document-oriented systems, MongoDB*

*. Institute of Engineering Univ. Grenoble Alpes

1. Introduction

De nos jours, les applications et systèmes d'information doivent gérer des larges quantités de données hétérogènes tout en répondant à des exigences fonctionnelles variées et à des besoins de performance et de passage à l'échelle. Les systèmes de gestion de données NoSQL apportent diverses solutions et offrent, pour la plupart, beaucoup de souplesse dans la structuration des données. Ils permettent une structuration des données avec une grande flexibilité et sans création préalable d'un schéma (contrairement aux SGBD relationnels). Dans ces solutions il n'y a pas de séparation claire des couches logiques et physiques.

Nos travaux portent sur les systèmes orientés documents, et plus précisément ceux utilisant JSON, MongoDB notamment. Dans ces systèmes, les données peuvent être structurées dans des collections de documents avec attributs atomiques ou de types complexes. La flexibilité dans la structuration des données est appréciée pour permettre un développement initial rapide. Cependant, on constate que les possibilités de structuration des données sont nombreuses et que le choix de structuration adopté reste assez crucial par son impact potentiel sur plusieurs aspects de la qualité des applications (Gómez *et al*., 2016). Le problème est comment définir ou analyser si une structure est bien adaptée aux besoins des applications. En effet, chaque structuration peut présenter des avantages et des inconvénients différents en matière d'empreinte mémoire, redondance de données engendrée, coût de navigation dans les structures et d'obtention de certaines données, lisibilité des programmes, parmi d'autres.

Il devient alors intéressant de pouvoir considérer plusieurs structurations candidates pour retenir un choix unique, ou temporel, ou plusieurs alternatives parallèles selon les cas. Effectuer une analyse et comparaison de plusieurs choix de structuration n'est pas évident, d'une part, par le nombre potentiellement grand de structurations possibles et, d'autre part, par l'absence de critères objectifs d'analyse [1]. Notre travail est une contribution dans ce sens. Nous cherchons à faciliter la compréhension, l'évaluation et la comparaison des structures de données orientés documents (JSON/BSON) où les possibilités sont nombreuses. Nous proposons d'abstraire et de travailler avec une notion de "schéma" de données même si MongoDB ne le traite pas en tant que tel. L'objectif est de clarifier les possibilités et caractéristiques de chaque "schéma" et de donner des critères objectifs pour l'évaluer et apprécier ses avantages et ses inconvénients.

La principale contribution de cet article est la proposition d'un ensemble de métriques structurelles pour des "schémas" de documents JSON. Ces métriques permettent de refléter la complexité des schémas et peuvent être utilisées pour établir des critères de qualité tels que leur lisibilité et maintenabilité. La définition de ces métriques s'appuie, entre autres, sur des expérimentations avec MongoDB, des travaux liés à XML et des métriques utilisées en Génie logiciel pour la qualité du code.

1. Pour le modèle de données orienté documents, on ne dispose pas à l'heure actuelle de critères analogues aux anomalies de conception et la normalisation du modèle relationnel.

Ce travail est partie du projet SCORUS, plus vaste, qui vise à assister les utilisateurs dans un processus de modélisation de schéma avec une approche de recommandations. SCORUS permet de (1) générer un ensemble de schémas semi-structurés orientés documents pour un modèle de données UML; (2) analyser ces schémas à l'aide des métriques proposées dans cet article; (3) proposer un top k de schémas semi-structurés selon les préférences identifiées. Cet article se concentre sur l'étape (2), l'analyse des schémas par la proposition d'un ensemble des métriques permettant de les comparer.

Dans la suite, la section 2 rappelle certains éléments de MongoDB et la motivation de nos travaux. Dans la section 3, nous présentons les métriques structurelles proposées pour mesurer les schémas. La section 4 fournit un scénario d'expérimentation qu'utilise les métriques pour comparer des schémas. Les travaux connexes sont décrits en section 5. Nos conclusions et perspectives de recherche sont présentées en section 6.

2. Contexte et Motivation

Comme déjà mentionné nous nous intéressons à des questions de qualité de structuration de données BSON/JSON au sein de systèmes type MongoDB. Rappelons que dans ce système (comme dans la plupart des NoSQL) il n'y a pas de gestion explicite d'un schéma de données. La manière de structurer les données reste néanmoins importante car elle a un impact sur plusieurs aspects de la base de documents et des applications qui les utilisent.

Le format BSON, gère les données comme un ensemble de collections de documents (voir figure 1a, collections *Agencies* et *BusinessLines*). Un document est simplement un ensemble de paires `attribut:valeur`. Le type des valeurs peut être atomique ou complexe. Par complexe nous entendons soit un tableau de valeurs de tout type ou un document qui dans ce cas est dit *imbriqué*. Notons que la valeur d'un attribut peut être l'identifiant d'un document ou la valeur d'un attribut d'un document d'une autre collection. Cela permet de *référencer* un ou plusieurs documents.

Ce système de types est à la fois simple et puissant car il permet beaucoup de flexibilité dans la création de structures complexes. Dans un cas simple comme celui de notre exemple, l'association 1-N entre Agence et "BusinessLines", est susceptible d'être représentée de plusieurs manières. La figure 1a montre un choix pour la collection *Agencies* qui utilise des références vers *BusinessLines* alors que le choix illustré sur 1b imbrique les documents correspondant aux "business". Sur 1b la collection *BusinessLines* n'est pas créée et il n'y a pas de duplication de données.

De manière générale, tout en garantissant la complètude des données, les collections peuvent être structurées et reliées de divers manières, e.g. collections séparées et sans imbrication, collections complètement imbriquées, combinaison d'imbrication et de referencement ou duplication de données. Le choix de la meilleure structuration n'aura probablement pas une réponse unique ni absolue et dépendra des priorités et besoins d'accès du moment.

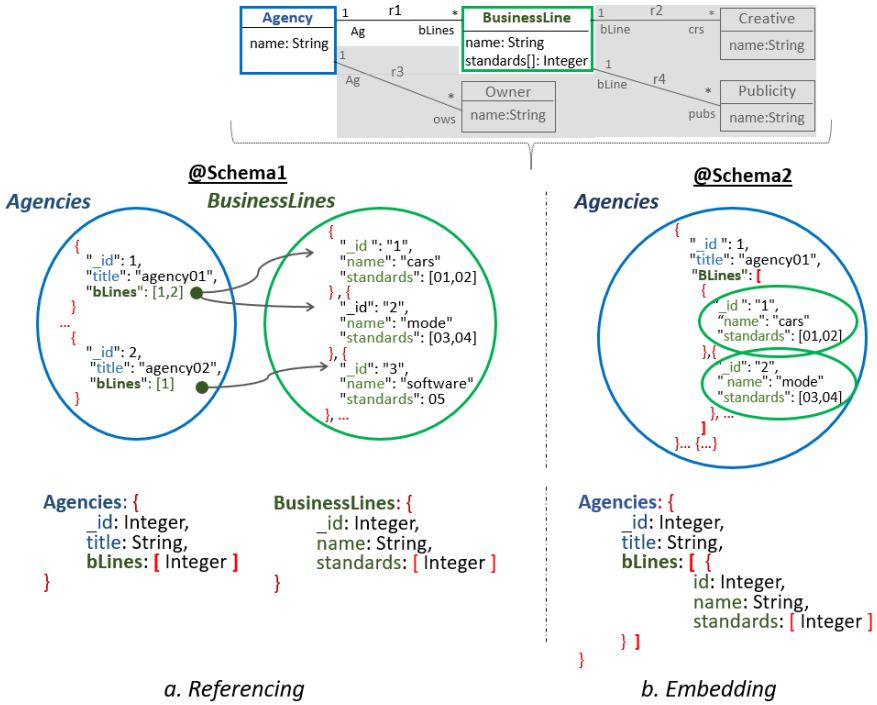


Figure 1. Exemples d'instances MongoDB et abstractions de leurs schémas basés sur un modèle UML

La manière dont sont structurées les données a un fort impact sur la taille de la base, les performances des requêtes et la lisibilité du code des requêtes, ce qu'influence la maintenabilité et l'usabilité de la base ainsi que de ses applications. Cela a été constaté de manière expérimentale (Gómez *et al.*, 2016) où plusieurs patrons de comportement ressortent. Notamment, les collections avec des documents imbriqués sont favorables aux requêtes qui suivent l'ordre de l'imbrication. Cependant, l'accès aux données dans un autre ordre est désavantagé et les performances des requêtes nécessitant l'accès à des données intégrées à différents niveaux dans la même collection sera pénalisé. La raison est que la complexité des manipulations requises dans ce cas, est proche de celle des jointures de plusieurs collections. En outre, les collections avec des documents imbriqués ont une empreinte mémoire plus importante que la représentation équivalente avec des références. Dans le choix de la structuration des données, les priorités peuvent s'avérer divergentes, comme le souhait de dupliquer des documents dans plusieurs collections parce qu'ils sont consultés dans des contextes différents mais aussi le souhait de réduire le coût du stockage.

Nous avons analysé un certain nombre de caractéristiques critiques sur les structures afin d'établir des critères qui aident dans le choix d'un schéma. Dans la suite nous proposons des métriques mesurables sur les schémas de manière à avoir des éléments objectifs d'appréciation et de comparaison au regard des priorités de l'utilisateur.

3. Métriques Structurelles

Dans cette section nous proposons un ensemble de métriques structurelles qui reflètent des aspects clés de la complexité des "schémas" semi-structurés. L'objectif est de faciliter leur analyse et comparaison sans création de la base de données. Des informations statistiques sur les données peuvent bien sur compléter l'analyse lorsqu'elles sont disponibles. La table 1 résume les métriques proposées. Les sections 3.2 à 3.6, définissent et illustrent ces métriques. Dans la suite φ dénote une collection, t un type de document et x un schéma.

Tableau 1. Métriques structurelles proposées

Catégorie	Nom des métriques	Description	Par		
			sch	Col	type
Existence	*colExistence*	Existence d'une collection		x	
	docExistence	Existence d'un type de document dans une collection		x	x
Imbrication	*colDepth*	Profondeur maximale d'une collection		x	
	globalDepth	Profondeur maximale d'un schéma	x		
	DocDepthInCol	Niveau où un type de document se trouve dans une collection		x	x
	maxDocDepth	Niveau le plus profond où apparaît un type de document	x		
	minDocDepth	Niveau le mois profond où apparaît un type de document	x		
Largeur	*docWidth*	"Largeur" d'un type de document		x	x
Référencement	*refLoad*	Nombre de références à une collection		x	
Redondance	*docCopiesInCol*	Copies d'un type de document t dans une collection		x	x
	docTypeCopies	Nombre d'utilisations d'un type de document	x		

Afin de faciliter la manipulation des variantes de schéma (dans le cadre de la génération automatique) et d'évaluer les métriques, nous utilisons la structure de graphe présentée en sous-section 3.1.

3.1. Structure de graphes

Nous considérons un modèle de données UML, avec un ensemble de classes $E = \{e_1, ..., e_n\}$. Les propriétés ou attributs d'une classe e_i sont désignés par le type te_i et ses associations par l'ensemble $R(e_i) = \{r_1, ..., r_n\}$. Pour chaque association on connaît le rôle des entités reliées. Dans la suite nous appelons schéma semi-structuré, l'ensemble de collections et le type des documents qui seront utilisés dans la base pour représenter les données. Différents schémas sont envisageables pour un même modèle entité-association.

Afin de faciliter le calcul des métriques d'un schéma, nous construisons un arbre tel qu'illustré sur la figure 2 pour chaque schéma. Le noeud racine, @ShemaX, a un noeud fils par *collection* présente dans le schéma (e.g . collections *Agencies, Creatives, Owners*). Il s'agit de collections de documents dont le type est représenté par le sous-arbre fils, noeud avec sous-fixe l_0 (e.g. $tAgency@l_0$ pour documents agency au niveau

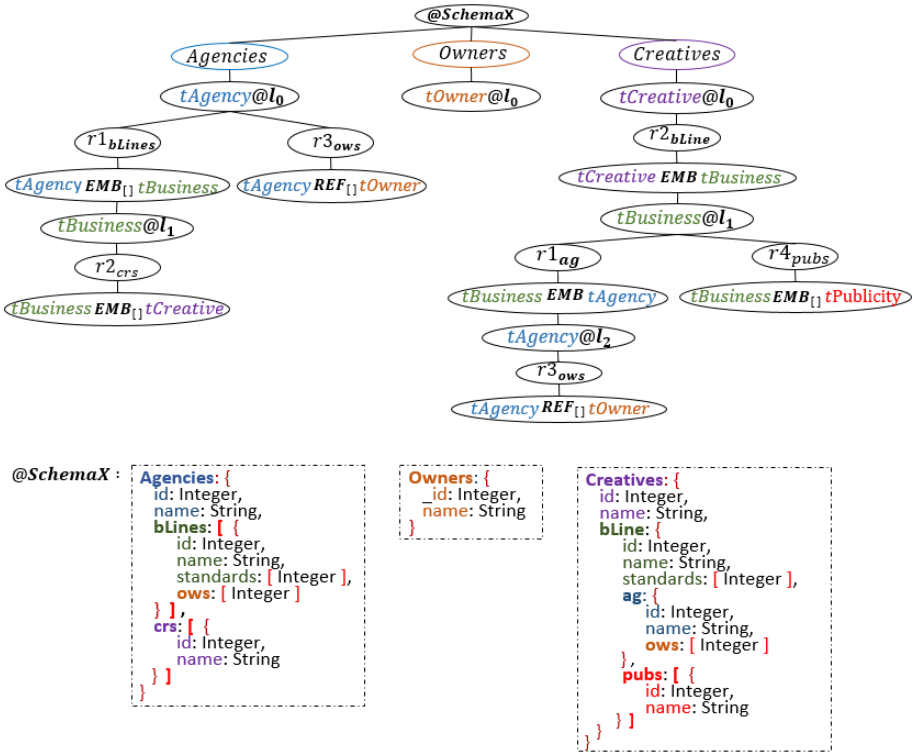


Figure 2. Exemple de représentation des graphes

0). Dans ce sous-arbre, les attributs de type atomique sont dans ce noeud $@l_0$ [2], et les attributs complexes (imbrication ou référence de documents) sont exprimés dans les noeuds fils. Les références sont matérialisées par des *noeuds REF* et les imbrications de documents par des *noeuds EMB* (e.g. $tAgencyEMB[]tBusiness$).

Lors de la création de schémas semi-structurés, les attributs complexes sont utilisés pour stocker les associations $R(e_i)$ des classes e_i. Un noeud avec le nom de l'association est créé (e.g. $r1_{bline}$) avec pour fils un noeud *REF* ou *EMP* selon le choix. Des arrays, notés $[]$, peuvent être utilisés pour les association n-aires. Par exemple, pour les business d'une agence, association $r1_{bline}$, le document agency aura un attribut de type array de documents business (noeud $tAgencyEMB[]tBusiness$).

Les imbrications de documents induisent un niveau de profondeur supplémentaire dans la structure. Ceci est matérialisé dans l'arbre par un *noeud niveau* l_i qui permettra de savoir facilement à quel niveau de profondeur se trouvera un document (e.g. $tBusiness@l_1$). Concernant les références, elles apparaissent à différents niveaux et référencent le type apparaissant au niveau 0 d'une collection.

2. Attributs non listés sur la figure

3.2. Métriques d'existence

Le choix de créer une collection pour un type de document sera motivé principalement par le besoin d'accès rapide ou fréquent à l'extension du type ou à un document du type en question. Au contraire l'imbrication d'un type de document dans un autre peut être motivé par le fait que l'information est fréquemment consultée ensemble. S'assurer qu'un type de document n'est pas imbriqué à certains endroits peut aussi être intéressant, notamment si le document est peu accédé dans ce contexte ou si l'on cherche à réduire la complexité d'une collection. Dans cette catégorie nous définissons des métriques qui reflètent l'existence d'un type de document t dans un schéma. Nous considérons deux cas : (1) la existence d'une collection de documents de type t, et (2) la présence de documents de type t imbriqués dans d'autres documents. Ces cas sont couverts respectivement, par la métrique *colExistence* et la métrique *docExistence* définies ci-après.

Existence de collection :

$$colExistence(t) = \begin{cases} 1 & \text{le noeud } t@l_0 \text{ apparaît dans le schéma} \\ 0 \end{cases} \quad (1)$$

Existence de type imbriqué : l'imbrication de documents de type t est matérialisé dans le graphe par un noeud $*EMB\,t$.

$$docExistence(\varphi, t) = \begin{cases} 1 & t \in \varphi \quad \text{un noeud } *EMB\,t \text{ apparaît dans un chemin partant de } \varphi \\ 0 & t \notin \varphi \end{cases} \quad (2)$$

Notons sur la figure 2 que pour les types *tAgency, towners et tCreative* il y a des collections (noeuds $@l_0$) alors que ce n'est pas le cas pour *tPublicity*. Ceux-ci existent exclusivement imbriqués dans des documents *tCreative*. Notons aussi que des documents de type *tBusiness* sont imbriqués dans deux collections, *Agencies et Creatives*. Nous verrons dans la suite que la prise en compte de ce fait peut s'avérer pertinent dans l'analyse des schémas.

3.3. Métriques d'imbrication

En général, plus une information sera imbriquée profondément, plus il sera coûteux d'y accéder sauf si l'information intermédiaire est aussi recherchée par la requête. Savoir à quel niveau d'imbrication apparaît un type de document permet d'évaluer les coûts de navigation et d'aller-retour entre les niveaux ("intra-joint") pour y accéder, ou des opérations de restructuration nécessaires pour extraire le format le plus approprié. Cette catégorie est consacrée aux métriques qui indiquent le niveau d'imbrication des documents.

Profondeur d'une collection : la métrique *colDepth* (3) indique le niveau de profondeur où se trouve le document le plus imbriqué. L'imbrication des documents est représentée par les noeuds EMB dans le graphe.

$$colDepth(\varphi) = max(depth(p_i)) \quad p_i \text{ est un chemin partant du noeud } \varphi \tag{3}$$

$$depth(p) = n \qquad \text{nombre de noeuds } EMB \text{ dans le chemin } p \tag{4}$$

Profondeur d'un schéma : la métrique *globalDepth* (5) indique le niveau d'imbrication le plus profond des collections d'un schéma.

$$globalDepth(x) = max(colDepth(\varphi_i)) \qquad \forall \text{ collection } \varphi_i \in x \tag{5}$$

Connaître la profondeur d'imbrications des collections aide à mieux cerner leur cas d'utilisation et à estimer la pertinence de la structure. Les imbrications successives contribuent à une certaine forme de complexité mais n'implique pas forcement des requêtes moins performantes. Une collection très imbriquée peut être avantageuse si des requêtes prioritaires nécessitent une majorité des informations imbriquées. Si c'est pas le cas, l'impact des opérations de projections sera à prendre en compte (voir métriques suivantes) ainsi que la restructuration des données pour la réponse si le chemin d'accès de la requête et le sens d'imbrication des données ne coïncident pas.

Sur l'exemple, la profondeur des collections *Owners, Agencies et Creatives* est 0, 2, et 2 respectivement. La profondeur maximale du schéma est de 2. Notons que dans la collection *Creatives*, le type *tAgency* n'ajoute pas de niveau d'imbrication, il ajoute uniquement un tableau avec des références sur *Owners*.

Profondeur d'un type de document : la métrique *docDepthInCol* (6) indique le niveau où apparaît un type de document t dans une collection φ. Si les éléments de la collection sont de type t (noeud $t@l_0$) la profondeur est zéro, sinon on cherche le niveau le plus profond où est imbriqué un document de ce type (noeuds $EMB\,t$) en suivant les chemins racine-feuilles.

$$docDepthInCol(\varphi, t) = \begin{cases} 0 & \text{le noeud fils de } \varphi \text{ est } t@l_0 \\ max(docDepth(p_i, t) & p_i \text{ est un chemin de la racine } \varphi \text{ à une feuille} \end{cases} \tag{6}$$

$$docDepth(p, t) = n \quad \text{nombre de noeuds } EMB \text{ entre la racine et } *\,EMB\,t \tag{7}$$

Par exemple, dans la collection *Creatives*, le niveau d'imbrication de *tPublicity* est 2, celui de *tCreative* est 0. *tCreative* est aussi imbriqué au niveau 2 de la collection *Agencies*. Nous introduisons également les métriques *maxDocDepth* (8) et *minDocDepth* (9) qui indiquent le niveau le plus et le moins profond où le type de document apparaît dans le schéma.

$$maxDocDepth(t) = max(DocDepthInCol(\varphi_i, t)) \quad \varphi_i \in x \ \wedge \ t \in \varphi_i \tag{8}$$

$$minDocDepth(t) = min(DocDepthInCol(\varphi_i), t) \quad \varphi_i \in x \ \wedge \ t \in \varphi_i \tag{9}$$

Connaître les niveaux minimum et maximum permet d'estimer combien de niveaux intermédiaires il faut traiter pour l'accès le plus ou le moins direct. Sur l'exemple, notons qu'il n'y a pas de collection de documents *tBusiness*, *minDocDepth(tBusiness)* = 1.

3.4. Largeur des documents

Ici nous nous intéressons à la complexité d'un type de document en termes du nombre d'attributs et de leur type, atomique ou complexe (documents ou arrays de documents imbriqués). Ces métriques sont motivées par le fait que des documents avec plusieurs attributs complexes peuvent induire des opérations d'accès et de projection plus conséquentes. En effet, pour extraire les attributs nécessaires à l'évaluation d'une requête, il est nécessaire d'enlever les autres attributs ce qui s'avère plus coûteux pour des document "larges". Lors de l'évaluation d'un schéma, on pourra notamment analyser le choix d'une structure à la fois "large" et très imbriquée.

La métrique *docWidth* (10), reflète la "largeur" d'un type de document en se basant sur le nombre d'attributs atomiques (coefficient a=1), le nombre d'attributs qu'imbriquent un document (coefficient b=2), le nombre d'attributs de type array de valeurs atomiques (coefficient c=1) et array de documents (qui ont plus de poids, coefficient d=3).

$$\begin{aligned} docWidth(t,\varphi) = {} & a * nbrAtomicAttributes(t,\varphi) + \\ & b * nbrDocAttributes(t,\varphi) + \\ & c * nbrArrayAtomicAttributes(t,\varphi) + \\ & d * nbrArrayDocAttributes(t,\varphi) \end{aligned} \tag{10}$$

Les métriques indiquant le nombre d'attributs peuvent être utilisées séparément selon les analyses souhaités. La taille des arrays n'est pas prise en compte ici car elle n'est pas forcément disponible. Si c'est le cas, il semble intéressant de différencier les ordres de grandeur des arrays. Des arrays de taille 4 ou 5 sont du même ordre de grandeur, contrairement à des arrays prévus pour de milliers d'éléments.

Les collections textitAgencies et *Creatives* de l'exemple, utilisent des document de type *tBusiness* mais ils n'ont néanmoins pas les mêmes attributs. Dans Creatives le type inclut des arrays d'agences et publicity, *docWidth(Creatives,tBusiness)* = 8 , contrairement à *Agencies* où *docWidth(Agencies,tBusiness)* = 4.

3.5. Taux de référencement

Le maintien de l'intégrité référentielle devient un problème pour les collections dont les documents sont beaucoup référencés par de documents d'autres collections. Pour une collection avec des documents d'un certain type t, la métrique *refLoad* (11) indique le nombre d'attributs (d'autres types) qui sont des références potentielles sur les documents de type t.

$$refLoad(\varphi) = n \qquad \text{soit } t@l_0 \text{ le noeud fils de } \varphi\text{, n est le nombre de noeuds } *REF\ t \text{ du schéma} \quad (11)$$

Pour la collection *Owners* de notre exemple, son type est référencé par 2 collections: *Agencies* la référence au niveau 0 alors que *Creatives* la référence dans un document imbriqué au niveau 2.

3.6. Métriques de redondance

Nous nous intéressons ici à la redondance de données qui peut exister dans la base. La redondance des documents peut accélérer les accès et limiter certaines opérations coûteuses (par exemple des jointures). Cependant, elle impacte l'empreinte mémoire de la base et sa maintenabilité en matière de cohérence (complexité d'écriture des programmes et coût). Pour la métrique de redondance *docCopiesInCol* (12), nous utilisons l'information de cardinalité des associations conjointement avec les choix faits sur le schéma. La redondance apparaît pour certains cas de représentation de l'association par imbrication de documents.

$$docCopiesInCol(t,\varphi) = \begin{cases} 0 & : t \notin \varphi\ docExistence(\varphi,t)=0 \\ 1 & : \text{ le noeud fils de } \varphi \text{ est } t@l_0 \\ \prod card(r_{rol},t) & : \begin{array}{l} r_{rol} \text{ parent de un noeud } EMB \\ \text{dans le chemin entre } \varphi \text{ et } *EMBt \end{array} \end{cases} \quad (12)$$

$$card(r,\varepsilon) = n \qquad \text{cardinalité de } r \text{ du côté } \varepsilon \text{ dans le modèle UML} \quad (13)$$

Dans la collection *Creatives* du schéma de la Figure 2, l'attribut pour business, nommé *bline*, introduit de la redondance pour les agences. Par l'association r1 une agence A peut être associée à n1 business. Il y aura donc autant de copies du document A. Si de plus un business est référence par n2 creatives (association r2) , il y aura n1 x n2 copies du document A.

Par ailleurs, nous proposons la métrique *docTypeCopies(t)* qui indique le nombre de fois qu'un type de document est utilisé dans le schéma. Ceci reflète le nombre de structures qui peuvent potentiellement stocker des documents de type t. Cette métrique utilise la métrique d'existence.

4. Scénario de validation

Comme déjà dit, ce travail s'inscrit dans le cadre du projet SCORUS, qui vise à assister les utilisateurs dans les processus de choix de structuration des documents. Pour un modèle de données UML, SCORUS permettra (1) de générer automatiquement un ensemble de variantes de "schémas" JSON possibles et (2) de donner les métriques pour chacun d'eux. Tenant compte des priorités des applications et des requêtes fréquentes, ces métriques seront utilisées pour établir des critères de choix et comparer les schémas

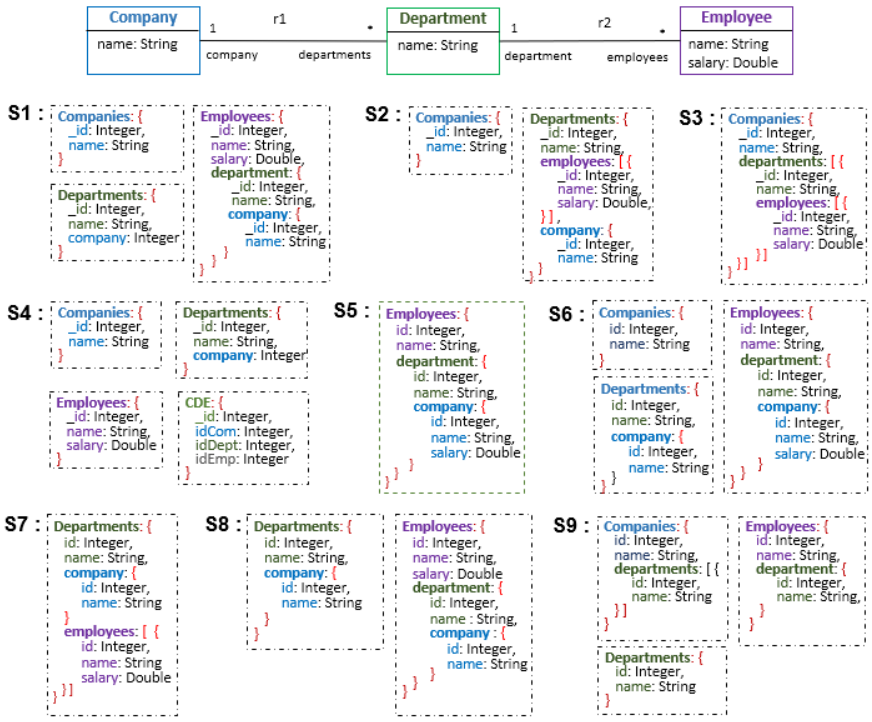


Figure 3. Ensemble de schémas étudiés

entre eux. Il s'agit en priorité de faire émerger le ou les schémas les plus favorables selon certains critères mais aussi d'écarter des choix très défavorables ou encore, d'envisager des schémas alternatifs qui n'étaient pas forcement considérés au départ. Ceci est arrivé lors de notre expérimentation où une des alternatives générées automatiquement s'est avéré être pertinente alors qu'elle ne faisait pas partie des choix "naturels" du développeur.

Dans la suite nous considérons un scénario d'utilisation des métriques proposées. Nous utilisons un cas simple, voir figure 3, pour lequel 9 variantes de structuration JSON sont étudiées. Ce cas a été utilisé lors d'une expérimentation avec des bases MongoDB (Gómez *et al.*, 2016) où l'impact de la structuration des schémas ressort. Nous nous plaçons dans le même contexte applicatif et reprenons des informations sur l'accès aux données afin de les utiliser dans l'analyse des variantes des "schémas" étudiés. Nous avons évalué les métriques pour les 9 schémas. Celles que nous utilisons ici sont indiquées sur la figure 2.

D'un point de vue applicatif, nous disposons des informations suivantes. Les requêtes les plus prioritaires portent sur les entreprises et le nom de leurs départements (priorité forte) mais aussi pour connaître l'employé qui a le salaire le plus élevé dans l'entreprise en donnant l'identifiant de l'entreprise ou le nom de l'entreprise.

Tableau 2. Évaluation des schémas

Métriques \ Schéma	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
colExistence(tCompany)	1	1	1	1	0	1	0	0	1
docCopies(tCompany)	1	1	1	1	1	3	1	1	1
refLoad(Employees)	0			1	0	0		0	0
colExistence(tCompany)	1	0	0	1	1	1	0	0	1
docWidth(Companies,l1)	1	1	3	1		1			3
docExistence(tDepartment,Companies)	0	0	1	0		0			1

Ces informations sur les accès prioritaires ainsi que d'autres informations permettent d'établir des critères d'analyse des schémas. Tenant compte les accès prioritaires, la collection de Companies joue un rôle important (critère1) ainsi que la facilité pour manipuler les instances (critère 5). Les départements sont accèdés via les entreprises (critère 6). De plus on sait que la cohérence des données sur les entreprises est importante. Il est donc préférable de limiter les copies pour ces données (critère 2). Par ailleurs, l'accès à l'ensemble d'employés (critère 4) exclusivement n'est pas prioritaire.

La table 3 illustre la formalisation de certains critères. Chaque ligne montre l'évaluation d'un critère sur les 9 schémas alternatifs étudiés. Les valeurs ont été normalisées (entre 0 et 1) et introduisent un ordre relatif entre les schémas. Par exemple au regard du critère 4, les schémas S1, S4, S5, S6 et S9 sont à privilégier par rapport aux autres.

Tableau 3. Critères

	Critère \ Schéma	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
1	$f_{c_1}(s) = colExistenceCompanies(s)$	1.00	1.00	1.00	1.00	0.00	1.00	0.00	0.00	1.00
2	$f_{c_2}(s) = docCopiestCompany^{min}(s)$	1.00	1.00	1.00	1.00	1.00	0.33	1.00	1.00	1.00
3	$f_{c_3}(s) = refLoadEmployees^{max}(s)$	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
4	$f_{c_4}(s) = colExistenceEmployees(s)$	1.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	1.00
5	$f_{c_5}(s) = levelWidthCompaniesL_1{}^{min}(s)$	1.00	1.00	0.33	1.00	0.00	1.00	0.00	0.00	0.33
6	$f_{c_6}(s) = docDptInCompanies^{min}(s)$	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00

L'analyse des schémas est multi-critères (6 critères dans notre cas). Les critères peuvent avoir le même poids ou bien on peut privilégier certains critères. La fonction d'évaluation d'un schéma, noté *schemaEvaluation* fait la somme pondérée des critères.

$$schemaEvaluation(s) = \sum_{i=1}^{|Criteria|} weight_{criterion_i} * f_{criterion_i}(s) \qquad (14)$$

Nous avons évalué avec trois pondérations différents: même poids pour tous les critères (cas1), priorité aux critères concernant la facilité d'utilisation de companies (cas2), ajout de priorité de la collection employée en supposant qu'il est motivé par un nouveau patron d'accès (cas 3). La figure 4 montre le résultat de l'évaluation des 9 schémas pour les trois cas.

Criteria	Factor		
	Case1	Case2	Case3
Criterion 1	16.7	50	30
Criterion 2	16.7	10	10
Criterion 3	16.7	0	0
Criterion 4	16.7	0	20
Criterion 5	16.7	15	15
Criterion 6	16.7	25	25

a. Ensembles des poids

Schema / Cas	s1	s2	s3	s4	s5	s6	s7	s8	s9
Cas 1	83.33	66.67	72.22	66.67	50.00	72.22	33.33	33.33	88.89
Cas 2	75.00	75.00	90.00	75.00	10.00	68.33	10.00	10.00	90.00
Cas 3	75	55	70	75	30	68.33	10	10	90

b. Evaluation des schémes par cas

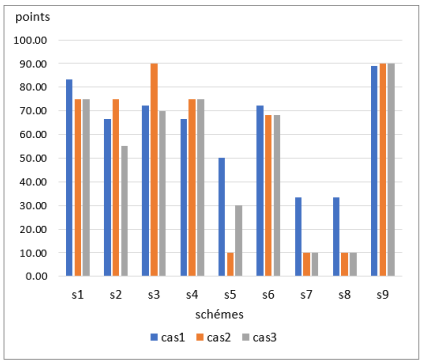


Figure 4. Évaluation des schémas

Les évaluations placent les schémas S5, S7, S8 comme les moins bons dans les trois cas considérés. La structuration dans S5 et S7 se base sur une seule collection qui n'est pas prioritaire dans les critères pris en compte. Alors que S3 se démarque dans le cas 2, pour la forte priorité de la collection *Companies*, seule collection de S3 qu'imbrique des documents en accord avec les critères.

S9 et S6, parmi d'autres schémas, sont stables dans leurs scores pour les trois cas. S9 est le meilleur car il correspond à tous les critères. Les bons résultats aux trois cas dénote une forme de "polyvalence" du schéma qui permet de résister aux évolutions des priorités. S6 paie le coût de ne pas considérer l'imbrication dans la collection *Agencies* et d'introduire de la redondance de documents alors qu'on préfère l'éviter (critère 2).

Les critères à considérer et leur poids [3] dépendent du contexte applicatif mais peuvent aussi correspondre à de bonnes pratiques préconisées pour le développement ou à une priorité générale. Par exemple, adopter des schémas très "compacts" pour limiter l'empreinte mémoire lorsque des données seront peu utilisées. Ou, en donnant priorité à la qualité logicielle, privilégier les schémas les plus "lisibles". Sachant que les critères peuvent diverger et évoluent certainement, l'utilisation des métriques et critères pour un choix de schéma peut aider dans un processus continue de "tuning" de la base qui peut conduire à des évolutions de la structuration ou à la création de copies des données avec des structures différentes. Pendant un certain temps, une base pourrait avoir, pour les mêmes données, une copie avec une schéma Sx et une autre copie avec un schéma Sy.

5. Travaux connexes

Dans l'étude de l'état de l'art, nous nous sommes intéressés aux travaux concernant des système NoSQL ainsi qu'à des propositions antérieures pour des données

3. Le choix des poids est un sujet qui reste à approfondir

complexes, des documents XML et des métriques logicielles. (Klettke *et al*., 2002) s'appuie sur le modèle de qualité de software ISO 9126 et adapte cinq métriques pour évaluer des documents XML en travaillant sur la DTD. Ces travaux nous ont servi de point de départ. Ils travaillent sur une représentation sous forme de graphe et les métriques considèrent le nombre de références, noeuds et font un rapprochement avec la complexité cyclomatique (McCabe, 1976). (Pušnik *et al*., 2014) propose 6 métriques associées chacune à un aspect de qualité telles que la structure, la clarté, l'optimalité, le minimalisme, la réutilisation et la flexibilité. Ces métriques utilisent 25 variables qui mesurent le nombre d'éléments, d'annotations, de références et de types parmi d'autres. (Li, Henry, 1993 ; Chidamber, Kemerer, 1991 ; McCabe, 1976) travaillent sur des métriques de logiciel avec paradigmes procedurale et orienté objet. Plusieurs métriques sont proposées pour refléter notamment les niveaux de couplage entre composants et classes, la taille des objets, des hiérarchies de classes et le nombre de méthodes. Nous avons étendu et adapté ces propositions pour notamment prendre en compte les particularités d'imbrication de documents et types d'attributs JSON en vue d'une utilisation dans une base de documents telle que MongoDB.

(Mior *et al*., 2017) et (Lombardo *et al*., 2012) s'intéressent aux alternatives de "modélisation" des données dans Cassandra (modèle "big table") avec des objectifs d'étude sur le stockage et les performances de requêtes implémentés avec SET et GET. (Lombardo *et al*., 2012) propose la création de plusieurs versions des données avec des structures différentes chacune étant plus adaptée pour l'évaluation d'une requête différente dans l'esprit des requêtes pré-calculées. Dans ce travail, aucune métrique n'est proposée pour évaluer les différentes versions. (Zhao *et al*., 2014) propose un algorithme pour la création systématique d'une base de données orientées documents à partir du modèle entité-relation. Cet algorithme propose une dé-normalisation de ce modèle avec pré-calcul des jointures naturelles par imbrication de documents. Le schéma résultant correspond au modèle du schéma S6 de notre scénario de validation. Ce choix engendre en général de la redondance des données. (Zhao *et al*., 2014) propose une métrique pour cela en utilisant la connaissance du volume des données. Dans cet article, nous proposons un ensemble plus large de métriques structurelles qui inclut deux métriques pour analyser la redondance sans connaissance du volume des données. Si les informations sur les données sont disponibles, elles peuvent être utilisées en complément.

Des travaux récents s'intéressent à l'analyse du schéma d'une base de données déjà implémentée. (Gallinucci *et al*., 2018) propose d'abstraire le schéma en considérant les variantes présentes des documents dans une collection et introduit une métrique d'entropie qui permet de définir la précision avec laquelle les documents ont été classés. Dans les outils existants, notons MongoDBCompass [4] qui permet de monitorer le temps d'exécution des requêtes, de connaître le volume des données d'une collection de documents et d'extraire des informations par rapport à la structure d'une collection. Cela fonctionne donc sur des bases déjà opérationnelles. Nous avons mentionné JSON

4. MongoDBCompass, https://docs.mongodb.com/compass/master/. Accessed: 2018-02-12.

schema [5] , qui est le résultat d'efforts pour faciliter la validation de documents JSON. Certains outils analysent des documents JSON dans le but d'abstraire un "schéma" permettant d'identifier les collections et les types sous-jascents.

D'autres travaux, sans formaliser ou suggérer des métriques sur des schémas semi-structurés, fournissent des lignes directrices, des bonnes pratiques et des aspects à prendre en compte dans le choix des structures. (Abiteboul, 1997) fournit des aspects à considérer pour les données semi-structurées et un aperçu de propositions de modèles et de langages de requête pour ces données. (Sadalage, Fowler, 2012) discutent divers modèles de données et produits NoSQL (MongoDB, Cassandra et Neo4j) et quelques problématiques de migration d'une base relationnelle vers des *BigTables*, documents et graphes. (Copeland, 2013) et MongoDB [6] proposent des lignes directrices pour la création de bases MongoDB en s'appuyant sur des cas appliqués à différents domaines. Ces bonnes pratiques peuvent être formalisées dans les critères que nous proposons afin d'être prises en compte dans l'analyse des schémas.

6. Conclusion et perspectives

Dans ce travail, nous nous sommes intéressés à des questions de qualité des structures de données pour des bases de documents JSON, tel que MongoDB. La flexibilité de structuration de ces bases est appréciée par la souplesse qu'elle permet pour représenter des données semi-structurés. Cependant, cette flexibilité a un coût dans les performances, le stockage, la lisibilité et la mantenabilité des bases et des applications. Ainsi, le choix de la structuration des données est très importante et ne doit pas être négligé. En considérant des travaux en génie logiciel et dans le contexte des bases orienté objet et XML, nous avons défini des métriques structurelles pour des "schémas" JSON. Ces métriques, organisées en catégories, reflètent des éléments de complexité du schéma qui jouent sur des aspects de qualité de la base. Elles peuvent ainsi être utilisées pour analyser et comparer différentes manières de structurer les données.

Nous avons présenté un scénario d'utilisation des métriques avec plusieurs variantes de schéma et certains critères et priorités applicatifs. L'analyse avec les critères, permet d'écarter certains schémas et de mettre en avant d'autres. Ces résultats sur les aspects structurels ont été comparés et, sont bien en phase, avec les résultats d'expériences d'évaluation de performances que nous avons menés avec des bases contenant des données. Il est intéressant de noter que lors du travail sur les structures nous avons pu considérer à "faible coût" plus de variantes de schémas que lors de l'expérimentation avec les bases. Cela a apporté un résultat inattendu qui est l'identification d'un schéma différent avec de très bonnes caractéristiques.

Les métriques proposées n'ont pas l'ambition de représenter un ensemble complet mais sont une base qui va probablement évoluer. La suite des travaux, incluent des

5. Json schema, http://json-schema.org/. Accessed: 2018-02-12

6. Rdbms to mongodb migration guide. (2017, Nov). White Paper. Consulté sur https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide

validations à plus grande échelle. Nous allons notamment poursuivre le développement du système Scorus (mentionné en introduction) afin de compléter l'outil de génération automatique de schémas. Nous allons également travailler dans la formalisation d'un système de recommandation pour faciliter la définition des critères en utilisation les métriques, les requêtes fréquentes et autres préférences fonctionnelles ou non fonctionnelles des utilisateurs potentiels.

Remerciements

Nous remercions G. Vega, J. Chavarriaga et C. Labbé pour les échanges autour de ce travail ainsi qu'aux relecteurs anonymes pour leur retours.

Bibliographie

Abiteboul S. (1997). Querying semi-structured data. In *Proceedings of the 6th international conference on database theory*, p. 1–18. London, UK, UK, Springer-Verlag. Consulté sur http://dl.acm.org/citation.cfm?id=645502.656103

Chidamber S. R., Kemerer C. F. (1991). *Towards a metrics suite for object oriented design* (vol. 26) n° 11. ACM.

Copeland R. (2013). *Mongodb applied design patterns*. Oreilly.

Gallinucci E., Golfarelli M., Rizzi S. (2018). Schema profiling of document-oriented databases. *Information Systems*, vol. 75, p. 13–25.

Gómez P., Casallas R., Roncancio C. (2016). Data schema does matter, even in nosql systems! In *Research challenges in information science (rcis), 2016 ieee tenth international conference on*, p. 1–6. Grenoble, France, IEEE.

Klettke M., Schneider L., Heuer A. (2002). Metrics for xml document collections. In *International conference on extending database technology*, p. 15–28.

Li W., Henry S. (1993). Object-oriented metrics that predict maintainability. *Journal of systems and software*, vol. 23, n° 2, p. 111–122.

Lombardo S., Nitto E. D., Ardagna D. (2012). Issues in handling complex data structures with nosql databases. In *14th international symposium on symbolic and numeric algorithms for scientific computing, SYNASC 2012, timisoara, romania, september 26-29, 2012*, p. 443–448. Consulté sur http://dx.doi.org/10.1109/SYNASC.2012.59

McCabe T. J. (1976). A complexity measure. *IEEE Transactions on software Engineering*, n° 4, p. 308–320.

Mior M. J., Salem K., Aboulnaga A., Liu R. (2017). Nose: Schema design for nosql applications. *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, n° 10, p. 2275–2289.

Pušnik M., Heričko M., Budimac Z., Šumak B. (2014). Xml schema metrics for quality evaluation. *Computer science and information systems*, vol. 11, n° 4, p. 1271–1289.

Sadalage P. J., Fowler M. (2012). *Nosql distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

Zhao G., Lin Q., Li L., Li Z. (2014, Nov). Schema conversion model of sql database to nosql. In *P2p, parallel, grid, cloud and internet computing (3pgcic), 2014 ninth international conference on*, p. 355-362.

FURQL : une extension floue de SPARQL

Olivier Pivert [1], **Olfa Slama** [2], **Virginie Thion** [3]

1. *Univ. Rennes 1, IRISA, Lannion, France*
   Olivier.Pivert@irisa.fr
2. *Univ. Rennes 1, IRISA, Lannion, France*
   Olfa.Slama@irisa.fr
3. *Univ. Rennes 1, IRISA, Lannion, France*
   Virginie.Thion@irisa.fr

RÉSUMÉ. *La publication de données ouvertes liées sur le web est un phénomène en pleine croissance. L'étude des modèles et langages permettant l'exploitation de ces données s'est donc grandement intensifiée ces dernières années. Les données publiées sont généralement de nature hétérogène et ne présentent pas de régularité structurelle. De plus, elles sont souvent porteuses de notions graduelles. Dans ce contexte, il est nécessaire de pouvoir proposer des langages de requête aussi flexibles que possible. Dans cet article, nous proposons une extension du langage SPARQL, fondée sur la théorie des ensembles flous, permettant (1) d'interroger une extension floue du modèle de données RDF dans laquelle les triplets sont porteurs de notions graduelles, et (2) d'exprimer des préférences floues portant non seulement sur les données mais également sur la structure du graphe de données, que celui-ci soit flou ou non. Cet article est une version résumée en langue française de l'article (Pivert* et al.*, 2017).*

ABSTRACT. *The Resource Description Framework (RDF) is the graph-based standard data model for representing semantic web information, and SPARQL is the standard query language for querying RDF data. Because of the huge volume of linked open data published on the web, these standards have aroused a large interest in the last years. This paper proposes a fuzzy extension of the SPARQL language that improves its expressiveness and usability. This extension allows (1) to query a fuzzy RDF data model, and (2) to express fuzzy preferences on data and on the structure of the data graph, which has not been proposed in any previous fuzzy extensions of SPARQL. This article is a summarized French version of (Pivert* et al.*, 2017).*

MOTS-CLÉS : *RDF, SPARQL, requêtes floues*

KEYWORDS: *RDF, SPARQL, fuzzy queries*

1. Introduction

Dans sa version classique, SPARQL permet un filtrage booléen des données RDF, n'intégrant pas de préférence utilisateur. Des travaux de la littérature tels que (Cheng *et al*., 2010) et (Ma *et al*., 2015) proposent une extension de SPARQL introduisant des préférences utilisateur. Dans ces approches, les préférences concernent les littéraux portés par les données mais pas la structure du graphe. Il est également nécessaire de pouvoir prendre en compte des graphes RDF dans lesquels les données sont intrinsèquement décrites de façon pondérée. Ce poids peut représenter une notion graduelle telle qu'une intensité, un coût ou un degré d'appartenance. Par exemple, une personne peut être l'amie d'une autre avec un degré croissant en fonction de l'intensité de la relation d'amitié. Le modèle RDF a donc été enrichi par divers auteurs de façon à pouvoir intégrer ce type d'information de façon native, et des langages de requête flexibles portant sur ce modèle enrichi doivent être définis. Notre objectif est d'étendre le langage SPARQL de façon à lui permettre d'exprimer des préférences utilisateur pour exprimer des requêtes flexibles, portant sur des données RDF porteuses ou non de notions graduelles. Nous proposons une extension de la notion de patron de graphe, fondée sur la théorie des ensembles flous, permettant (1) d'interroger une *extension floue du modèle de données RDF* dans laquelle les triplets sont porteurs de notions graduelles, et (2) d'exprimer des *préférences floues* portant non seulement sur les données mais également sur la *structure* du graphe de données. Nous proposons ensuite FURQL, une extension de SPARQL fondée sur ces notions.

Cet article est organisé comme suit. La section 2 introduit les notions nécessaires à la compréhension de la suite. Dans la section 3, qui constitue le cœur de la contribution, nous définissons la notion de *patron flou de graphe*. En nous fondant sur cette notion, nous proposons le langage FURQL et son implantation dans la section 4. Les travaux connexes sont présentés en section 5. Enfin, la section 6 rappelle les contributions de l'article et esquisse quelques perspectives liées à ce travail.

2. Notions préliminaires

Les notions préliminaires introduites ci-dessous concernent les modèles *RDF* et *RDF flou*, ainsi que le langage de requête SPARQL.

Modèle RDF. Le vocabulaire de RDF (W3C RDF, 2014) est composé des ensembles infinis disjoints de noms de ressources, de littéraux et de nœuds blancs (nœuds spéciaux pour lesquels l'URI ou le littéral n'est pas donné) respectivement notés $\mathcal{U}$, $\mathcal{L}$ et $\mathcal{B}$ dans la suite. L'élément RDF de base est le triplet. Un triplet est de la forme $\langle \mathbf{s}, \mathbf{p}, \mathbf{o} \rangle \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$. Le premier élément du triplet, **s**, dit *sujet*, est une ressource décrite; le deuxième élément, **p**, dit *prédicat*, est la propriété attachée à la ressource **s** et le troisième élément, **o**, dit *objet*, est la valeur de la propriété **p** attachée à le ressource **s**. Un triplet indique que le sujet **s** a la propriété **p** avec la valeur **o**. Par exemple, le triplet ⟨`Adele, créateur, Hello`⟩ indique que `Adele` a `Hello` comme propriété `créateur`, ce qui peut être interprété comme `Adele` est `créateur` de `Hello`.

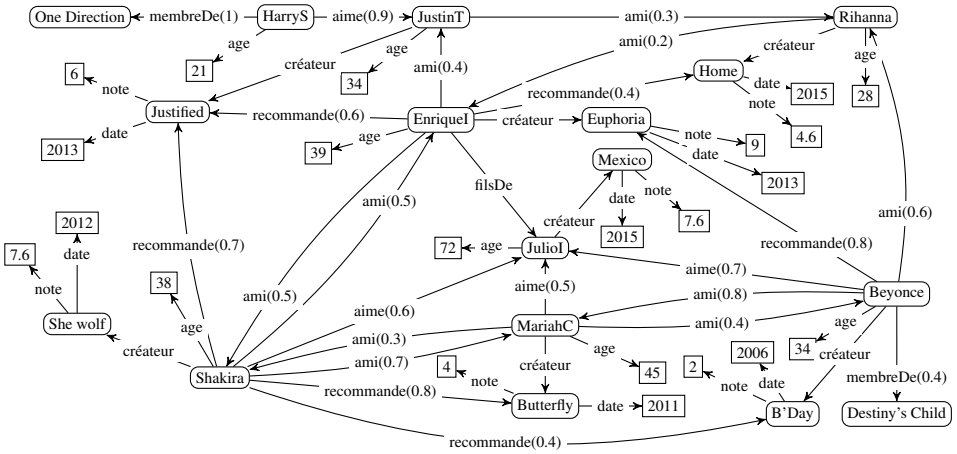


FIGURE *1. Graphe RDF flou G_{MB} inspiré de MusicBrainz*

Un ensemble de triplets RDF peut être représenté sous la forme d'un graphe étiqueté orienté (nommé *graphe RDF* ou simplement *graphe* dans la suite), dans lequel chaque triplet $\langle \mathbf{s}, \mathbf{p}, \mathbf{o} \rangle$ correspond à un arc étiqueté par **p** ayant pour origine **s** et pour destination **o**. La figure 1 en est un exemple. On considère dans la suite que les graphes manipulés sont saturés (les triplets déductibles y sont explicités).

Modèle RDF flou. Les extensions floues du modèle RDF proposées dans la littérature permettent de représenter des notions graduelles au sein d'un graphe RDF. Dans ce papier, nous considérons le modèle de données de la définition 1 qui synthétise les modèles RDF flous de la littérature dont le principe commun consiste à ajouter un degré flou dans $[0, 1]$ à chaque triplet RDF. Un degré attaché à un triplet $\langle \mathbf{s}, \mathbf{p}, \mathbf{o} \rangle$ exprime à quel point l'objet **o** satisfait la propriété **p** sur le sujet **s**. Par exemple, le triplet flou $\langle$`Beyonce, recommande, Euphoria`$\rangle$ auquel est attaché le degré 0.8 indique que $\langle$`Beyonce, recommande, Euphoria`$\rangle$ est satisfait au niveau 0.8, ce qui peut être interprété comme « `Beyonce` recommande *fortement* `Euphoria` ».

DÉFINITION 1. — *Un graphe RDF flou, noté graphe F-RDF, est un couple $(\mathcal{T}, \zeta)$ tel que (i) $\mathcal{T}$ est un ensemble fini de triplets de $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$, (ii) ζ est une fonction d'appartenance sur les triplets $\zeta : \mathcal{T} \to [0, 1]$.*

La fonction $\zeta(t)$ représente l'intensité de la relation portée par t. Intuitivement, ζ associe des degrés dans $[0, 1]$ aux arcs du graphe. $\zeta(t) = 0$ signifie que t n'appartient pas au graphe. $\zeta(t) = 1$ signifie que t est totalement satisfait, ce qui correspond à la notion classique de triplet non flou (dans le graphe G_{MB} de la figure 1, ce type d'arc apparaît sans degré associé).

Les degrés peuvent être donnés ou calculés, matérialisés ou non. Dans sa forme la plus simple, un degré peut correspondre au calcul d'une notion statistique reflétant l'intensité de la relation à laquelle le degré est attaché. Par exemple, l'intensité d'une relation d'amitié d'une personne p_1 vers une autre personne p_2 peut être calculée par la proportion d'amis communs par rapport au nombre total d'amis de p_1.

REMARQUE 2. — Un graphe RDF classique, non flou, est un cas particulier de graphe F-RDF pour lequel le co-domaine de ζ est $\{0, 1\}$. Ansi, les concepts et le langage d'interrogation flexible FURQL définis dans la suite, sont applicables (et tout à fait pertinents) dans le cadre des graphes RDF classiques. □

EXEMPLE 3 (Graphe RDF flou). — La figure 1 est un exemple de graphe F-RDF inspiré de MusicBrainz [1]. Ce graphe, noté G_{MB} dans la suite, est utilisé tout au long de l'article afin d'illustrer les notions introduites. Ses nœuds représentent des artistes (des musiciens ici) et des albums. Pour des raisons de lisibilité, chaque URI est remplacé par une valeur correspondant au nom du nœud plutôt qu'à l'URI lui-même, peu lisible. Pour les mêmes raisons, nous omettons les préfixes de façon à alléger la forme des données manipulées. Des valeurs littérales peuvent être attachées aux artistes et albums, comme l'âge de l'artiste, la date de sortie et la note (évaluation) globale d'un album. Le graphe contient à la fois des relations floues (p.e. ami, aime, recommande, membreDe) et des relations non floues (p.e. créateur, date). Les relations floues n'existent pas dans la base MusicBrainz initiale. Elles ont été ajoutées de façon à l'enrichir. Les degrés flous associés aux relations, reflétant l'intensité de celles-ci, sont obtenus par des calculs statistiques simples. Par exemple, le degré associé à un arc de la forme Art − membreDe → Group correspond à la proportion d'années pendant lesquelles l'artiste a été membre du groupe au regard du nombre d'années d'existence du groupe. Il est évidemment possible de considérer des degrés flous issus de calculs beaucoup plus complexes ou définis par expertise. □

On s'appuie dans la suite sur des notions classiques de la théorie des graphes flous (Rosenfeid, 2014) : le *chemin*, la *distance* et la *force* de la connexion entre deux nœuds.

Soit G un graphe F-RDF, fixé pour la suite. Classiquement, un *chemin* p dans G est décrit par une liste éventuellement vide de triplets de la forme $(t_1, \cdots, t_k, \cdots, t_n)$ où $\{t_i \mid 1 \leq i \leq n\} \subseteq G$ et pour tout $1 \leq k \leq n-1$, l'objet de t_k est le sujet de t_{k+1}. Étant donnés deux nœuds x et y, $Chemins(x, y)$ représente l'ensemble des chemins sans cycle [2] menant de x à y dans G c'est-à-dire l'ensemble des chemins de la forme $(t_1, \cdots, t_k, \cdots, t_n)$ telle que x est le sujet de t_1 et y est l'objet de t_n. Étant donnés deux nœuds x et y, la *distance* associée au couple de nœud (x,y) est définie par $Distance(x, y) = min_{p \in Chemins(x, y)}\ Longueur(p)$ où $Longueur(p)$ est la Longueur du chemin p dans le graphe flou (Rosenfeid, 2014), calculé par $Longueur(p) = \sum_{t \in p} \frac{1}{\zeta(t)}$. Étant donnés deux nœuds x et y, la *force* associée au couple de nœud (x,y)

1. https://musicbrainz.org/
2. Considérer les chemins avec cycle ne changerait pas le résultat des expressions de *Distance* et *ST*.

est définie par $ST(x,\, y) = max_{p \in Chemins(x,\, y)}\, ST\_chemin(p)$ où $ST\_chemin(p)$ est la force du chemin p dans le graphe flou : $ST\_chemin(p) = \min(\{\zeta(t) | t \in p\})$.

Le langage SPARQL. SPARQL (Prud'hommeaux, Seaborne, 2008) est le langage de requête standard recommandé par W3C pour interroger des données RDF. Il s'agit d'un langage déclaratif fondé sur la recherche de patrons de graphes, dans le sens où le moteur de requête recherche les ensembles de triplets du graphe de données satisfaisant un patron défini dans la requête. Pour présenter les choses simplement, dans sa forme basique, un patron de graphe est un ensemble de triplets contenant des variables et composés à l'aide des opérateurs `UNION`, `FILTER`, `OPTIONAL` et `.` (concaténation). La requête 1 est un exemple de requête SPARQL visant à récupérer les albums de 2012 créés ou aimés par Shakira, avec la note associée si celle-ci est disponible.

```
SELECT ?album ?r WHERE {
  { ?artist name "Shakira". ?artist créateur ?album. }
  UNION { ?artist name "Shakira". ?artist aime ?album. }
  OPTIONAL { ?album note ?r. }
  ?album date ?d.
  FILTER (?d = "2012") }
```

Requête 1 – Exemple de requête SPARQL

Dans la suite, nous définissons le concept de *patron flou de graphe* permettant d'exprimer des préférences floues sur les données d'un graphe flou F-RDF (*via* des conditions floues) et sur sa structure (*via* des expressions régulières floues).

3. Les patrons flous de graphe

La notion de *patron flou de graphe* que nous introduisons ci-dessous repose sur celle de *patron de graphe* introduite dans (Pérez *et al*., 2009).

Avant de donner la définition d'un patron flou de graphe, il est nécessaire de définir la notion d'*expression régulière floue*. Dans la suite, on pose l'existence d'un ensemble infini $\mathcal{V}$ de variables tel que $\mathcal{V} \cap (\mathcal{U} \cup \mathcal{L}) = \emptyset$. Par convention, un élément de $\mathcal{V}$ est préfixé par un point d'interrogation.

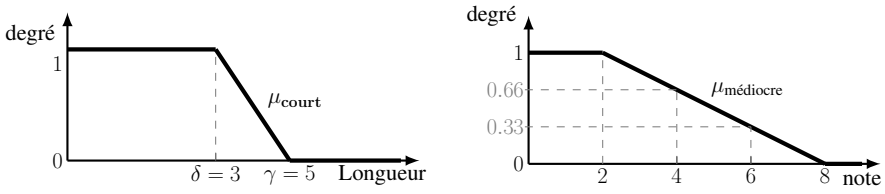
DÉFINITION 4. — *L'ensemble $\mathcal{F}$ des expressions régulières floues est défini à partir de l'ensemble $\mathcal{U}$ des URI de façon récursive comme suit :*

– *ϵ est une expression régulière floue de $\mathcal{F}$ représentant un chemin vide;*

– *$u \in \mathcal{U}$ et '_' sont des expressions régulières floues de $\mathcal{F}$;*

– *si $A \in \mathcal{F}$ et $B \in \mathcal{F}$ alors $A|B$, $A.B$, A^* et A^{cond} sont des expressions régulières floues de $\mathcal{F}$.*

Le caractère '_' représente un élément quelconque de $\mathcal{U}$, $A|B$ représente deux expressions alternatives, $A.B$ représente la concaténation d'expressions, A^* représente la répétition classique d'expression (clôture de Kleene), A^{cond} représente les chemins



(a) Représentation du terme flou *court* (b) Représentation du terme flou *médiocre*

FIGURE 2. *Représentation des termes flous*

satisfaisant l'expression A avec satisfaction de la condition $cond$ où $cond$ est la combinaison booléenne de formules atomiques de la forme : $sprop$ IS $Fterm$ *où $sprop$ est une propriété structurelle sur le chemin défini par A, et $Fterm$ est un terme flou (par exemple le terme* court *défini en figure 2.(a)).*

Dans la suite, on limite les propriétés structurelles de chemin à ST et $Distance$ (voir la section 2). Par ailleurs, on utilise le raccourci classique A^+ pour $A.A^*$.

DÉFINITION 5. — *L'ensemble des patrons flous de graphe est défini de façon récursive comme suit :*

– un triplet flou de $(\mathcal{U} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{F} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{V})$ est un patron flou de graphe;

– si P_1 et P_2 sont des patrons flous de graphe alors (P_1 ET *P_2), (P_1* UNION *P_2) et (P_1* OPT *P_2) sont des patrons flous de graphe;*

– si P est un patron flou de graphe et C est une condition floue alors (P FILTER *C) est un patron flou de graphe. Une condition floue est une combinaison logique impliquant des termes flous définie par : (i) si $\{?x, ?y\} \subseteq \mathcal{V}$ et $c \in (\mathcal{U} \cup \mathcal{L})$, alors $bound(?x)$, $?x\,\theta\,c$ et $?x\,\theta\,?y$ sont des conditions floues, où θ est un comparateur flou ou non flou, (ii) si $?x \in \mathcal{V}$ et $Fterm$ est un terme flou alors $?x$* IS *$Fterm$ est une condition floue, (iii) si C_1 et C_2 sont des conditions floues alors $(\neg C_1)$ et $(C_1 \odot C_2)$ (où $\odot$ est un connecteur flou) sont des conditions floues. Un connecteur flou peut être, dans sa forme la plus simple, une conjonction floue $\wedge$ (resp. disjonction floue $\vee$), généralement interprétée par la norme triangulaire* minimum *(resp.* maximum*).*

L'article (Pivert *et al*., 2017) contient la définition formelle de la sémantique associées à un patron flou de graphe (c.-à-d. permettant d'interpréter des préférences utilisateur), relativement à un graphe de données flou (c.-à-d. contenant éventuellement des données graduelles). Par manque de place, nous ne donnons ici qu'une intuition de cette sémantique, appuyée par un exemple.

Intuitivement, étant donné un graphe G de données F-RDF, la sémantique associée à un patron flou de graphe P définit l'ensemble des correspondances possibles du patron P dans un sous-graphe de G isomorphe (autrement dit, si P est une requête, la sémantique définit l'ensemble de ses réponses trouvées dans G). Une correspondance

prend la forme d'un mapping associant chaque variable du patron à un élément de G (sémantique jusqu'ici classique de patron de graphe, voir (Pérez *et al.*, 2009)). La sémantique que nous proposons permet de prendre en compte les préférences utilisateur introduites dans l'extension floue de patron de graphe, appliquées à un graphe de données flou. En nous fondant sur la théorie des ensembles flous, les préférences sont exploitées afin d'associer un degré de satisfaction à chaque sous-graphe réponse, qualifiant le degré d'adéquation de cette réponse à la requête.

EXEMPLE 6. — On considère le patron flou de graphe $P_{\text{rec\_médiocre}}$ suivant : *(?Art1,* $(ami^+)^{Distance \text{ IS } court}$*.créateur, ?Alb)* AND *(?Art1, recommande, ?Alb)* AND *((?Alb, note, ?r)* FILTER *(?r is médiocre))* illustré dans la figure 3.

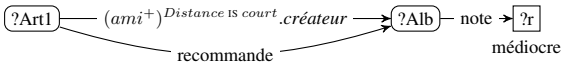


FIGURE *3. Représentation du patron* $P_{rec\_médiocre}$

Intuitivement, $P_{\text{rec\_médiocre}}$ récupère les artistes *(?Art1)* dans G_{MB}, tels que *(?Art1)* recommande un album mal noté *(?Alb)* créé par un autre artiste qui est un ami proche de *(?Art1)*. La figure 4 donne l'ensemble des sous-graphes de G_{MB} satisfaisant le patron $P_{\text{rec\_médiocre}}$. Deux mappings sont exhibés : $m_1 = \{?\text{Art1} \rightarrow \text{EnriqueI}, ?\text{Alb} \rightarrow \text{Justified}, ?\text{r} \rightarrow 6\}$ ayant permis la mise en correspondance de $P_{\text{rec\_médiocre}}$ avec le sous-graphe g_1 et $m_2 = \{?\text{Art1} \rightarrow \text{Shakira} , ?\text{Alb} \rightarrow \text{Butterfly}, ?\text{r} \rightarrow 4\})\}$ ayant permis la mise en correspondance de $P_{\text{rec\_médiocre}}$ avec le sous-graphe g_2.

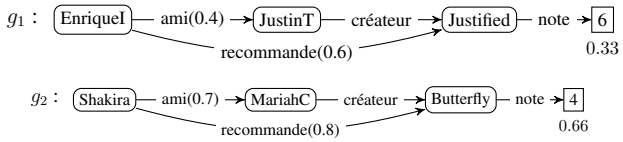


FIGURE *4. Sous-graphes de* G_{MB} *satisfaisant* $P_{rec\_médiocre}$

Le cadre que nous avons introduit associe les réponses suivantes à $P_{\text{rec\_médiocre}}$ interprété sur le graphe G_{MB} de la figure 1.

$$[P_{rec\_médiocre}]_{G_{MB}} = \{(m_1, 0.33), (m_2, 0.66)\}$$

En d'autres termes, le sous-graphe (flou) réponse g_2 peut être qualifié de relativement proche de la requête utilisateur (au regard des préférences de cet utilisateur), alors que le sous-graphe g_1 l'est moins. Ce résultat s'explique par un degré d'amitié assez fort exprimé dans g_2 entre *Shakira* et *MariahC*, ainsi que par une note relativement basse affectée à l'album *Butterfly* (ce qui était bien recherché par l'utilisateur). Le détail de ce calcul se trouve dans (Pivert *et al.*, 2017). □

4. Le langage de requête FURQL et son implantation

Nous introduisons maintenant le langage de requête FURQL (*FUzzy Rdf Query Language*), qui consiste à étendre les patrons de graphe de SPARQL par des patrons flous de graphe définis dans la section précédente. Seule la syntaxe du langage est présentée ici (sa sémantique découle trivialement de celle de SPARQL et de l'extension de ses patrons).

L'extension FURQL permet l'occurrence de patrons flous de graphe dans la clause `WHERE` et l'occurrence de conditions floues dans la clause `FILTER`. La syntaxe d'une expression floue de graphe est proche de celle de chemin, comme défini dans SPARQL 1.1 (Harris, Seaborne, 2013), permettant d'exhiber des nœuds reliés par des chemins exprimés sous forme d'une expression régulière. On permet ici l'expression d'une propriété floue portant sur les nœuds reliés (une propriété d'un chemin concerne par exemple la *distance* ou la *force* de connexion des nœuds).

EXEMPLE 7. — La requête 2 en language FURQL a pour objectif de récupérer les artistes qui recommandent un album d'un ami proche, l'album étant mal noté. Le patron utilisé dans cette requête est le patron $P_{\text{rec\_médiocre}}$ de l'exemple 6. La requête effectue de plus une alpha-coupe des réponses obtenues par l'utilisation de la clause `CUT`. Ici, seules les réponses ayant un degré de satisfaction supérieur ou égal à 0.4 sont retournées. L'utilisation de la clause `CUT` est optionnelle.

```
SELECT ?art1 WHERE {
 { ?art1 (ami+ | Distance IS court) ?art2. ?art2 créateur ?alb.
   ?art1 recommande ?alb. ?alb note ?r. }
FILTER (?r IS médiocre) } CUT 0.4
```

Requête 2 – Requête FURQL contenant $P_{\text{rec\_médiocre}}$

Le résultat de l'exécution de cette requête sur G_{MB} est le singleton {`Shakira`} correspondant à $m(?art1)$ dans le mapping résultat {?art1 $\rightarrow$ Shakira , ?alb $\rightarrow$ Butterfly, ?r $\rightarrow$ 4}. Il s'agit du seul mapping de $[P_{\text{rec\_médiocre}}]_{G_{MB}}$ ayant un degré de satisfaction supérieur ou égal à 0.4 (voir l'exemple 6). $\diamond$ □

Cette extension est implantée sous la forme d'un prototype nommé SURF téléchargeable à l'adresse https://www-shaman.irisa.fr/surf/. SURF prend la forme d'une surcouche logicielle permettant la prise en compte de requêtes FURQL, que l'on associe à un moteur SPARQL standard (et donc éventuellement *endpoint* distant). Cette couche logicielle est composée de deux modules : (Module 1) un *compilateur de requête* FURQL produisant une requête ou un ensemble de requêtes SPARQL dont l'objectif est de récupérer les données utiles à l'évaluation de la requête FURQL, composées non seulement des sous-graphes satisfaisant la requête mais également des informations permettant le calcul des degrés de satisfaction associés aux réponses, et (Module 2) un module de *traitement des données floues* qui calcule les degrés de satisfaction des réponses à partir des données récupérées, puis trie les réponses.

5. Travaux connexes

Les travaux connexes concernent deux familles d'approches qui étendent SPARQL par i) des capacités *flexibles* de navigation dans les chemins d'un graphe RDF, et ii) des capacités d'interrogation flexible à base de préférences.

Dans la première catégorie de travaux, des langages de requêtes (Kochut, Janik, 2007 ; Anyanwu *et al*., 2007 ; Pérez *et al*., 2010 ; Alkhateeb *et al*., 2009) ont été définis de façon à lever les limitations de SPARQL en termes d'expression de chemin. Ces langages ont permis l'extension de SPARQL par l'utilisation d'expressions régulières décrivant des chemins (appelées expressions régulières de chemin dans la suite), utilisées dans les patrons de requête. Nous ne rentrons pas dans le détail de ces travaux car les expressions régulières de chemins ont été introduites dans la norme SPARQL à partir de sa version 1.1 (Harris, Seaborne, 2013).

La seconde catégorie de travaux connexes concerne les extensions de SPARQL à base de préférences. Dans (Pivert *et al*., 2016), nous dressons un état de l'art de ces approches, qui peuvent être classifiées en deux catégories : les approches quantitatives fondées sur la théorie des ensembles flous ou sur les requêtes top-k, et les approches qualitatives fondées sur le mécanisme des requêtes Skyline. Grossièrement, les approches quantitatives permettent de dégager un ordre total des réponses (les approches top-k limitant de plus le nombre de réponses rapatriées), alors que les approches qualitatives en dégagent un ordre partiel.

Notre contribution se situe dans la sous-catégorie des approches quantitatives reposant sur l'introduction de préférences fondées sur la théorie des ensembles flous. Les travaux de la littérature de cette catégorie, c'est-à-dire (Cheng *et al*., 2010), (Wang *et al*., 2012) et (Ma *et al*., 2015) permettent d'exprimer des préférences concernant les valeurs (littérales principalement) portées par les données mais *pas* concernant la structure du graphe RDF. À notre connaissance, cette limitation concerne d'ailleurs également les autres types d'approches d'interrogation flexible proposées, qu'elles soient quantitatives ou qualitatives (voir (Pivert *et al*., 2016) pour plus de détails).

6. Conclusion

Dans cet article, nous avons proposé une extension du langage SPARQL permettant l'expression de préférences fondées sur la théorie des ensembles floues. Cette extension repose sur la définition de la notion de patron flou de graphe étendant la notion de patron de graphe SPARQL. L'extension prend la forme d'un language nommé FURQL, plus expressif que toutes les propositions existantes de la literature, dont les principales caractéristiques sont les suivantes : i) FURQL considère le modèle de données *RDF flou* permettant d'exprimer des relations graduelles (dont le modèle RDF non flou est un cas particulier); i) FURQL permet l'expression de conditions floues portant non seulement sur les valeurs des données du graphe mais aussi sur sa structure. Enfin, nous avons abordé brièvement l'implantation de FURQL.

Il existe de nombreuses perspectives à ce travail. Nous envisageons d'étendre FURQL avec des préférences plus sophistiquées dont certaines font appel à des notions provenant du domaine de l'analyse des réseaux sociaux (centralité ou prestige d'un noeud) ou de la théorie des graphes (par exemple, clique, etc). Il vaut également la peine d'étudier la manière dont notre cadre pourrait être appliqué à la gestion de dimensions de qualité des données (par exemple, précision, cohérence, etc.) qui sont en général d'une nature graduelle.

Remerciements: Ce travail a été partiellement financé par la DGE (Direction Générale des Entreprises), *via* le projet ODIN (Open Data INtelligence).

Bibliographie

Alkhateeb F., Baget J.-F., Euzenat J. (2009). Extending SPARQL with regular expression patterns (for querying RDF). *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, n° 2, p. 57–73.

Anyanwu K., Maduko A., Sheth A. (2007). SPARQ2L: towards support for subgraph extraction queries in RDF databases. In *Proc. of the 16th international conference on world wide web*, p. 797–806.

Cheng J., Ma Z., Yan L. (2010). f-SPARQL: a flexible extension of SPARQL. In *Proc. of the intl. conf. on database and expert systems applications*, p. 487–494.

Harris S., Seaborne A. (2013). *SPARQL 1.1 query language.* W3C Recommendation http://www.w3.org/TR/sparql11-query.

Kochut K. J., Janik M. (2007). SPARQLer: Extended SPARQL for semantic association discovery. In *Proc. of the 4th european semantic web conference (ESWC'07)*, p. 145–159.

Ma R., Jia X., Cheng J., Angryk R. (2015). SPARQL queries on RDF with fuzzy constraints and preferences. *Journal of Intelligent & Fuzzy Systems*, vol. 202, p. 1-13.

Pérez J., Arenas M., Gutierrez C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)*, vol. 34, n° 3, p. 16:1–16:45.

Pérez J., Arenas M., Gutierrez C. (2010). nSPARQL: A navigational language for RDF. *Journal of Web Semantics*, vol. 8, n° 4, p. 255–270.

Pivert O., Slama O., Thion V. (2016). SPARQL extensions with preferences: a survey. In *Proc. of the 31st annual acm symposium on applied computing*, p. 1015–1020.

Pivert O., Slama O., Thion V. (2017, juillet). Fuzzy Quantified Queries to Fuzzy RDF Databases. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* Naples, Italy.

Prud'hommeaux E., Seaborne A. (2008). *SPARQL query language for RDF.* W3C recommendation. http://www.w3.org/TR/rdf-sparql-query/.

Rosenfeid A. (2014). Fuzzy graphs. In *Proc. of the us–japan seminar on fuzzy sets and their applications*, p. 77.

W3C. (2014). *RDF overview and documentations.* (http://www.w3.org/RDF/)

Wang H., Ma Z., Cheng J. (2012). fp-Sparql: an RDF fuzzy retrieval mechanism supporting user preference. In *Proc. of FSKD*, p. 443–447.

Interrogation de données hétérogènes dans les systèmes NoSQL orientés graphes

Mohammed El Malki[2], Hamdi Ben Hamadou[1], Max Chevalier[1], André Péninou[2], Olivier Teste[2]

1. Université Toulouse 3 Paul Sabatier, IRIT (CNRS/UMR5505)
prenom.nom@irit.fr

2. Université Toulouse 2 Jean Jaurès, UT2C, IRIT (CNRS/UMR5505)
prenom.nom@irit.fr

RESUME. La flexibilité des systèmes NoSQL, qui consiste à ne plus garantir un schéma unique pour un ensemble de données, aboutit à des masses de données hétérogènes rendant leur interrogation plus complexe pour les utilisateurs, qui doivent connaître les différentes formes (c'est-à-dire les différents schémas) des données manipulées. Cet article se focalise sur cette problématique de l'interrogation des données hétérogènes dans les systèmes NoSQL orientés graphes. L'enjeu est de simplifier pour les utilisateurs l'interrogation de ces masses de données hétérogènes en rendant transparente leur hétérogénéité. L'article propose de construire un dictionnaire de similarité entre labels et attributs. A partir de ce dictionnaire la requête utilisateur peut être automatiquement réécrite pour intégrer la variabilité des données.

ABSTRACT. The NoSQL systems falls within the "schemaless" principle consisting in providing more than a single schema for a dataset, thus allowing a wide variety of representations. This flexibility leads to a large volume of heterogeneous data, wish makes their interrogation more complex for the users, who are compelled to know the different forms (i.e. the different schemas) of this data. This paper addresses this issue and focus on simplifying for users the heterogeneous data interrogation process in graph-oriented NoSQL systems. The paper proposes to build a similarities dictionary between labels and attributes. From this dictionary, the user query is automatically extended to integrate the variability of underlying data.

Mots-clés : NoSQL, similarité, flexibilité des schémas

KEYWORDS: NOSQL, NEO4J, SCHEMALESS, SIMILARITY

1. Introduction

En raison de leur capacité à gérer efficacement d'importantes masses de données, les systèmes de stockage « not-only-SQL » ou NoSQL, connaissent un important développement (Floratou et al., 2005) (Stonebraker, 2010). Parmi les différentes approches NoSQL, les systèmes orientés graphes permettent de modéliser les données sous la forme de graphes (Holzschuher and Peinl, 2013). Les données sont représentées sous la forme de nœuds, de relations et de propriétés (Roussy 2016), permettant ainsi de modéliser les différentes interactions entre les données. Ce type de représentation joue un rôle central dans de nombreux domaines tels que les réseaux sociaux, le Web sémantique, les sciences du vivant (interactions de protéines).

Les systèmes NoSQL, et donc les systèmes orientés graphes, caractérisés par le principe de « *schemaless* » (Cattel, 2010) ne garantissent plus un schéma unique pour un ensemble de données. Ainsi chaque nœud et chaque relation possède son propre ensemble de propriétés, permettant ainsi une grande variété de représentation (Chevalier et al., 2015). Cette flexibilité aboutit à des masses de données hétérogènes (schémas différents), rendant leur interrogation plus complexe pour les utilisateurs, qui doivent connaître les différentes formes (c'est-à-dire les différents schémas) des données manipulées. Cet article se focalise sur cette problématique de l'interrogation des données hétérogènes dans les systèmes NoSQL orientés graphes. L'enjeu est de simplifier pour les utilisateurs l'interrogation de ces masses de données hétérogènes en rendant transparente leur hétérogénéité. Cet article propose de construire un dictionnaire de similarité entre labels et attributs. A partir de ce dictionnaire la requête utilisateur peut être automatiquement réécrite pour intégrer la variabilité des données réelles.

Le reste du document est structuré comme suit. La section 2 illustre le problème, la section 3 traite l'état de l'art. Nous présentons notre solution d'interrogation des données hétérogènes, appelé *EasyGraphQuery*, dans la section 4. Les premiers résultats de l'évaluation expérimentale sont présentés dans le section 5.

2. Illustration du problème

2.1. Notations préliminaires

La modélisation des données dans les systèmes NoSQL orientés graphes consiste à considérer la base de données comme un graphe. La Figure 1 illustre un exemple simple de graphe $G = (V, E, \gamma)$ où $V = \{u_1,\ldots u_{13}\}$ représente les nœuds, $E = \{e_1,\ldots e_9\}$ représente les arêtes et $\gamma : E \rightarrow V \times V$ est une fonction déterminant les paires de nœuds reliés par les arêtes (appelés aussi relations).

Les différents nœuds peuvent être décrits sous une forme textuelle comme ci-dessous Figure 1. Chaque nœud comporte un ou plusieurs labels (ex. *Author : Poet*) et peut être caractérisé par des attributs (ex. *Firstname*). On constate que ce graphe comporte des éléments (nœuds et attributs) hétérogènes.

- u_1:Author{firstname:'Charles',lastname:'Baudelaire',birth_date:1821,date_of_death:1867}
- u_2:Writer{firstname:'Paul',lastname:'Verlaine',birth:1844}
- u_3:Author:Poet{firstname:'Guillaume',lastname:'Apollinaire',birth:1880,death:1918}
- u_4:author{firstname:'Arthur',lastname:'Rimbaud',birth_date:1854,death_of_death:1891}
- u_5:Book{number:1,title:'Les fleurs du mal',year:1857}
- u_6:Book{number:2,title:'Le Spleen de Paris',year:1869}
- u_7:book{number:3,title:'Les Paradis artificiels',year:1860}
- u_8:Work{number:4,title:'Poèmes saturniens',year:1866}
- u_9:Work{number:5,title:'Fêtes galantes'}
- u_{10}:Publication{number:5,title:'Alcools'}
- u_{11}:Book{number:6,title:'Une saison en enfer',year:1873}
- u_{12}:Work{number:7,title:'Les illuminations',year:1886}
- u_{13}:Publication{number:8,title:'Le Bateau ivre',year:1920}

Sur le graphe de la Figure 1, on peut remarquer que pour une sémantique probablement équivalente le nom d'une relation peut varier (soit `To_Write`, soit `To_Compose`) ; comme les nœuds du graphe, les arêtes (labels relations) et leurs attributs peuvent être hétérogènes.

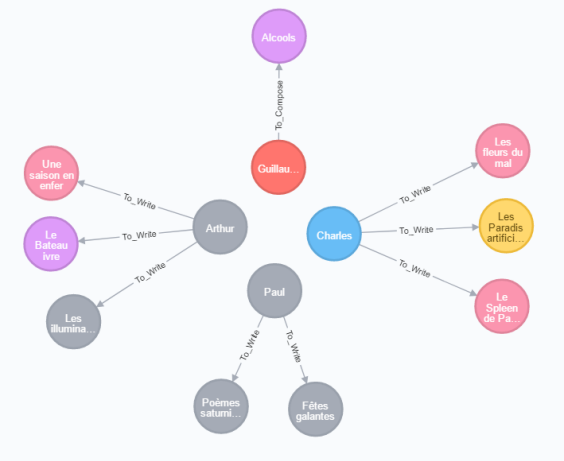


Figure 1 : Exemple de graphe.

2.2. Facettes de l'hétérogénéité

L'hétérogénéité peut être considérée selon différentes facettes (Shvaiko et Euzenat, 2005) suivant les éléments de structures qui constituent un graphe (attributs) mais aussi labels des nœuds ou des relations, ainsi que les extrémités reliées). La première facette, l'hétérogénéité structurelle, désigne le problème qu'une donnée peut être représentée par des éléments de structure variables. La seconde facette, l'hétérogénéité syntaxique, désigne le problème qu'un élément de structure peut être désigné de manière variable ; *par exemple, les attributs 'birth_date' et 'birth' dans*

les nœuds désignent toutes les deux une date de naissance d'un auteur. La troisième facette, l'hétérogénéité sémantique, désigne le problème que deux éléments différents peuvent correspondre à une même donnée, ou inversement qu'un élément peut correspondre à des données variables ; *par exemple, les relations 'To_Write' et 'To_Compose' ont le même sens.*

Dans cet article nous étudions ces différentes facettes de l'hétérogénéité. Néanmoins, nous ne traitons pas l'hétérogénéité d'entités (Getoor et Machanavajjhala, 2012) en considérant que pour chaque entité conceptuelle, lui correspond un composant élémentaire du graphe, c'est-à-dire, un nœud ou une relation ; nous ne considérons pas la possibilité qu'une entité corresponde à un sous-graphe.

2.3. Problématique de l'interrogation de graphes NoSQL hétérogènes

Nous utilisons le système de stockage *Neo4j* pour illustrer notre étude de cas. Ce système propose le langage propriétaire *Cypher* (Holzschuher and Peinl, 2013) permettant d'exprimer des requêtes manipulant la base de données orientée graphe. Nous utiliserons ce langage pour illustrer nos propos. Nous limitons notre étude aux opérateurs de projection et de sélection (ou restriction).

Exemple. Considérons, toujours en utilisant le graphe de la Figure 1, une requête exemple pour chacun des opérateurs.

Projection. « `Obtenir le nom, le prénom et l'intitulé des œuvres littéraires des auteurs` »

```
match (n:Author)-[]-(m)
return n.firstname, n.lastname, m.title
```

On obtient alors le résultat ci-dessous ne faisant apparaitre qu'une partie des auteurs et des ouvrages réalisés par les auteurs. Ce résultat incomplet est dû à l'hétérogénéité du graphe. Ainsi l'auteur `Paul Verlaine` n'apparaît pas car le nœud n'est pas de type (on parle de label) `Author` mais `Writer`.

firstname	lastname	title
Charles	Baudelaire	Les Paradis artificiels
Charles	Baudelaire	Le Spleen de Paris
Charles	Baudelaire	Les fleurs du mal
Guillaume	Apollinaire	Alcools

Projection et Sélection. « `Obtenir le titres des œuvres littéraires de l'auteur Baudelaire` »

```
match (n:Author)-[]-(m:Book)
where lastname = 'Baudelaire' return m.title
```

On obtient encore un résultat incomplet, du point de vue de l'utilisateur, encore dû à l'hétérogénéité des données présentent dans le graphe. Le problème dans ce cas est dû à l'hétérogénéité des labels associés aux nœuds qui sont étiquetés soit `Book` soit `book`. Ce dernier n'est pas reconnu.

title
Les fleurs du mal
Le Spleen de Paris

On constate donc qu'une utilisation classique de Cypher dans un contexte de graphe hétérogène peut conduire l'utilisateur à bâtir des analyses et des décisions sur des données incomplètes.

Nous proposons dans cet article une approche permettant à un utilisateur d'exprimer simplement une requête à partir des attributs, sans avoir à tenir compte des différences structurelles, syntaxiques et sémantiques, tout en conservant les structures originelles des graphes. La requête permet d'obtenir un résultat « complet », de manière transparente par rapport à l'hétérogénéité des données (sans avoir à connaître et à manipuler avec exhaustivité les différentes facettes de l'hétérogénéité présente).

3. Etat de l'art

Dans cette section, même si ce travail peut être assimilée à une forme d'alignement nous abordons uniquement les deux approches principales d'interrogation utilisées et appliquées d'une manière générale dans les systèmes NoSQL.

L'approche d'homogénéisation dite *pivot* consiste à modifier la structure lors du stockage et à interroger les données dans un schéma pivot homogène. Par exemple, (Tahara et al., 2014) proposent le système *Sinew* qui aplatit les données et les charge dans un SGBD relationnel (tables). (Beyer et al., 2011) propose un nouveau langage de script pour interroger simultanément des données stockées dans des magasins différents et les requêtes sont découpées pour être distribuées en se basant sur le paradigme *Mapreduce* (Thusoo et al., 2009). Au-delà des coûts d'évaluation des requêtes, l'utilisateur doit connaitre les structures ou les métadonnées de l'ensemble des structures pour les interroger correctement. Cette approche d'homogénéisation a pour avantage de faciliter l'interrogation pour l'utilisateur qui manipule ainsi un schéma unique mais nécessite des pré-traitements pouvant s'avérer coûteux et difficilement compatibles avec des environnements dynamiques.

La seconde approche considérée dans la littérature consiste à inférer les différents schémas pour permettre leur interrogation. Dans ce contexte, des travaux s'attaquent à la problématique d'intégration et la découverte de nouveaux schémas (Wang et al., 2015) et (Herrero et al., 2016). Les travaux de (Herrero et al., 2016) proposent d'extraire séparément tous les schémas présents afin d'aider l'utilisateur à connaitre tous les schémas et tous les attributs présents (Wang et al., 2015), (Hamdi et al., 2018). Dans (Wang et al., 2015) les auteurs proposent d'homogénéiser tous les schémas dans un même « schéma universel » (skeleton) afin d'aider l'utilisateur dans la découverte d'attributs ou de sous-schémas. Comme l'approche précédente, l'intégration de nouveaux schémas nécessite de revoir tous les autres schémas déjà stockés ce qui va à l'encontre de la technologie NoSQL dont la vélocité est une des caractéristiques principales ; De plus l'hétérogénéité doit toujours être gérée par l'utilisateur lors des requêtes.

4. Le système EasyGraphQuery

Le principe que nous retenons est de gérer la variabilité (ou au moins une partie de celle-ci) automatiquement. On suppose que l'utilsateur pose une requête en ne connaissant qu'un schéma de données ; par exemple *Author, Birthdate, To_write*. La requête est ensuite réécrite et étendue pour intégrer la variabilité des données réelles. Pour cela, un dictionnaire permet d'associer à chaque élément du schéma (labels et attributs) une liste d'éléments « équivalents » que nous calculons par des mesures de similarité.

Le calcul de la similarité est effectué entre les éléments du graphe susceptibles d'être hétérogènes. Nous prenons en compte dans cet article les labels et les attributs des nœuds et des arêtes, pris séparément. Les facettes de l'hétérogénéité prises en compte sont l'hétérogénéité structurelle, l'hétérogénéité syntaxique et l'hétérogénéité sémantique. Ainsi, deux matrices sont construites afin de déterminer les similarités entre éléments du graphe : la matrice de similarité syntaxique est basée sur la mesure *Leivenshtein* (**Erreur ! Source du renvoi introuvable.**a) tandis que la matrice de similarité sémantique se base sur la mesure Lin (**Erreur ! Source du renvoi introuvable.**b). Nous ne détaillons pas également les pré-traitements appliqués lors de multi-termes comme pour `To_Write` ou `birth_date` lors des calculs des matrices. On peut envisager d'étendre l'approche avec d'autres mesures de similarités, et d'améliorer le processus par une combinaison pondérée de ces diverses mesures (Shvaiko et Euzenat, 2005) (Megdiche, et al., 2016).

Exemple. Considérons le graphe de la Figure 1. Le label `Author` du nœud u_1 (première ligne des matrices) est comparé avec les différents labels des nœuds du graphe. Pour déterminer les labels similaires ∇_{Author} nous utilisons $\forall j \in [1,N]$, $max\ (Leivenshtein\ (u_1, u_j), Lin(u_1, u_j)) \geq 0.8$[1] ; ∇_{Author} = { `Author`, `author`, `Writer` }.

De manière analogue, le label `To_Write` de l'arête est comparé avec les autres labels des arêtes. On obtient alors dans notre cas ∇_{To_Write} = { `To_Write`, `To_Compose` }.

Le processus de calcul des similarités est également appliqué sur les attributs des nœuds et des arêtes. En appliquant la même combinaison des mesures de similarité *Leivenshtein* et *Lin* contenues dans les matrices, nous construisons le dictionnaire des données des attributs ; $\Delta_{Author.birth\_date}$ = { `Author.birth_date`, `Writer.birth`, `Author.birth`, `author.birth` }. Par ailleurs, ces deux matrices de similarité servent à construire le dictionnaire de similarité, constitué d'une clé correspondant à un

[1] Le choix du seuil est arbitraire, la fonction du calcul du seuil ne fait pas l'objet de cet article.

attribut donné et de sa valeur correspondant à la liste des attributs similaires (syntaxique et sémantiques).

4.1. Modélisation des données et du dictionnaire

Dans ce qui suit nous formalisons les différentes définitions nécessaires à la modélisation des données et du dictionnaire.

Définition 1. Un graphe G est défini par (V, E, γ)
– $V = \{u_1,\ldots, u_N\}$ est l'ensemble des nœuds du graphe ;
– $E = \{e_1,\ldots, e_M\}$ est l'ensemble des arêtes du graphe ;
– $\gamma : E \rightarrow V$ x V est la fonction qui associe chaque arête aux sommets reliés.

On note $\mathcal{L} = \{l_1, \ldots, l_L\}$ un ensemble de termes désignant les différents labels de nœuds et de relations possibles.

Définition 2. Un nœud u_i est défini par (L_i, S_i)
– $L_i \subseteq \mathcal{L}$ est l'ensemble des labels caractérisant le nœud ;
– $S_i = \{a_{i,1},\ldots,a_{i,n_i}\}$ est le schéma du nœud, constitué par un ensemble d'attributs.

On note $S_V = \bigcup_{i=1}^{i=N} \left(\bigcup_{l_k \in L_i} \bigcup_{a_{i,j} \in S_i} l_k.a_{i,j} \right)$ le schéma des nœuds du graphe.

Définition 3. Une arête e_i est définie par $(l_i, S_i, u_{i,1}, u_{i,2})$
– $l_i \in \mathcal{L}$ est le label caractérisant l'arête ;
– $S_i = \{a_{i,1},\ldots,a_{i,n_i}\}$ est le schéma l'arête constitué par un ensemble d'attributs ;
– $u_{i,1}$ et $u_{i,2}$ sont les nœuds origine et cible reliés par l'arête ; $\gamma(e_i) = \{(u_{i,1}, u_{i,2})\}$.

On note $S_E = \bigcup_{i=1}^{i=M} \left(\bigcup_{a_{i,j} \in S_i} l_i.a_{i,j} \right)$ le schéma des arêtes du graphe. On note alors $S_G = S_N \cup S_M$ le schéma des attributs du graphe.

On remarque que $\mathcal{L} = \left(\bigcup_{i=1}^{i=N} L_i \right) \cup \left(\bigcup_{i=1}^{i=M} l_i \right)$.

Exemple. Considérons le graphe de la Figure 1.

– $V = \{ u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13} \}$;
– $E = \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 \}$;
– $\gamma = \{ (e_1,(u_1, u_5)), (e_2,(u_1, u_6)), (e_3,(u_1, u_7)), (e_4,(u_2, u_8)), (e_5,(u_2, u_9)), (e_6,(u_3, u_{10})), (e_7,(u_4, u_{11})), (e_8,(u_4, u_{12})), (e_9,(u_4, u_{13}))\}$.

Dans la Figure 1, on trouve le nœud u_1 = ({*Author*}, {*firstname*, *lastname*, *birth_date*, *date_of_death*}) et le nœud u_7 = ({*Book*}, {*number*, *title*, *year*}). De même, on trouve l'arête e_3 = (*Book*, {}, u_1, u_7).

Afin de prendre en compte les différentes facettes de l'hétérogénéité du graphe (structurelle, syntaxique et sémantique), nous introduisons un dictionnaire de données

permettant de déterminer pour chaque élément du graphe (label de nœud ou d'arête, attribut de nœud ou d'arête) les éléments similaires.

Définition 4. Le dictionnaire des données $dict_{label}$ est défini par

$$dict_{label} = \{ (l_i, \nabla_i) \}$$

– $l_i \in \mathcal{L}_G$ est un label du graphe ;

– $\nabla_i = \bigcup_{\forall l_j \in \mathcal{L}_G | sim(l_i,l_j) \geq \delta} l_j \subseteq \mathcal{L}_G$ est l'ensemble des labels similaires. La fonction de similarité, notée *sim*, calcule un score normalisé compris entre [0..1] exprimant le taux de ressemblance (plus l_i et l_j sont similaires, plus le score est proche de 1). $\delta \in [0..1]$ est le seuil à partir duquel les labels l_i et l_j sont considérés comme similaires.

Définition 5. Le dictionnaire des données $dict_{attribut}$ est défini par

$$dict_{attribut} = \{ (l_i.a_{i,k}, \Delta_{i,k}) \}$$

– $l_i.a_{i,k} \in S_G$ est un attribut du graphe ;

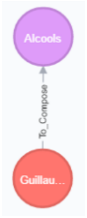
– $\Delta_{i,k} = \bigcup_{\forall l_j.a_{j,l} \in S_G | sim(a_{i,k},a_{j,l}) \geq \delta} l_j.a_{j,l} \subseteq S_G$ est l'ensemble des attributs similaires.

Remarque. Dans cet article nous ne prenons pas en compte l'hétérogénéité structurelle au niveau des labels mais uniquement entre les attributs ; cela signifie qu'un attribut peut être situé à diverses positions dans le graphe, repérées par les labels qui préfixent la propriété.

4.2. Noyau algébrique d'opérateurs

L'interrogation repose sur un ensemble d'opérateurs élémentaires formant un noyau minimum fermé. On note G_{in} le graphe interrogé et G_{out} le graphe résultat. Les graphes sont représentés sous une forme tabulaire.

Exemple. Considérons le sous graphe ci-dessous issu de la Figure 1 et sa forme tabulaire. Les labels et les références des arêtes aux nœuds reliés ne sont pas exprimés dans la forme tabulaire.



Graph Table
u3 : {firstname : 'Guillaume', lastname : 'Apollinaire', birth : 1880, death : 1918 }
u10 : {number : 5, title : 'Alcools' }
e6 : {year : 1913 }

Figure 2 : Représentation tabulaire des graphes.

Ces opérateurs élémentaires permettent d'exprimer des opérations de projection et de sélection (restriction).

Définition 6. La projection permet de réduire le graphe aux éléments de structure spécifiés dans le patron structurel (« pattern ») et la liste d'attributs projetés.

$$_{Pattern}\pi_{Attribute}(G_{in}) = G_{out}$$

– *Pattern* est un chemin de la forme l_0–l_1–…–l_p où chaque $l_i \in \mathcal{L}_G$ désigne la classe d'un nœud ou d'une arête.
– *Attribute* est un ensemble (possiblement vide) d'attributs $l_i.a_{i,k}$ où $l_i \in Pattern$ et $a_{i,k} \in S_i$.

Définition 7. L'opérateur de sélection permet de restreindre les éléments de structures aux seuls éléments satisfaisant un prédicat de sélection ; on note :

$$_{Pattern}\sigma_{Predicate}(G_{in}) = G_{out}$$

– *Pattern* est un chemin de la forme l_1–l_2–…–l_p où chaque $l_i \in \mathcal{L}_G$ désigne la classe d'un nœud ou d'une arête.
– *Predicate* est un prédicat (ou condition) de sélection. Un prédicat simple est une expression $l_i.a_{i,k}\ \omega_k\ v_k$ avec $a_{i,k} \subseteq S_i$ est un attribut, $\omega_k \in \{=;>;<;\neq;\geq;\leq\}$ est un opérateur de comparaison, et v_i une valeur. Les prédicats peuvent se combiner avec les opérandes $\Omega = \{\vee, \wedge, \neg\}$ formant un prédicat complexe.

Les prédicats de sélection complexe combinant plusieurs prédicats sont représentés sous sa forme normale conjonctive : $Predicate = \wedge_x(\vee_y p_{x,y})$.

Exemple. Considérons les requêtes de la section 2.3. Nous pouvons exprimer ces requêtes en représentation algébrique (interne) comme suit :

Projection. « `Obtenir le nom, le prénom et l'intitulé des œuvres littéraires des auteurs` »

$$_{\texttt{Author- -1}}\pi_{\texttt{Author.firstname,Author.lastname,1.title}}$$

Le résultat obtenu est donné dans le tableau ci-dessous. Lorsque les attributs sont projetés, les identifiants des nœuds (u_i) et des arêtes (e_i) sont perdus ; ceci rompt le principe de fermeture du noyau algébrique, ne permettant donc pas de combiner ce résultat avec une nouvelle opération.

Table 1 : Résultat de l'opération de projection.

G_{out}
{firstname : 'Charles', lastname : 'Baudelaire', title : 'Les Paradis artificiels' }
{firstname : 'Charles', lastname : 'Baudelaire', title : 'Les fleurs du mal' }
{firstname : 'Charles', lastname : 'Baudelaire', title : 'Le Spleen de Paris' }
{firstname : 'Guillaume', lastname : 'Apollinaire', title : 'Alcools' }

Projection et Sélection. « Obtenir le titres des œuvres littéraires de l'auteur Baudelaire »

$$\text{Author- -1}\pi_{\text{Author.firstname,Author.lastname,1.title}}(\text{Author- -1}\sigma_{\text{Author.lastname='Baudelaire'}})$$

Le résultat obtenu est donné dans le tableau suivant.

Table 2 : Résultat de la composition d'opérations de sélection et de projection.

G_{out}
{firstname : 'Charles', lastname : 'Baudelaire', title : 'Le Spleen de Paris' }
{firstname : 'Charles', lastname : 'Baudelaire', title : 'Les fleurs du mal' }

L'utilisation de cette représentation interne des opérations d'interrogation sur les graphes ne prend pas en charge l'hétérogénéité des éléments présents dans le graphe. Les résultats de ces requêtes restent donc incomplets au regard des informations présentent dans le graphe. Nous présentons dans la suite le processus de réécriture de ces requêtes internes permettant d'obtenir un résultat complet, de manière transparente pour l'utilisateur, et dynamique (sans transformation des données).

4.3. Algorithme de réécriture des requêtes

Notre approche consiste à faciliter l'interrogation pour les utilisateurs, par reformulation automatique des requêtes. Ce processus exploite le dictionnaire et indirectement les matrices de similarité des données afin de reformuler la requête en déterminant les éléments (nœuds, arêtes et attributs) similaires. L'algorithme suivant décrit ce processus de récriture automatique de la requête utilisateur.

La fonction *exists*(*A, L*) permet de vérifier l'existence dans L, du pattern constitué à partir des labels issus de A. L'opération d'union, notée $\cup$, permet d'unifier deux graphes ; $G_1 \cup G_2 = G_{out} \mid V_{out} = V_1 \cup V_2$; $E_{out} = E_1 \cup E_2$; $\gamma_{out} : E_{out} \rightarrow V_{out} \times V_{out} \mid e_{out} \in \gamma_1 \vee e_{out} \in \gamma_2$.

Algorithme : Extension automatique de la requête utilisateur

entrée : Q
sortie : Q_{ext}

begin
  $Q_{ext} \leftarrow id$
  foreach $q_x \in Q$ **do**
    switch q_x **do**
      case $_{Pattern}\pi_{Attribute}$ **do** *// projection*
        $L_{ext} \leftarrow \prod_{i=1}^{p} \nabla_i$
        $A_{ext} \leftarrow \prod_{\forall a_{i,k} \in Attribute} \Delta_{i,k}$
        $q_{ext} \leftarrow id$
        foreach $L \in L_{ext}$ **do**
          foreach $A \in A_{ext}$ **do**

```
                    if exists(A,L) then q_ext ← q_ext ∪ _L π_A
                    end
                end
            end
            Q_ext ← Q_ext o ( q_ext )
        end
        case _Pattern σ_Predicate do                    // sélection
            L_ext ← ∏_{i=1}^{p} ∇_i
            P_ext ← ∧_x ( ∨_y ( ∨_{a_{l,k}∈Δ_{x,y}} l_x.a_{l,k} ϖ_{l,k} v_{l,k} ) )
            q_ext ← id
            foreach L∈L_ext do
                foreach P∈P_ext do
                    if exists(P,L) then q_ext ← q_ext ∪ _L σ_P
                    end
                end
            end
            Q_ext ← Q_ext o ( q_ext )
        end
    end
end
end.
```

Exemple. Considérons la requête $_{\texttt{Author-}\ -1}\pi_{\texttt{Author.firstname,Author.lastname,1.title}}(_{\texttt{Author-}\ -1}\sigma_{\texttt{Author.firstname='Charles'}\wedge\texttt{Author.lastname='Baudelaire'}})$.

L'opérateur de projection est réécrit en fonction des différents labels similaires du pattern, ∇_{Author} = { `Author`, `author`, `Writer` } et ∇_{Book} = { `Book`, `book`, `Publication` }, et des différents attributs similaires, $\Delta_{Author.firstname}$ = { `Author.firstname`, `Writer.firstname`, `author.firstname` } et $\Delta_{Author.lastname}$ = { `Author.lastname`, `Writer.lastname`, `author.lastname` }.

L'algorithme construit les ensembles suivants à partir desquels l'opérateur est réécrit.

L_{ext} = { `Author`, `author`, `Writer` } x {} x { `Book`, `book`, `Publication` }
= { {`Author,,Book`}, {`Author,,book`}, {`Author,,Publication`}, {`Writer,,Book`}, {`Writer,,book`}, {`Writer,,Publication`}, {`author,,Book`}, {`author,,book`}, {`author,,Publication`} }

A_{ext} = { `Author.firstname`, `Writer.firstname`, `author.firstname` } x { `Author.lastname`, `Writer.lastname`, `author.lastname` } x { `Book.title`, `book.title`, `Publication.title` }
= { {`Author.firstname,Author.lastname,Book.title`}, …, {`author.firstname,author.lastname,Publication.title`} }

La projection ainsi réécrite est de la forme suivante :

$_{\texttt{Author- -Book}}\pi_{\texttt{Author.firstname,Author.lastname,Book.title}} \cup$

$_{\texttt{Writer- -Book}}\pi_{\texttt{Writer.firstname,Writer.lastname,Book.title}} \cup$

$_{\texttt{author- -Book}}\pi_{\texttt{author.firstname,author.lastname,Book.title}} \cup$

$_{\texttt{Author- -book}}\pi_{\texttt{Author.firstname,Author.lastname,Book.book}} \cup$

$_{\texttt{Writer- -book}}\pi_{\texttt{Writer.firstname,Writer.lastname,Book.book}} \cup$

$$\text{author- -book}\pi_{\text{author.firstname,author.lastname,Book.book}} \cup$$

$$\text{Author- -Publication}\pi_{\text{Author.firstname,Author.lastname,Book.Publication}} \cup$$

$$\text{Writer- -Publication}\pi_{\text{Writer.firstname,Writer.lastname,Book.Publication}} \cup$$

$$\text{author- -Publication}\pi_{\text{author.firstname,author.lastname,Book.Publication}}$$

L'opérateur de sélection est réécrit en fonction des différents labels similaires du pattern de sélection, ∇_{Author} = { Author, author, Writer }, et des différents attributs similaires du prédicat, $\Delta_{Author.lastname}$ = { Author.firstname, Writer.firstname, author.firstname } et $\Delta_{Author.lastname}$ = { Author.lastname, Writer.lastname, author.lastname }.

L'algorithme construit les ensembles suivants à partir desquels l'opérateur est réécrit.

L_{ext} = { Author, author, Writer } x {} x { Book, book, Publication }
= { {Author,,Book}, {Author,,book}, {Author,,Publication},
{Writer,,Book}, {Writer,,book}, {Writer,,Publication},
{author,,Book}, {author,,book}, {author,,Publication} }

On représente les prédicats normalisés avec des ensembles :

Autheur.firstname='Charles' ∧ Autheur.lastname='Baudelaire'
≡ { { Author.firstname='Charles' }, { Author.lastname='Baudelaire' } }

Ainsi :

P_{ext} = { { Author.firstname='Charles', Writer.firstname='Charles', author.firstname='Charles' }, { Author.lastname='Baudelaire', Writer.lastname='Baudelaire', author.lastname='Baudelaire' } }

La sélection devient alors :

$$\text{Author- -Book}\sigma_{\text{Author.firstname='Charles'}\wedge\text{Author.lastname='Baudelaire'}} \cup$$

$$\text{Writer- -Book}\sigma_{\text{Writer.firstname='Charles'}\wedge\text{Writer.lastname='Baudelaire'}} \cup$$

$$\text{author- -Book}\sigma_{\text{author.firstname='Charles'}\wedge\text{author.lastname='Baudelaire'}} \cup$$

$$\text{Author- -book}\sigma_{\text{Author.firstname='Charles'}\wedge\text{Author.lastname='Baudelaire'}} \cup$$

$$\text{Writer- -book}\sigma_{\text{Writer.firstname='Charles'}\wedge\text{Writer.lastname='Baudelaire'}} \cup$$

$$\text{author- -book}\sigma_{\text{author.firstname='Charles'}\wedge\text{author.lastname='Baudelaire'}} \cup$$

$$\text{Author- -Publication}\sigma_{\text{Author.firstname='Charles'}\wedge\text{Author.lastname='Baudelaire'}} \cup$$

$$\text{Writer- -Publication}\sigma_{\text{Writer.firstname='Charles'}\wedge\text{Writer.lastname='Baudelaire'}} \cup$$

$$\text{author- -Publication}\sigma_{\text{author.firstname='Charles'}\wedge\text{author.lastname='Baudelaire'}}$$

4.4. L'architecture de EasyGraphQuery

L'ensemble des mécanismes est mis en œuvre dans le système *EasyGraphQuery* dont l'architecture est donnée Figure 3, et **Erreur ! Source du renvoi introuvable.**comprenant les composantes suivantes :

– ***Dictionary*** : (cf. Section 4.1).

- ***SimilartyMatrix*** : il s'agit des matrices de similarité syntaxique et sémantique stockées sous forme d'un fichier JSON, hébergé dans le répertoire de Neo4J.
- ***Query Rewriter engine*** : prend en entrée la requête adressée par l'utilisateur, extrait les attributs similaires depuis le dictionnaire et reformule la requête.
- ***Synchronisation Engine***. Permet d'actualiser le dictionnaire à chaque requête d'insertion. Pour ce faire, on utilise deux fichiers pour sauvegarder la date des dernières mises à jours ; le premier *DMJ_Dictionary* enregistre, pour chaque attribut, la date de modification faite au niveau du dictionnaire, sous la forme « *attribut : date de modification* » tandis que le second *DMJ_Matrix* enregistre pour chaque attribut la date de modification au niveau des matrices de similarité. Avant de toute actualisation du dictionnaire on compare si la date de *DMJ_Matrix* est plus récente.
- ***Data structure extractor* :** extrait les noms des labels et des attributs depuis le fichier des logs pour éventuellement les insérer dans les matrices.

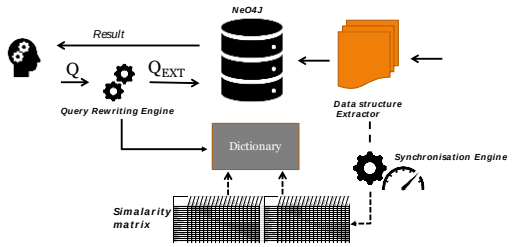


Figure 3 : L'architecture d'*EasyGraphQuery*.

La création du dictionnaire est faite de manière automatique au moment de l'insertion des données puis il est actualisé à chaque nouvelle insertion ou une actualisation des données. Pour des raisons de performance, la mise à jour est effectuée en continu et en arrière-plan.

5. Expérimentations

Jeu de données. Pour valider notre approche, nous considérons des données d'ontologies en raison de leur forte hétérogénéité structurelle. Nous avons utilisé la collection *Conference Track* mise à disposition par OAEI 2017[2]. Nous avons généré des instances synthétiques : 16 ontologies décrivant chacune une organisation d'une conférence. Le but étant d'évaluer le coût d'interrogation, les temps de génération et de chargement ne sont pas évalués.

Environnement de tests. Nous employons un cluster composé d'une machine (i5-4 cœurs, 8Go de RAM, 2To de disque, 1Gb/s de réseau) sur lequel nous avons installé une instance de Neo4J – version 3.2.

[2] http://oaei.ontologymatching.org/2017/

Le jeu de requêtes. Nous avons défini un jeu de 10 requêtes ; 3 requêtes de sélection, 3 requêtes de projection et 4 requêtes pour évaluer la combinaison sélection-projection. Pour la projection, le nombre d'attributs projeté est choisi entre 1 et '*'.

5.1. Construction du dictionnaire et de la matrice de similarité

Dans cette expérience nous étudions le temps nécessaire pour la création et l'actualisation du dictionnaire de similarité. Le Tableau 1 montre le temps de maintenance du dictionnaire au-fur-à-mesure que les ontologies sont insérées. Le résultat est nettement influencé par le nombre d'éléments (labels, relations, attributs) déjà présents dans le graphe. En effet, le fichier des logs est régulièrement analysé par notre parser mais il n'est pas nettoyé à chaque passage.

Tableau 1 Temps de maintenance du dictionnaire (en secondes) en fonction du nombre d'ontologies

Nombre d'ontologies	2	4	6	8
Tems de création /mise à jour (en secondes)	1.3s	4.2s	13.5s	18.7s
Taille du dictionnaire (KB)	2.7KB	2.9KB	3.4KB	3.5KB
Taille du fichier de logs analysé (parsé) en KB	1333KB	14534KB	17602KB	21265KB

5.2 Evaluation du module de réécriture de requêtes

Dans cette expérience, nous étudions le coût additionnel de notre solution, c'est à dire une interrogation avec une reformulation de la requête via notre algorithme de similarité, par rapport au coût de la requête sans reformulation, dite requête initiale. Nous comparons aussi le coût de la requête reformulée par rapport au cumul des coûts des sous-requêtes i.e. celles provenant de la décomposition des requêtes reformulées.

Tableau 2 Comparaison du temps d'éxécution (en secondes) des requêtes reformulées et des requêtes initiales (sans reformulation)

		Requête reformulée	Requête initiale	Cumul de requêtes résultantes
Projection	Q1.1	316	222	316
	Q1.2	0.160	0.008	0,166
	Q1.3	0.027	0.0013	0.027
Sélection	Q2.1	4.05	2.98	4.8
	Q2.2	0.77	0.77	0.85
	Q2.3	2.34	1.73	3.73
Combinaison (projection & sélection)	Q3.1	0.2734	0.2082	0.4062
	Q3.2	0.0055	0.0059	0.0073
	Q3.3	0.434	0.0431	0.9342
	Q3.4	0.324	0.324	0.659

Le Tableau 2 rapporte le temps d'exécution des requêtes reformulées, les requêtes initiales et le cumul des temps d'exécution des sous-requêtes. Une première

comparaison porte sur les temps d'exécution des requêtes réécrites et le cumul de temps d'exécution des sous requêtes. Nous pouvons observer que notre solution est, au pire, égale au cumul des sous requêtes, et elle peut aller jusqu'à 2 fois plus vite (cas des requêtes de combinaison par exemple dans le cas de notre jeu de données). En revanche, elle est au mieux égale au temps d'exécution d'une requête initiale.

Pour mieux interpréter ces résultats nous avons tracé l'exécution de nos requêtes. où nous pouvons remarquer par exemple que pour la requête Q1.2 reformulée (où notre algorithme fait appel à l'opérateur '*Union'*), Neo4J lance l'exécution des deux '*Match'* en parallèle ; et l'union des deux résultats n'est consolidé qu'après la fin de l'exécution du dernier '*Match'* (celui comportant le plus grand nombre de lignes). Plus précisent dans cette requête de projection Q2.1, deux types de labels sont évalués : le premier correspond au label de la requête initiale et qui traite 35054 lignes ; le second correspond au label ajouté par notre algorithme de réécriture et qui traite 10000 lignes. Le nombre de lignes explique les résultats du Tableau 2 et montre pourquoi notre solution est au au pire égale au temps d'exécution de la sous requêtes la plus lente et au mieux elle est égale à la requête initiale.

6. Conclusion

Dans cet article nous avons défini une approche qui repose sur la construction de dictionnaires de similarité des données permettant une réécriture des requêtes utilisateurs sans transformer les données stockées. Cette approche calcule, pour chaque attribut l'ensemble des attributs similaires (hétérogénéités syntaxique, sémantique et structurelle) pour réécrire de manière transparente les requêtes des utilisateurs. Les requêtes réécrites permettent de compléter les requêtes initiales et retourner l'ensemble complet des données. L'originalité de notre approche est de prendre en compte la variabilité de données « à la place de l'utilisateur ». Cette variabilité n'est plus résolue lors de l'écriture de chaque requête mais lors de la construction du dictionnaire qui reste un élément à approfondir. Nous avons abordé ce point au travers de matrices de similarité dans cet article et d'autres modalités de calcul restent à étudier. L'apport majeur est que la même requête utilisateur évaluée à des moments différents sera évaluée en fonction de l'état courant du dictionnaire : si des nouvelles données hétérogènes ont été ajoutées, cette variabilité sera automatiquement prise en compte dans l'évaluation de la requête.

En perspectives, nous souhaitons élargir le noyau algébrique d'opérateurs supportés dans notre approche ; en intégrant les opérations d'agrégation. Nous souhaitons également étudier l'impact des matrices avec un volume plus important.

Bibliographie

Beyer, K. S., V. Ercegovac, R. Gemulla, A. Balmin, M. Eltabakh, C.-C. Kanne, F. Ozcan, et E. J. Shekita . Jaql : A scripting language for large scale semi structured data analysis. VLDB (2011).

Floratou, A., N. Teletia, D. J. DeWitt, J. M. Patel, et D. Zhang (2012). Can the elephants handle the nosql onslaught ? VLDB 5(12), 1712–1723.

Florian Holzschuher and René Peinl. 2013. Performance of graph query languages: comparison of cypher, gremlin and native access in Neo4J. (EDBT '13), 195-204.

Getoor, L., Machanavajjhala, A. (2012). Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, Vol.5(12), 2018-2019.

H. Ben Hamadou, F. Ghozzi, A. Péninou, O. Teste. Interrogation de données structurellement hétérogènes dans les bases de données orientées documents (EGC 2018).

M. Chevalier, M. El Malki, A. Kopliku, R. Tournier, Olivier Teste. How Can We Implement a Multidimensional Data Warehouse Using NoSQL? Springer, p. 108-130, Vol. 241, (LNBIP), 2015.

R. Cattell. 2011. Scalable SQL and NoSQL data stores. SIGMOD Rec. 39, 4 (May 2011), 12-27.

I. Megdiche, O. Teste, C. Trojahn dos Santos, An Extensible Linear Approach for Holistic Ontology Matching, (ISWC'16), p.393-410.

S. Melnik, H. Garcia-Molina and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its application to schema matching, ICDE (2002), p. 117-128.

Shvaiko, P., Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV*, Stefano Spaccapietra (Ed.) Springer, 146-171.

Stonebraker, M. (2012). New opportunities for new sql. Communications of the ACM 5(11), 10–11.

Tahara, D., T. Diamond, et D. J. Abadi (2014). Sinew : a sql system for multi-structured data. In 2014 SIGMOD, pp. 815–826. ACM.

Thusoo, A., J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, et R. Mur- thy (2009). Hive : a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment 2(2), 1626–1629.

Wang, L., S. Zhang, J. Shi, L. Jiao, O. Hassanzadeh, J. Zou, et C. Wangz (2015). Schema management for document stores. Proceedings of the VLDB Endowment 8(9), 922–933.

Ontologies et contexte

Vers une typologie de contexte pour les systèmes de recommandation

Elsa Negre

Paris-Dauphine University
PSL Research Universities
CNRS UMR 7243, LAMSADE
75016 Paris, France
elsa.negre@dauphine.fr

RÉSUMÉ. *La montée en volume des données et informations de sources diverses exige un filtrage d'informations efficace afin d'être au plus près de l'utilisateur et répondre au mieux à ses besoins. Dans cet objectif, les systèmes de recommandation contextuels qui prennent en compte le contexte de l'utilisateur dans leur processus de recommandation, ont été proposés. Cependant il n'existe toujours pas de définition unique pour ce contexte. Dans cet article nous proposons une typologie du contexte utilisateur dans le cadre des systèmes de recommandation contextuels, afin de pallier les manques des précédentes propositions et répondre à un spectre assez large de cas d'application. En effet, cette typologie se veut générique avec une grande applicabilité.*

ABSTRACT. *The rise in volume of data and information from various sources requires effective information/data filtering to be closer to the user and best meet his/her needs. For this purpose, context-aware recommender systems that take into account the context of the user in their recommendation process, have been proposed. However, there is still no unique definition for context. In this article, we propose a typology of the user context for (context-aware) recommender systems, in order to overcome the shortcomings of the previous proposals and to answer a rather wide spectrum of application cases. Indeed, this typology is generic with great applicability.*

MOTS-CLÉS : *Systèmes de recommandation, Contexte, Systèmes d'information, Aide à la décision*

KEYWORDS: *Recommender systems, Context, Information systems, Decision support systems*

DOI:10.3166/HSP.1.1-16

1. Introduction

La quantité d'information disponible sur le web est de plus en plus importante. Les utilisateurs peuvent facilement être envahis par ces données et informations. C'est alors que les techniques informatiques qui facilitent la recherche, l'extraction et le filtrage d'informations pertinentes paraissent nécessaires. L'une d'entre elles est la recommandation. Un système de recommandation propose à l'utilisateur des éléments qui sont susceptibles de l'intéresser. Beaucoup de systèmes de recommandation traditionnels, comme *Amazon* ou *Netflix* ont fait leurs preuves. Ils se basent essentiellement sur les notes attribuées par les utilisateurs aux éléments. Ces dernières années une nouvelle approche de recommandation a émergé, appelée recommandation contextuelle, qui a comme objectif d'être au plus près de l'utilisateur. En effet, on peut améliorer la pertinence des recommandations en prenant en compte des informations complémentaires, comme l'environnement et la situation actuelle dans laquelle l'utilisateur se trouve, ou plus précisément son contexte actuel. Des travaux comme ceux de (Riboni, Bettini, 2011) ont prouvé la corrélation entre le comportement d'un utilisateur et son contexte, d'où l'importance de l'intégration du contexte utilisateur dans le processus de recommandation.

Cependant la notion de contexte reste floue. En effet, à cause d'un manque de consensus, il n'existe toujours pas de définition unique pour le contexte.
L'intérêt du contexte dans les systèmes de recommandation n'a été constaté que très récemment (Baltrunas *et al*., 2012). En effet, les systèmes de recommandation traditionnels se basent seulement sur les utilisateurs et les éléments pour faire leurs recommandations, sans prendre en compte le contexte actuel de l'utilisateur. Cependant celui-ci peut influencer les intérêts de l'utilisateur, c'est pourquoi il est important de prendre en compte ce type d'informations complémentaires afin d'obtenir des recommandations plus appropriées (Baltrunas *et al*., 2012).

L'objectif de cet article est d'améliorer la représentation du contexte de l'utilisateur dans le cadre des systèmes de recommandation (contextuels), en proposant une typologie de contexte sous la forme d'une catégorisation hiérarchique des facteurs de contexte. Cette typologie se veut générique et applicable dans de nombreux domaines. Cela est un pas vers l'amélioration des systèmes de recommandation.

Cet article est organisé ainsi : nous présentons dans la section 2, la notion de contexte, puis dans la section suivante, notre typologie de contexte. La section 4 illustre l'applicabilité et la généricité de notre typologie de contexte. Enfin, nous concluons et énonçons quelques perspectives de recherche en section 5.

2. Contexte

Malgré de bonnes performances des systèmes de recommandation, les recommandations ne sont parfois « pas assez pertinentes ». Par exemple, supposons que recommander un film à un jeune homme de 20 ans consiste à lui proposer des films de guerre ou d'action et supposons que lorsqu'il est avec une amie, il préfère qu'on lui recommande des films romantiques. Ici le contexte influence les préférences, les envies et les intérêts des utilisateurs et de ce fait, leurs décisions. Nous souhaitons donc intégrer les

données/informations (qu'elles soient qualitatives et/ou quantitatives) contextuelles au système de recommandation qui devient un système de recommandation contextuel.

La notion de « contexte » a été étudiée dans divers domaines, notamment en informatique ubiquitaire (*pervasive and ubiquitous computing*) (Riboni, Bettini, 2011 ; Henricksen, Indulska, 2006 ; Schilit, Theimer, 1994) mais il est difficile d'établir une définition standard (unique) en raison de la nature multiforme du contexte (Bazire, Brézillon, 2005). La définition la plus acceptée est celle de (Dey *et al*., 2001) qui définit le contexte comme toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité peut être une personne, un lieu ou un objet qui est considéré pertinent dans l'interaction entre un utilisateur et une application, tout en incluant ces deux derniers. Afin de connaître le contexte, il faut collecter les informations contextuelles. Le contexte peut être capturé, collecté explicitement ou implicitement (Mostefaoui *et al*., 2004).

Les systèmes de recommandation traditionnels se basent seulement sur les utilisateurs et les éléments pour faire leurs recommandations, alors que les systèmes de recommandation contextuels, quant à eux, prennent en compte le contexte de l'utilisateur qui peut influencer les intérêts/besoins de l'utilisateur. Ainsi, les systèmes de recommandation contextuels existants sont plus efficaces/pertinents (c'est-à-dire plus en adéquation avec les besoins de l'utilisateur) que les systèmes de recommandation traditionnels (Adomavicius, Tuzhilin, 2011). Cependant, leur performance est impactée par le contexte, qui est flou et omniprésent (Bazire, Brézillon, 2005).

Notre objectif est d'aller vers une modélisation du contexte pour une meilleure prise en compte de celui-ci. Pour atteindre cette modélisation, dans (Ferdousi *et al*., 2017), à partir d'une étude bibliographique, nous avons identifié dans un premier temps tous les facteurs de contexte qui ont été étudiés.

Par ailleurs, la modélisation du contexte reste encore compliquée compte tenu de la nature des données et/ou informations contextuelles : en effet le modèle doit pouvoir gérer les sources de données variées, l'hétérogénéïté au niveau de leur qualité et de leur durée de vie et la nature imparfaite de celles-ci (Henricksen, Indulska, 2006). Il existe différents modèles de contexte (Bettini *et al*., 2010) : attribut/valeur, mots clés, graphes, logique, ...

Il est à noter que dans cet article, nous abordons la représentation et la modélisation du contexte mais pas son intégration au sein du système de recommandation. Cependant, plusieurs techniques existent pour l'intégration du contexte dans le processus de recommandation : (i) le pré-filtrage, où seules les données qui correspondent au contexte sont sélectionnées (comme entrées), puis une méthode de recommandation traditionnelle est appliquée sur celles-ci, (ii) le post-filtrage, où une méthode de recommandation traditionnelle est appliquée sur la totalité des données, puis les résultats de la recommandation (sorties) sont ajustés en fonction du contexte, et (iii) le *context modeling* dans lequel les données et informations contextuelles sont directement prises en compte dans le processus de recommandation (les étapes) comme une dimension supplémentaire (Adomavicius, Tuzhilin, 2011).

3. Les facteurs de contexte de l'utilisateur

Afin de décrire plus concrètement en quoi consiste le contexte d'un utilisateur, de multiples catégorisations de facteurs de contexte ont été proposées (Ingwersen, Järvelin, 2005). A partir d'une étude bibliographique [1] (dont une partie est synthétisée dans le tableau 1) et des facteurs de contexte présentés par (Adomavicius, Tuzhilin, 2011), nous avons regroupé/recoupé et structuré les travaux existants. Puis nous avons proposé une catégorisation hiérarchique, comme illustrée sur la figure 1 (Ferdousi *et al.*, 2017). Cette catégorisation hiérarchique est un arbre, c'est-à-dire un graphe non orienté, acyclique et connexe, vu comme la généralisation de listes (puisque les listes peuvent être représentées par des arbres) favorisant ainsi son utilisation. Nos objectifs pour cette nouvelle proposition de catégorisation des facteurs de contexte sont de répondre aux besoins des systèmes de recommandation contextuels tout en (i) satisfaisant [2] la définition de (Dey *et al.*, 2001), (ii) compensant les lacunes (non-satisfaction de la définition de (Dey *et al.*, 2001) ou caractérisation non optimale) des précédentes propositions, (iii) permettant de travailler le contexte sur plusieurs niveaux [3] et (iv) permettant de l'appliquer sur un spectre assez large de cas d'applications [4].

Notre catégorisation a comme principales familles de contexte : le contexte physique, le contexte personnel et le contexte technique (Ferdousi *et al.*, 2017). Le contexte utilisateur est alors l'union de toutes ces familles de contexte, chacune composée de plusieurs unités [5] :

1. Le contexte physique regroupe tous les aspects sur lesquels la position géographique de l'utilisateur va avoir une forte influence. Nous avons regroupé plusieurs unités dans cette catégorie :

a) L'unité temporelle : comme le moment de la journée, semaine/weekend, la saison, les évènements (anniversaire, nouvel an, etc),

b) L'unité spatiale : qui selon le cas d'application peut être représentée par la position géographique exacte (coordonnées GPS, longitude/latitude) ou par des classes nominales (pour dire si l'utilisateur est chez lui, au travail, en voyage, etc).

---

1. Il est à noter que les facteurs "Individualité / Profil utilisateur" et "Localisation" sont les deux seuls facteurs de contexte utilisés par tous. Il est également important de remarquer qu'aucun des travaux ne prend en compte tous les facteurs. Cela vient du fait que selon les domaines d'application, certains facteurs de contexte sont considérés plus utiles que d'autres.

2. Les caractéristiques des lieux, des personnes et des objets qui jouent un rôle dans l'interaction entre l'utilisateur et l'application sont respectivement représentés par les unités du contexte physique, par l'unité sociale du contexte utilisateur et par l'unité équipement du contexte physique. Les caractéristiques de l'utilisateur et de l'application apparaissent dans les contextes personnel et technique.

3. Notre catégorisation à plusieurs niveaux peut s'adapter à différents cas d'applications et à leurs besoins, et permet de travailler le contexte de façon plus générale ou plus fine.

4. Nous espérons tendre vers une représentation, rassemblant tous les facteurs de contexte possible, pour une application au plus grand nombre de domaines.

5. Nous sommes conscients que notre catégorisation en trois grandes familles est discutable, mais comme toute catégorisation hiérarchique, il est possible de grouper/dissocier certaines familles/unités afin d'avoir un plus grand/petit nombre de familles/unités.

Tableau 1. Extrait de facteurs de contexte présents dans la littérature

	(Benouaret, 2015)	(Akermi *et al.*, 2015)	(Schilit, Theimer, 1994)	(Brown *et al.*, 1997)	(Burke *et al.*, 1997)	(Petrelli *et al.*, 2000)	(Nguyen, 2010)
Temps	X	X		X			X
Individualité / Profil d'utilisateur	X	X	X	X	X	X	X
Activité	X	X					X
Relations	X						X
Localisation	X	X	X	X	X	X	X
Objet			X		X	X	X
Saison				X			
Température				X			
Contexte social						X	X
Contexte matériel						X	X

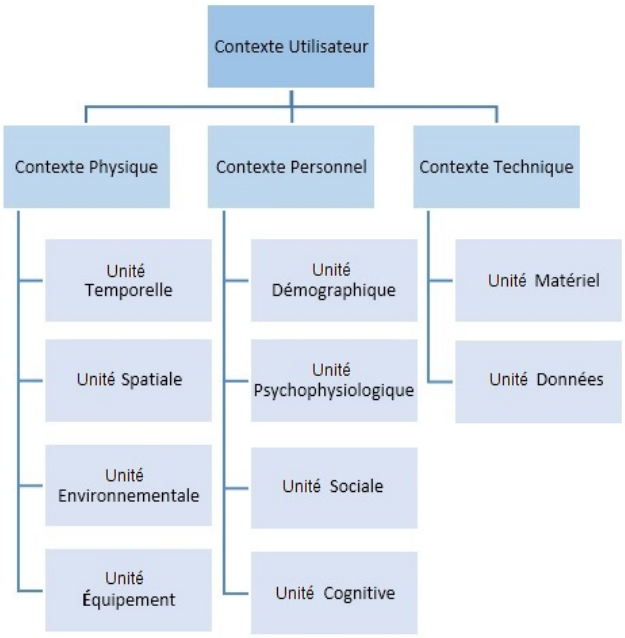


Figure 1. Catégorisation des facteurs de contexte

c) L'unité environnementale : en fonction du cas d'application, cette unité peut représenter des caractéristiques environnementales comme la température, la météo, le niveau de luminosité ou le niveau sonore du lieu où l'utilisateur se trouve, et/ou la situation régionale de ce lieu, comme une guerre, une catastrophe naturelle, une crise économique, ...

d) L'unité équipement : tout ce (non humain : objet ou espace) qui entoure l'utilisateur comme : barbecue, ustensiles, électroménager (friteuse, four, etc), imprimante, jardin/terrasse, ...

2. Le contexte personnel représente les informations plus spécifiques de l'utilisateur qui sont regroupées en quatre unités :

a) L'unité démographique regroupe notamment les informations sur l'identité (le nom, l'âge, le sexe, la nationalité, etc) de l'utilisateur.

b) L'unité sociale fait référence à la présence et au rôle des autres personnes autour de l'utilisateur. Selon le cas d'application, on peut prendre en compte seulement les personnes présentes pendant l'utilisation de l'application par l'utilisateur, les personnes avec lesquelles on veut partager l'activité en question, ou bien aller plus loin en prenant en compte les relations plus fines, comme les amis, la famille, les collègues, les voisins, ...

c) L'unité psychophysiologique représente l'aspect psychophysiologique de l'utilisateur, comme son état d'esprit, son humeur, son degré de fatigue, ...

d) L'unité cognitive fait référence aux expériences de l'utilisateur, ses objectifs, ses contraintes, son activité, ...

3. Le contexte technique représente les caractéristiques des dispositifs utilisés par l'utilisateur. Cela peut être :

a) Le matériel utilisé par l'utilisateur pour accéder au système de recommandation contextuel, comme le dispositif utilisé, les processeurs, la capacité du réseau, ...

b) Les données qui sont manipulées par l'application, leur format (texte, film, audio, image, etc), leur sources de provenance, la qualité de ces données, leur période de validité, leur exactitude, ...

Il est à noter que chaque unité sera instanciée en fonction du cas d'application, de la pertinence et de la disponibilité des indicateurs associés.

Finalement, nous souhaitons aller plus loin qu'une nouvelle catégorisation en proposant notre catégorisation (Ferdousi *et al*., 2017) comme une typologie, i.e. un système de classification générique et standard pour les données/informations contextuelles dans les systèmes de recommandation. A cet effet, dans la section suivante, nous montrons l'applicabilité de notre typologie sur trois domaines d'application avec des enjeux et des données différents.

4. Applications

Ainsi, nous précisons dans cette section comment instancier notre catégorisation du contexte dans trois domaines d'applications : OLAP (*OnLine Analytical Process*),

un environnement commercial, et la gestion de crise. Nous les présentons en fonction de la densité des informations contextuelles, c'est-à-dire de la quantité de familles/unités de contexte prises en compte.

4.1. En OLAP

Recommander une requête OLAP à un décideur d'une grande entreprise française consistera à retourner des requêtes relatives au chiffre d'affaire annuel de la société. Mais si la société s'ouvre à l'international (États-Unis par exemple) ou si certaines données (anciennes) ont été archivées alors le décideur préfèrerait que le système lui recommande des requêtes « récentes » ou en rapport avec les États-Unis. En OLAP, les systèmes de recommandation prennent souvent un log parmi leurs données d'entrée. Or, les analyses OLAP (stockées dans le log) sont lancées régulièrement mais cette régularité peut être annuelle. Donc, le log peut contenir des données très anciennes sans rapport avec les analyses réalisées au moment de l'utilisation par le système de recommandation. De sorte que le système de recommandation retourne des recommandations non pertinentes.
Prendre en compte des informations contextuelles permettrait de pallier ce manque de pertinence.

4.1.1. Familles et unités de contexte

Nous avons commencé par déterminer quelles informations contextuelles sont utiles pour les systèmes de recommandation en OLAP (Negre, 2017). En effet, dans (Negre, 2017), cinq critères pertinents de contexte ont été détectés : Temps, Individualité/Profil Utilisateur, Activité, Relations et Contexte matériel. Nous les présentons ici selon la catégorisation des facteurs de contexte présentée précédemment. Il est à noter que les familles/unités de contexte, qui n'apparaissent pas, ont été considérées comme inutiles en fonction des spécificités d'OLAP.

4.1.1.1. Contexte Physique

- Unité temporelle : relative au critère *Temps* de (Negre, 2017).

4.1.1.2. Contexte Personnel

- Unité démographique : relative au critère *Individualité/Profil Utilisateur*.
- Unité sociale : relative au critère *Relations*.
- Unité cognitive : relative au critère *Activité*.

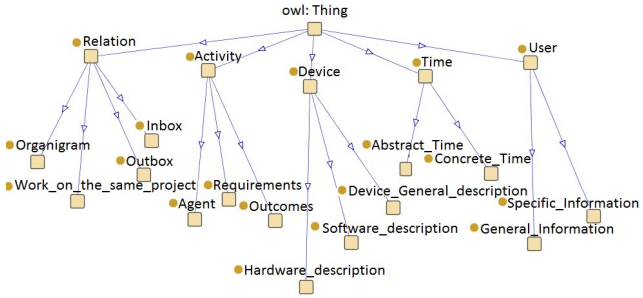
4.1.1.3. Contexte Technique

- Unité Matériel : relative au critère *Contexte matériel*.

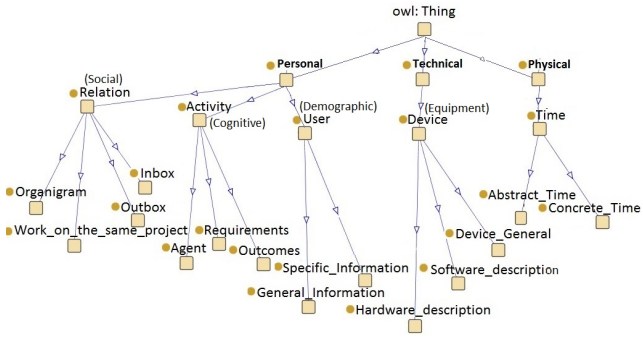
Finalement, ici, chaque critère de (Negre, 2017) est associé à une unité de contexte différente.

4.1.2. Modélisation

Une modélisation des cinq critères pertinents a été proposée dans (Negre, 2017) sous la forme d'une ontologie générale (ontologie de domaine) comme illustré par la figure 2a (par souci de lisibilité, nous nous limitons à afficher le premier niveau). D'après (Soualah Alila, 2015), qui a fait une comparaison entre les différents modèles de contexte, le modèle ontologique permet une bonne validation partielle des données et une bonne formalisation du modèle.



(a) Ontologie (cinq critères) de (Negre, 2017)



(b) Ontologie avec notre typologie

Figure 2. Ontologie générale de contexte dans le cadre d'un système de recommandation de requêtes OLAP.

Passer de la catégorisation en cinq critères de (Negre, 2017) à notre proposition n'a pas d'impact sur la modélisation puisque chaque critère de (Negre, 2017) a été associé à une seule unité de contexte (seuls les intitulés changent comme illustré par la figure 2b).

4.2. Dans un environnement commercial

Nous nous sommes intéressés aux données/informations contextuelles dans un environnement commercial particulier : celui de la vente de véhicules d'occasion (Arduin *et al.*, 2014). Il est à noter que nous avons commencé à partir d'observations « terrain » et que l'approche proposée est spécifique au cas rencontré.

Acheter un véhicule d'occasion
En France, en général, pour vendre un véhicule à un distributeur, ce véhicule doit être en bon état et pas trop vieux. Le distributeur achète votre véhicule uniquement si vous achetez un nouveau dans son entreprise. Il peut se permettre d'offrir un prix d'achat plus élevé car il fera une marge bénéficiaire significative avec le véhicule que vous achèterez. La promesse de la société sur laquelle porte notre étude, est d'acheter tous les véhicules sans condition, quel que soit leur état et sans obligation d'acheter un autre véhicule.

Vendre un véhicule d'occasion
Lorsqu'une personne (privée), Elsa, veut vendre son véhicule, elle demande une évaluation préalable du véhicule d'occasion sur le site de la société. Elle donne des informations sur les principales caractéristiques du véhicule telles que la marque, le modèle et l'âge. À la suite de cette requête, elle obtient une évaluation du prix d'achat du véhicule. Le système ne prend pas en compte l'usure du véhicule et il estime que le véhicule est en bon état. Lors de l'interrogation pour obtenir une première évaluation sur le site web, Elsa renseigne son adresse et son numéro de téléphone, elle sera contactée par la société pour prendre rendez-vous dans une agence. Quand Elsa va à l'agence, elle est reçue par un expert automobile, John. Il examine le véhicule et évalue les coûts pour reconditionner le véhicule. Cette étape est très importante car elle contrôle l'état de fonctionnement du véhicule, John doit faire un essai routier et identifier d'éventuelles anomalies. Les données sont intégrées et stockées via une plateforme informatique afin de demander aux experts en cotation une cotation du véhicule. La valeur que John obtiendra du service des achats à travers cette plateforme sera la meilleure proposition et John n'aura aucune marge de manœuvre sur cette proposition [6].

4.2.1. Familles et unités de contexte

Grâce à des entretiens et des échanges avec des experts et des vendeurs, il apparait que la prise en compte du contexte améliorerait l'évaluation du prix du véhicule ainsi que le succès de la transaction. Ainsi, quatre critères ont été détectés comme utiles : la localisation du véhicule, le marché, les coûts de réparation et le contexte personnel d'une vente particulière. Nous les présentons ici selon notre typologie :

4.2.1.1. Contexte Physique

- Unité spatiale : *La localisation du véhicule.* Par exemple, les experts savent que certains véhicules seront vendus différemment en fonction de leur situation géographique.

6. La façon dont les véhicules sont évalués n'est pas discutée ici. Cependant, cette évaluation est imprévisible parce qu'elle dépend de chaque expert en cotation. Pour mieux estimer le marché, ces experts utilisent différents sites publicitaires vendant des véhicules sur le web et examinent les prix. Ils évaluent le véhicule de sorte qu'il soit bien placé parmi les annonces similaires. Cette approche conduit parfois à ce que le prix d'achat offert au particulier ne soit pas conforme au prix réel de revente.

• Unité environnementale : *Le marché.* Par exemple, si un véhicule est très populaire, la société le gardera en stock moins de temps, donc ce sera moins cher. Cependant, si un véhicule n'est pas populaire, il se passera plus de temps entre l'achat du véhicule et sa revente, il y a donc un coût de stockage supplémentaire pour la société.

• Unité Équipement : *Les coûts de réparation.* Par exemple, une simple égratignure peut provoquer le remplacement d'une partie importante du véhicule et seul un expert peut le savoir.

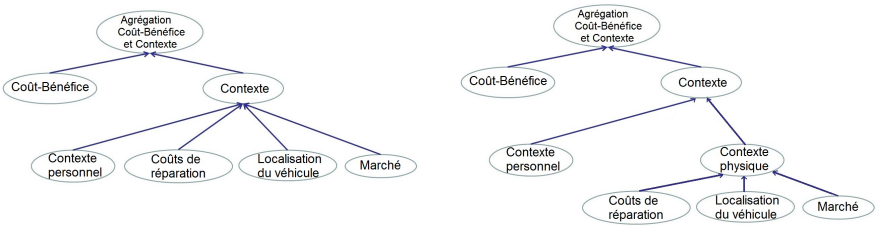
4.2.1.2. Contexte Personnel

• Unités démographique, psychophysiologique, sociale et cognitive : *Le contexte personnel d'une vente particulière.* Par exemple, les problèmes financiers ou familiaux peuvent engendrer un besoin urgent de vendre le véhicule quel que soit le prix ou bien, cela peut entraîner une tentative de négociation du prix aussi élevé que possible.

Finalement, ici, trois critères de (Arduin *et al*., 2014) sont associés, chacun, à une unité de contexte et un critère est associé à une famille de contexte.

4.2.2. Modélisation

Nous avons proposé de modéliser ce contexte en combinant une approche d'aide à la décision multicritère et une approche coût-bénéfice dans (Arduin *et al*., 2014). Les « meilleures » décisions possibles obtenues avec notre modèle pourront ensuite être utilisées par le système de recommandation comme des données/sources externes complémentaires.
Cette approche combinée est illustrée sur la figure 3a selon une hiérarchie de critères. Étant donné que les données relatives au coût-bénéfice n'étaient pas fournies par la société, nous avons modélisé uniquement, dans cette hiérarchie, le nœud associé aux facteurs de contexte.



(a) SANS la typologie de contexte *(b) AVEC la typologie de contexte*

Figure 3. Un modèle hiérarchique d'aide à la décision multicritère vu comme une agrégation des coûts-bénéfices et du contexte.

Passer de la catégorisation en quatre critères de (Arduin *et al*., 2014) à notre proposition n'a pas d'impact important sur la modélisation. En effet, le critère *Contexte personnel* est associé à une seule famille de contexte et les trois autres critères sont associés à trois unités de contexte différentes d'une même famille. Le modèle hiérarchique de la figure 3a deviendrait celui de la figure 3b permettant de faire apparaitre

une niveau intermédiaire dans la hiérarchie mais qui n'aurait pas d'impact sur l'analyse multicritère car celle-ci utilise des critères détaillés.

4.3. En gestion de crises

Dans le cadre de la gestion de crises, les informations contextuelles sont nombreuses. Celles relatives à l'individu peuvent être obtenues via son comportement, c'est-à-dire qu'elles sont collectées implicitement. Nous nous sommes donc intéressés au comportement des populations. Ainsi, nous avons mis en adéquation les concepts de comportement et de contexte.

4.3.1. Comportement

Le comportement est un concept qui nécessite d'être précisé et bien défini. Nos travaux s'appuient sur la définition proposée par (Sillamy, 1983) pour qui le comportement correspond aux « réactions d'un individu, considéré dans un milieu et dans une unité de temps donnée à une excitation ou un ensemble de stimulations ». Cette définition permet de situer clairement le comportement dans un espace spatio-temporel et comme une réponse à un ensemble d'excitations ou stimulations. Notons que nous limitons l'étude des comportements aux réactions observables par une entité extérieure.

Dans (Arru, Negre, 2017), nous avons décrit les comportements individuels en décomposant le plus finement possible les différents facteurs orientant les comportements, à l'aide d'éléments mesurables, les indicateurs.

Nous avons ainsi détecté vingt facteurs (et environ 70 indicateurs associés) du comportement, ayant une influence sur les réactions humaines en situation de crise.

4.3.2. Familles et unités de contexte

Nous présentons les facteurs de comportement selon la catégorisation des facteurs/unités de contexte présentée précédemment.

4.3.2.1. Contexte Physique

- Unité temporelle : les caractéristiques de la période pendant laquelle survient la crise (jour/nuit, ...) et la phase temporelle de la crise (avant, au commencement, ...)
- Unités spatiale et environnementale : les caractéristiques de la zone géographique (étendue de la zone, densité de la population, ...).
- Unité Équipement : la capacité d'interaction et de mobilité (fréquentation de la zone, nombre de smartphones par habitant, accès aux transports, ...).

4.3.2.2. Contexte Personnel

- Unité démographique : l'état civil (âge, sexe, lieu de résidence, ...)
- Unité psychophysiologique : la personnalité (désirs, principes moraux, sociabilité, ...), la motivation à évacuer/à défendre, les émotions (joie, tristesse, colère, ...) et les signaux physiologiques (rythme cardiaque, tension, niveau de transpiration, ...).

• Unité sociale : le niveau de responsabilité, les caractéristiques de l'entourage (densité des personnes à proximité de l'individu, présence de représentants de l'autorité, ...), le comportement des personnes les plus proches, et le comportement global de l'entourage.

• Unité cognitive : l'expérience et les capacités, les connaissances explicitées, l'évaluation du risque, la perception du système d'alertes et les actions courantes.

4.3.2.3. Contexte Technique

• Unité Données : les signaux perceptibles de la crise (visuels, sonores et olfactifs) et les alertes/informations transmises (quantité, qualité, ...).

Finalement, ici, chaque facteur de comportement est associé à une unité de contexte (une unité pouvant regrouper plusieurs facteurs de comportement).

4.3.3. Modélisation

Dans le cadre de la gestion de crise, modéliser le contexte en intégrant nos facteurs donne la possibilité de prendre en compte les réactions humaines. Avec un tel modèle, il est possible de tester des hypothèses sur la participation des différents facteurs aux comportements de crise pour une population donnée. Il est également possible de distinguer l'importance des différents facteurs impliqués pour différents types de crise. Les résultats obtenus à partir de ces analyses pourraient aider à la prévention et à la préparation des programmes de gestion des crises et aider à apporter des modèles qui fournissent des prédictions en temps réel. Une combinaison de différents modèles semble nécessaire à la fois pour représenter les définitions et la complexité de l'interaction, et pour pouvoir agréger les données pour offrir une vision synthétique des hypothèses validées. Beaucoup d'approches proposent déjà des combinaisons de plusieurs modèles (Lin, Lee, 1991 ; Ding *et al.*, 2006 ; Swat *et al.*, 2016). Nous avons retenu trois choix qui peuvent intégrer nos facteurs de comportement dans des objectifs d'analyse mathématique ou sémantique : les modèles attribut-valeur, les ontologies et les modèles prédictifs dans (Arru, Negre, 2017).

A première vue, une combinaison de ces trois modèles, combinant connaissance, expertise, expérience et technologie / ingénierie statistique, pourrait être envisagée, permettant de bénéficier de chacune, obtenant ainsi la modélisation la plus complète et aussi la plus réaliste possible.

Dans (Arru, Negre, 2017), les vingt facteurs de comportement ont été organisés selon deux classes : les facteurs liés à l'individu et ceux liés à l'environnement (comme illustré sur la figure 4a). Passer à notre typologie (comme illustré sur la figure 4b) permettrait de modéliser les facteurs à différents niveaux et ainsi permettre des analyses plus poussées. Par exemple, des analyses des comportements par famille/unité de contexte (trois familles et dix unités) seraient peut-être plus pertinentes qu'une analyse essayant de corréler les vingt facteurs ensembles ou selon deux classes.

Comportements	Facteurs liés à l'individu	Etat civil
		Personnalité
		Motivation à s'enfuir/se défendre
		Émotions
		Signaux physiologiques
		Responsabilité
		Expérience
		Connaissances explicitées
		Évaluation du risque
		Perception du système d'alerte
		Action courante
	Facteurs liés à l'environnement	Caractéristiques de la période
		Phase temporelle de la crise
		Caractéristiques de la zone géographique
		Capacité d'interaction
		Caractéristiques de l'entourage
		Comportement des personnes les plus proches
		Comportement global de l'entourage
		Signaux perceptibles de la crise
		Alertes / informations transmises

(a) Classification de (Arru, Negre, 2017)

	Famille	*Unité*	*Facteur*
Contexte - Comportement	Contexte physique	Temporelle	Caractéristiques de la période
			Phase temporelle de la crise
		Spatiale	Caractéristiques de la zone géographique
		Environnementale	Caractéristiques de la zone géographique
		Équipement	Capacité d'interaction
	Contexte personnel	Démographique	Etat civil
		Psychophysiologique	Personnalité
			Motivation à s'enfuir/se défendre
			Émotions
			Signaux physiologiques
		Sociale	Responsabilité
			Caractéristiques de l'entourage
			Comportement des personnes les plus proches
			Comportement global de l'entourage
		Cognitive	Expérience
			Connaissances explicitées
			Évaluation du risque
			Perception du système d'alerte
			Action courante
	Contexte technique	Données	Signaux perceptibles de la crise
			Alertes / informations transmises

(b) Les facteurs de comportement vus comme des facteurs de contexte selon notre typologie

Figure 4. Facteurs de comportement.

5. Conclusion

Dans cet article, nous proposons une typologie de contexte pour les systèmes de recommandation afin d'être au plus près des besoins de l'utilisateur et améliorer les recommandations. Cette typologie de contexte est présentée selon trois familles et dix unités de contexte. Elle a été instanciée en OLAP, dans un environnement commercial (de ventes de véhicules d'occasion) et en gestion de crise (via les comportements humains).

Le tableau 2 récapitule les informations contextuelles détectées comme pertinentes pour chaque cas d'application.

Nos propositions montrent ainsi quelles informations contextuelles peuvent être prises en compte dans le processus de décision et comment les modéliser en vue de leur intégration dans le système de recommandation. Ces données/sources externes permettront de mieux connaître l'utilisateur, d'enrichir son profil, d'améliorer la pertinence des recommandations, de densifier les données d'entrée avec des données complémentaires et ainsi d'améliorer les recommandations.

Tableau 2. Récapitulatif des informations contextuelles pertinentes selon les domaines/cas d'applications

Contexte	Unité	OLAP	Commerce (véhicules d'occasion)	Gestion de crise (comportements)
Physique	Temporelle	✓		✓
	Spatiale		✓	✓
	Environnementale		✓	✓
	Équipement		✓	✓
Personnel	Démographique	✓	✓	✓
	Psychophysiologique		✓	✓
	Sociale	✓	✓	✓
	Cognitive	✓	✓	✓
Technique	Matériel	✓		
	Données			✓

Pouvoir utiliser notre typologie [7] dans trois domaines différents, sans compromettre la modélisation du contexte, valide sa généricité et son applicabilité.

Par ailleurs, de nouvelles idées commencent à émerger dans le domaine des systèmes de recommandation (contextuels). En voici quelques exemples :

– (Pagano *et al.*, 2016) ont récemment initié l'idée selon laquelle la prochaine évolution des systèmes de recommandation serait de passer d'un système de recommandation contextuel (*context-aware*) qui s'adapte au contexte à un système de recommandation piloté par le contexte (*context-driven*). L'idée est de ne plus prendre en compte le contexte comme une information complémentaire, mais de le faire passer à l'avant plan, c'est-à-dire faire de la recommandation en se basant sur ce qui se passe autour de l'utilisateur à l'instant donné (la situation), et ce qu'il est en train d'accomplir (l'intention), au lieu de se baser sur son comportement dans le passé. Cet axe de recherche soulève de nombreux défis liés notamment à la modélisation et à l'intégration d'information (contextuelle) continue/temps réel sans endommager sa valeur, et à la sérialité des éléments (il existe une dépendance séquentielle entre les éléments puisque le contexte est une information continue).

– La prolifération des sites de commerce électronique, des médias sociaux, ... a permis aux utilisateurs de fournir des commentaires, d'exprimer leurs préférences/intérêts et de maintenir des profils utilisateurs dans de multiples systèmes, reflétant la variété de leurs goûts/intérêts. Tirer parti de toutes ces informations disponibles dans différents systèmes et relatives à différents champs/spécialités peut être bénéfique pour générer des profils utilisateurs plus complets et de meilleures recommandations, par exemple, en proposant des recommandations personnalisées « croisées » pour des éléments de champs différents. Les systèmes de recommandation multi-domaines (*cross-domain recommender systems*) visent à générer ou à améliorer des recommandations pour un champs particulier en exploitant les profils utilisateurs (ou toutes autres données/informations) issues d'autres champs (Ricci *et al.*, 2011).

7. Il est à noter que notre typologie est actuellement "hors sol". Nos travaux futurs viseront à étudier la possibilité de la rattacher à des ontologies comme celles de (Wang *et al.*, 2004) ou (Schmidt, 2006)

Cet axe de recherche émergent soulève de nombreux défis comme la définition de la tâche/stratégie de tels systèmes ou encore les techniques/modèles appropriés pour le transfert des données/informations/connaissances d'un champs à un autre, ... (Khan *et al*., 2017). Notre proposition de typologie de contexte est un pas en ce sens.

Bibliographie

Adomavicius G., Tuzhilin A. (2011). Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), *Recommender systems handbook*, p. 217–253. Boston, MA, Springer US.

Akermi I., Boughanem M., Faiz R. (2015). Une approche de recommandation proactive dans un environnement mobile. In *Inforsid*, p. 301-316.

Arduin P., Mayag B., Negre E., Rosenthal-Sabroux C. (2014). How to compromise on the best price? A group tacit knowledge-based multicriteria approach. *Journal of Decision Systems*, vol. 23, n° 1, p. 99–112.

Arru M., Negre E. (2017). People behaviors in crisis situations: Three modeling propositions. In *Information systems for crisis response and management - iscram, albi, may 2017.*

Baltrunas L., Ludwig B., Peer S., Ricci F. (2012, juin). Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, vol. 16, n° 5, p. 507–526.

Bazire M., Brézillon P. (2005). Understanding context before using it. In *Proceedings of the 5th international conference on modeling and using context*, p. 29–40. Berlin, Heidelberg, Springer-Verlag.

Benouaret I. (2015). Un système de recommandation sensible au contexte pour la visite de musée. In *CORIA 2015 - conférence en recherche d'infomations et applications - 12th french information retrieval conference, paris, france, march 18-20, 2015.*, p. 515–524.

Bettini C., Brdiczka O., Henricksen K., Indulska J., Nicklas D., Ranganathan A. *et al*. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, vol. 6, n° 2, p. 161 - 180. (Context Modelling, Reasoning and Management)

Brown P. J., Bovey J. D., Chen X. (1997). Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Commun.*, vol. 4, n° 5, p. 58-64.

Burke R. D., Hammond K. J., Young B. C. (1997). The findme approach to assisted browsing. *IEEE Expert*, vol. 12, p. 32–40.

Dey A., Salber D., Abowd G. (2001). *A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.*

Ding Z., Peng Y., Pan R. (2006). *A bayesian approach to uncertainty modelling in owl ontology*. Rapport technique. DTIC Document.

Ferdousi Z. V., Negre E., Colazzo D. (2017). Context factors in context-aware recommender systems. In *Atelier interdisciplinaire sur les systèmes de recommandation.*

Henricksen K., Indulska J. (2006, février). Developing context-aware pervasive computing applications: Models and approach. *Pervasive Mob. Comput.*, vol. 2, n° 1, p. 37–64.

Ingwersen P., Järvelin K. (2005). *The turn: Integration of information seeking and retrieval in context (the information retrieval series)*. Secaucus, NJ, USA, Springer-Verlag New York, Inc.

Khan M. M., Ibrahim R., Ghani I. (2017, juin). Cross domain recommender systems: A systematic literature review. *ACM Comput. Surv.*, vol. 50, n° 3, p. 36:1–36:34.

Lin C.-T., Lee C. S. G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on computers*, vol. 40, n° 12, p. 1320–1336.

Mostefaoui G. K., Pasquier-Rocha J., Brézillon P. (2004). Context-aware computing: A guide for the pervasive computing community. In *Icps*, p. 39-48. IEEE Computer Society.

Negre E. (2017). Prise en compte du contexte dans les systèmes de recommandation de requêtes olap. In *Actes des journées francophones sur les entrepôts de données et l'analyse en ligne, EDA 2017, lyon, france.*

Nguyen C. (2010). *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario*. Thèse de doctorat non publiée, Ecole Nationale Supérieure des Télécommunications de Bretagne.

Pagano R., Cremonesi P., Larson M., Hidasi B., Tikk D., Karatzoglou A. *et al*. (2016). The contextual turn: From context-aware to context-driven recommender systems. In *Proceedings of the 10th acm conference on recommender systems*, p. 249–252.

Petrelli D., Not E., Strapparava C., Stock O., Zancanaro M. (2000). Modeling context i s like taking pictures. In *Conference on human factors in computers, workshop "the what, who, where, when, why and how of context-awareness"*.

Riboni D., Bettini C. (2011, mars). Cosar: Hybrid reasoning for context-aware activity recognition. *Personal Ubiquitous Comput.*, vol. 15, n° 3, p. 271–289.

Ricci F., Rokach L., Shapira B., Kantor P. B. (Eds.). (2011). *Recommender systems handbook*. Springer.

Schilit B., Theimer M. (1994, sep/oct). Disseminating active map information to mobile hosts. *Network, IEEE*, vol. 8, n° 5, p. 22 -32.

Schmidt A. (2006). Ontology-based user context management: The challenges of imperfection and time-dependence. In R. Meersman, Z. Tari (Eds.), *On the move to meaningful internet systems 2006: Coopis, doa, gada, and odbase*, p. 995–1011. Springer Berlin Heidelberg.

Sillamy N. (1983). *Dictionnaire usuel de psychologie*. Bordas.

Soualah Alila F. (2015). *Camlearn* : une architecture de système de recommandation sémantique sensible au contexte : application au domaine du m-learning*. Thèse de doctorat non publiée. (Dijon)

Swat M. J., Grenon P., Wimalaratne S. (2016). Probonto–ontology and knowledge base of probability distributions. *Bioinformatics*.

Wang X. H., Zhang D. Q., Gu T., Pung H. K. (2004). Ontology based context modeling and reasoning using owl. In *Proceedings of the second ieee annual conference on pervasive computing and communications workshops*, p. 18–. IEEE Computer Society.

Composition Sémantique et Dynamique à base d'Agents des Services Cloud pour ERP

Hamza Reffad[1], Adel Alti [1], Philippe Roose [2]

1. *LRSD/Département d'informatique*
Université Farhat Abbas de Sétif, DZ-19000 Sétif
reffadh@yahoo.fr ;alti.adel @.univ-setif.dz

2. *LIUPPA – T2i*
IUT de Bayonne – Pays Basque, 64600, Anglet – France
Philippe.Roose@iutbayonne.univ-pau.fr

RESUME. Actuellement, la technologie Cloud est largement adoptée par les entreprises pour développer des solutions informatiques de qualité. En effet, les Petites et les Moyennes Entreprises (PME) sont à la recherche d'ERP « sur mesure » afin d'automatiser leurs activités commerciales. La complexité de la tâche de sélection et de composition de services augmente selon les évolutions des différents besoins fonctionnels et non fonctionnels des PME (contraintes et préférences). La plupart des systèmes ERP cloud existants (SAP, Oracle ERP cloud, etc.) ne sont pas suffisamment flexibles pour prendre en charge l'autoadaptation des processus d'affaire ERP. Cet article présente une nouvelle approche sémantique et dynamique à base d'agents pour la composition de services cloud afin d'obtenir un processus d'affaire ERP personnalisé. Une ontologie est proposée pour la description sémantique et la gestion du processus de construction ERP. Elle génère un processus d'affaire ERP virtuel selon les besoins fonctionnels du PME. Un algorithme en deux phases pour la composition des services cloud est proposé pour obtenir un processus d'affaire ERP optimal, concret et personnalisé. Les résultats expérimentaux montrent l'efficacité de l'approche proposé.

ABSTRACT. Nowadays, cloud technology is widely adopted by companies to develop quality computing solutions. Indeed, Small and Medium Enterprises (SMEs) are looking for the best customized ERP to automate their business activities. The complexity of the task of selection and composition of services increases with changes in the different functional and non-functional needs of SMEs (constraints and preferences). Most existing cloud ERP systems (SAP, Oracle, etc.) are not flexible enough to support ERP business process auto-adaptation. This article presents a new semantic and dynamic agent-based approach to cloud service composition to obtain a personalized ERP business process. A new ontology is proposed for the semantic description and management of the ERP construction process. It generates a virtual ERP business process according to the SME functional needs. A two-stage algorithm for the cloud services composition is proposed to obtain an optimal personalized concrete ERP business process. Experimental results show the effectiveness of the proposed approach.

Mots-clés : Cloud, optimisation, ontologie, service, NSGA-II, violation des contraintes.

KEYWORDS: Cloud; optimization, ontology, service, NSGA-II algorithm, constraints violation.

1. Introduction

Un logiciel Enterprise Resource Planning (ERP) est un outil qui permet le pilotage de l'entreprise. Il embarque, en un même logiciel et une seule base de données ainsi que les fonctionnalités nécessaires à la gestion de l'ensemble de l'activité d'une entreprise : gestion comptable, gestion commerciale, etc.

Vue la complexité et le cout élevé de ces ERP, les Petites et les Moyennes Entreprises (PME) recherchent un ERP sur mesure en tenant compte des évolutions de leurs activités. Avec la prolifération du Cloud Computing, les grands fournisseurs de systèmes ERP tels que SAP, Sage et Microsoft positionnent leurs offres ERP sous forme de modèle SaaS (Johansson et Ruivo, 2013). Néanmoins, ces systèmes ne sont pas suffisamment flexibles pour supporter les évolutions des besoins des entreprises (Li, 2009). La tendance vers une approche de composition de services cloud pour avoir un Processus d'Affaire (PA) offre deux avantages : la facilité d'intégration et la réduction des coûts (Tarantilis *et al.*, 2008). Ainsi, la disponibilité d'un grand nombre des services cloud hétérogènes avec différentes QoS sont offerts par plusieurs fournisseurs de services Cloud. Les développeurs ont tiré parti de ces services pour fournir un PA répondant aux besoins fonctionnels et non fonctionnels spécifiques des clients. Plusieurs méthodes d'optimisation de composition des services ont été proposées afin d'optimiser les paramètres de QoS (Sasikaladevi et Arockiam, 2012 ; Yu *et al.*, 2015 ; Asghari et Navimipour, 2016). Cependant, ces mécanismes ne tiennent pas compte des évolutions des contraintes et préférences du client. En plus, ils ne gèrent pas de manière efficace et flexible un nombre large de services hétérogènes. Cette hétérogénéité implique une dégradation de qualité de contrôle dans la sélection et la composition des services (Chang *et al.*, 2014).

Les clients souhaitent que les applications ERP puissent être personnalisées automatiquement en fonction de leurs besoins fonctionnels actuels, de leurs contraintes et préférences. Un filtrage sémantique permet d'améliorer la sélection des services en pénalisant les services qui violent les contraintes des clients. Cependant, le nombre des services Cloud et la qualité de service d'un fournisseur de Cloud n'est pas statique et peut évoluer dans le temps. Face aux besoins évolutifs des entreprises ainsi qu'à l'augmentation des services Cloud offrant différentes QoS, le développement des ERP personnalisés nécessite de proposer des solutions pertinentes qui répondent aux attentes des clients. Notre approche consiste à offrir au client un processus d'affaire ERP qui satisfait ses besoins fonctionnels en optimisant les QoS selon ses contraintes et préférences contextuelles.

Les contributions de ce travail sont : (1) l'extension de l'ontologie définit dans (Reffad et al., 2016) par l'inclusion de la description sémantique des contraintes et préférences contextuelles de l'entreprise cliente, (2) - développement d'un algorithme sémantique dynamique collaboratif en deux phases pour la composition des services Cloud. Dans la première phase, nous avons utilisé l'algorithme NSGA-II auquel nous avons ajouté une nouvelle relation de dominance basée sur une fonction de pénalité. Cette phase vise à sélectionner les services cloud composites pertinents qui respectent les contraintes des clients. La deuxième phase sélectionne un service cloud composite parmi les services engendrés par la première phase,

selon les préférences du client en utilisant la somme pondérée des QoS. À ce stade, nous avons proposé une nouvelle technique de calcul des poids de QoS.

Le reste de l'article est structuré comme suit : la section 2 détaille des travaux connexes. Dans la section 3, nous présentons notre approche en détails. Dans la section 4, des résultats expérimentaux sont présentés à l'aide des jeux de données aléatoires. Enfin, la section 5 conclut l'article en présentant quelques perspectives.

2. Travaux Connexes

Plusieurs approches ont été proposées pour l'optimisation de la composition des services Cloud. Les trois principales catégories d'approches sont les suivantes : approches basées sur les contraintes (Rosenberg *et al.,* 2009 ; Deng *et al.,* 2016), approches basées sur la classification sémantique (Alti *et al.,* 2014 ; Reffad *et al.,* 2016) et les approches basées sur les métaheuristiques (Sasikaladevi et Arockiam, 2012 ; Yu *et al.,* 2015 ;Chang *et al.,* 2015).

Pour les approches basées sur les contraintes, (Rosenberg *et al.,* 2009) ont proposé l'approche semi-automatique CaaS (Composition as a Service) combinée avec un langage spécifique pour le domaine nommé VCL (Vienna Composition Language). Ce langage est riche mais manque de descriptions sémantiques des services cloud hétérogènes. Ceci complique la tâche de décision au cours de la sélection des services. (Alti *et al.,* 2014) ont proposé une approche de composition sémantique automatique basée sur l'utilisation d'ontologies. Ils ont décrit une ontologie qui permet la génération automatique de l'assemblage de services hétérogènes de qualité. Ce travail ne tient pas compte de la croissance d'une variété de fournisseurs de services hétérogènes. Les approches basées sur les métaheuristiques distinguent deux principales catégories d'optimisation, pour la première, le problème multi-objectif peut être résolu en le réduisant à un problème mono-objectif via une technique de scalarisation. Nous pouvons distinguer les méthodes suivantes : recherche taboue (Pop et al., 2011), algorithme génétique (Sasikaladevi et Arockiam, 2012), optimisation des essaims de particules (Jun et Weihua, 2009), algorithme de concurrence impérialiste (Jula et al., 2011) et optimisation des colonies de fourmis (Yu *et al.,* 2015). Le principal inconvénient commun des méthodes de scalarisation est que la découverte du service composite avec le meilleur fitness peut pénaliser quelques QoS (i.e. si la fonction de fitness est la somme pondérée des critères QoS de sécurité et de temps de réponse, un service composite peut avoir le meilleur fitness avec une mauvaise sécurité). Pour le second, les approches d'optimisation multi-objectifs sont souvent utilisées lorsque deux objectifs conflictuels ou plus doivent être considérés simultanément afin de négocier un ensemble de solutions Pareto-optimales (NSGA-II, SPEA2 et E3-MOGA) (Huang *et al.,* 2012). Les approches Pareto-optimales génèrent un ensemble de solutions, mais dans de nombreuses situations, les clients n'ont besoin que d'une solution qui doit être sélectionnée automatiquement parmi les solutions résultantes. L'algorithme NSGAII est le plus populaire des algorithmes d'optimisation multi-objectif. Il permet d'atteindre le front Pareto-optimal par un minimum d'itérations.

Nous avons basé sur NSGA-II pour avoir un ensemble de solutions Pareto-optimal qui respecte les contraintes du client. Ensuite, une autre phase est exécutée pour sélectionner la solution finale selon les préférences du client.

Chen et al. (Chen *et al.*, 2015) ont défini une plate-forme CloudERP sur laquelle l'entreprise cliente pouvait personnaliser un système ERP entier correspondant à ses besoins sous forme d'un service composite. La méthode de composition de service proposée est basée sur un algorithme génétique avec la théorie 'rough set theory'.

Les travaux existants n'utilisent pas de mécanismes sémantiques au niveau du profil client pour satisfaire explicitement ses contraintes et ses préférences. De plus, ces travaux ne prennent pas la nature dynamique lié aux changements du contexte.

3. Approche dynamique collaboratif sémantique pour la composition des services cloud

3.1 Architecture générale

L'architecture générale de notre approche, illustrée à la figure 1, se base sur les agents durant la découverte, la sélection et la composition dynamique sémantique des services cloud afin d'obtenir un processus d'affaire ERP optimal selon les exigences du client. Elle se compose de trois entités :

– C**lient :** c'est l'entreprise qui a besoin d'un ERP personnalisé

– **Service Sémantique Proxy-Cloud :** il est constitué d'un agent planificateur ERP Virtual Composite Service (ERP-VCS), d'un agent maitre et des agents esclaves d'optimisation de service composite appelé ERP Cloud Composite Service (ERP–CCS-2S) et d'un gestionnaire du contexte.

- **Agent planificateur ERP-VCS :** est un moteur d'inférence sémantique chargé de générer le PA virtuel sous forme d'un Service Composite Virtuel (SCV) selon les besoins fonctionnels du client. Ce SVC se compose par des services virtuels (tâches). Chaque service virtuel (SV) est lié à un ensemble des services cloud concrets (SC) ayant les mêmes fonctionnalités avec différentes QoS.

- **Agent ERP–CCS-2S** : il prend en entrée les contraintes et les préférences du client décrites dans notre ontologie et la description sémantique des services Cloud (catégorie, rôle, paramètres d'entrées/sorties, paramètres de QoS et les spécifications de contexte), ensuite il produit un PA optimal sous forme d'un service cloud composite (SCC) en **fonction** des contraintes et préférences du client.

- **Agent de gestion sémantique du contexte** : le composant de gestion sémantique évalue dynamiquement le profil utilisateur (*besoins, contraintes, préférences*) et le contexte du service cloud (*connexion, taux de charge, services disponibles, etc.*) pour **sélectionner** les services cloud convenables à l'utilisateur. Chaque changement de contexte incite une réévaluation sémantique du contexte.

- **Le registre des services Cloud et profils clients** : contient la description sémantique des services cloud de tous les fournisseurs de services cloud et la description des profils des clients au sein de notre ontologie.

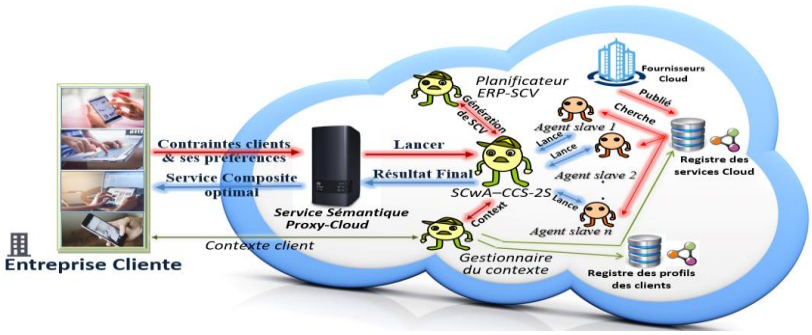


Figure 1. L'architecture générale de notre approche

3.2. Modèle ontologique

L'idée clé de l'ontologie proposée est l'assignement d'un niveau sémantique qui permet la description des services Cloud et des PAs et le filtrage sémantique des services. La figure 2 présente l'ontologie définie par les classes suivantes :

– **Client :** à un *nom*, des besoins fonctionnels, des contraintes et des préférences contextuelles. Un client spécifie ses besoins fonctionnels, ses contraintes et ses préférences visuellement à travers une interface utilisateur (GUI).

- Besoins fonctionnels : Les besoins fonctionnels peut être exprimé par des clauses 'AND' combinant des contraintes fonctionnelles simples. *Exemple*: (TaskCategory = `Buy' AND TransactionType=`Internet' AND TaskInput = `ProductDetails' AND TaskOutput =`Receipt`). Le moteur d'inférence génère un PA virtuel optimal sous forme d'une composition de services virtuels.

- Contraintes : Le client spécifie ses contraintes à partir d'une interface utilisateur. Il sélectionne une contrainte pour chaque QoS (temps de réponse : rapide, prix : moins cher, etc.). Pour unifier les valeurs de ces contraintes, elles sont transformées en valeurs sémantiques (*faible, moyenne, haute*). Ensuite, ces valeurs sémantiques sont mappées sur des intervalles voisins $\{I_1, I_2, ..., I_{nb}\}$ tels que $I_i \in [0,1]$. Finalement, la $i^{ème}$ contrainte sémantique de l'attribut q_i (notée C_i) est convertie en valeur quantitative :

$$C_i = \text{qmin}_i^{\text{semantic\_value}} \tag{1}$$

- Préférences : les préférences de QoS sont spécifiées par le client sous forme d'une liste ordonnée. Cet ordre attribue un nombre à chaque critère de QoS appelé explicitPriority. Ensuite, un poids w_i est assigné à chaque critère de QoS en fonction de son importance. Ce poids est calculé par l'équation 2.

$$w_i = \frac{e^{P_i}}{\sum_{j=1}^{n} e^{P_j}} \tag{2}$$

$$P_i = explicitPriority_i + constraintRank_i \times n \tag{3}$$

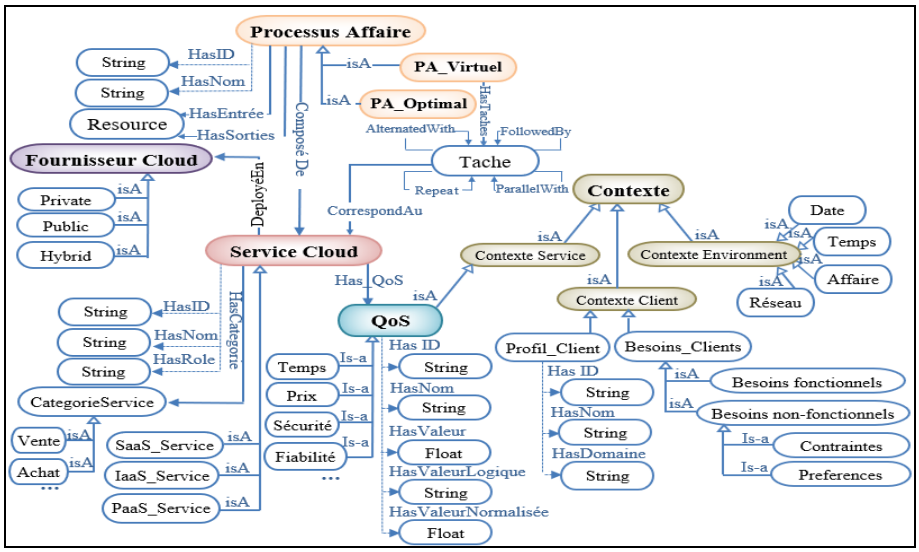


Figure 2. Ontologie générique d'un processus d'affaire pour ERP.

Où : n est le nombre de QoSs ; *explicitPriority* est un nombre entre 1 et *n* indiquant la priorité explicite d'une QoS. Si une préférence n'est pas spécifiée, son *explicitPriority est nulle* ; *constraintRank* est un nombre entier qui indique l'importance de la contrainte de QoS (0 : *aucun,* 1 : *faible,* 2 : *moyen* and 3 : *haut*).

– **Service :** L'ontologie couvre les services Cloud d'affaires, tels que l'achat, vente, finance, etc. Chaque service est étoffé par un ensemble d'attributs de QoS.

– **Processus d'affaire :** cette classe décrit tous les processus d'affaires en termes de séquences de tâches. Chaque tâche regroupe des services ayant une même fonctionnalité avec des QoS différentes.

– **Contexte :** le contexte est tout type d'informations qui peut être collecté à partir du service (*nom, version, besoins en ressources*), de l'environnement (*heure, lieu*) et de l'utilisateur (*contraintes et préférences*).

3.3 *Algorithme détaillé*

Le but de notre travail est de proposer une approche afin de satisfaire aux besoins fonctionnels du client et d'optimiser ses contraintes et ses préférences évolutives. L'approche proposée est implémentée par une ontologie générique contextuelle cloud qui implémente le concept des Agents et NSGA-II. Pour notre travail, nous avons retenu l'algorithme NSGA-II qui est connu comme très performant et efficace dans le domaine de l'optimisation multi-objectif (Huang *et al.*, 2012). Nous avons inclus l'approche multi-agents qui permet d'améliorer le temps d'exécution à cause de son aspect parallèle. L'objectif principal est la

sélection du service cloud composite optimal en termes de QoS avec un minimum dégrée de violation de contraintes du client. Il se base sur des services sémantiques concrets disponibles enregistrés dans un registre de l'ontologie. La capacité des agents de supervision de contexte est exploitée pour réagir aux changements de contexte des services et/ou aux changements des préférences des clients. L'utilisation des solutions trouvées permet d'exploiter l'expérience de recherche acquise par les agents de construction de solutions dans les itérations futures de l'algorithme. Les étapes de l'algorithme sont :

– **Etape1 : Génération automatique et sémantique du processus virtuel :** le modèle de PA global est généré automatiquement en utilisant un algorithme de chainage arrière et les liens sémantiques entre les SVs (Alti *et al.,* 2014) selon les besoins fonctionnels des clients. Il est optimisé en se basant sur la réputation de chaque service virtuel pour éviter la redondance de services.

– **Etape 2 (phase 1) : Optimisation parallèle multi-objectifs au niveau de chaque agent esclave :** dans cette étape chaque agent esclave a comme objectif de vérifier la satisfaction des contraintes clients afin de sélectionner le meilleur service cloud composite (SCC). L'optimisation est effectuée en adoptant le (NSGA-II) guidée par une fonction de pénalité afin de sélectionner le meilleur service composite en termes de QoS en respectant les contraintes du client.

1. **Codage des solutions :** Un chromosome ou individu représente un service composite (PA) sous forme d'une chaine de services Cloud (gènes).
2. **Evaluation** : On évalue chaque SCC par l'agrégation des valeurs de QoS normalisées des SC.
3. **Sélection** : Dans notre approche on a on a assigné un degré de violation pour chaque attribut de QoS pour pénaliser les services qui ne respectent pas les contraintes du client. Ce degré est calculé comme suit :

$$\deg_{v\_CCS} = \sum_i w_i \times \deg^{i}_{v\_ccs} \tag{4}$$

$$\text{Avec } \deg^{i}_{v\_ccs} = \begin{cases} C_i - q_i & \text{If } C_i > q_i \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

Où : w_j : est le poids du ième attribut de QoS ; q_i : est la valeur agrégée du ième attribut de QoS ; C_i : est la valeur de contrainte du client du ième attribut de QoS ;

La nouvelle relation de dominance considère que le SCC dominant est celui qui a le plus faible degré de violation de contrainte. Si les degrés de violation de deux SCC sont égaux, la relation de dominance classique de NSGA-II est appliquée.

– **Etape 3 (phase 1) : Fusion et resélections des Pareto-optimaux au niveau d'agent maître SCwA–CCS-2S.** A la fin de toutes les itérations les résultats obtenus par les agents esclaves sont fusionnés et ordonnées. Ensuite, les meilleurs services composites (Pareto-optimal global) sont sélectionnés.

– **Etape 4 (phase 2) : Sélection d'une solution optimale.** Cette étape sélectionne une solution unique (SCC) à partir de l'ensemble du Pareto-optimal (PCCS) obtenu à la première phase. Le score de chaque solution est calculé par l'équation 6. La solution avec le meilleur score (maximum) est sélectionnée.

Où : nb est le nombre de paramètre de QoS ; w_j est le poids du j^{ème} attribut de QoS ; q'_j est la valeur agrégée du j^{ème} attribut de QoS ;

Le gestionnaire de contexte évalue en permanence le contexte du client (les préférences et contraintes du client) pour fournir une solution SCC en conséquence. En cas de modification des préférences du client, la solution est prise directement à partir des solutions du Pareto-optimal actuel sans devoir relancer la phase 1.

4. Résultats d'expérimentations

Nous avons évalué la satisfaction des contraintes et préférences du client, plusieurs expérimentations ont été réalisées sur un PC Intel (R) 4.0 GHz, 4 Go de RAM, Windows 7 (32 bits) et NetBeans. Le processus d'affaire étudié contient 10 services virtuels et chaque service virtuel est lié à un ensemble de candidats comprenant 100 services cloud. Le client choisi ses contraintes comme suit : Prix : *moins cher*, Temps de réponse : *rapide ;* qui se transforment en valeur sémantique « Haut ». La figure 3 montre clairement que tous les SCCs de l'ensemble du Pareto-optimal respectent les contraintes du client. En d'autres termes, toutes les valeurs des attributs de QoS de tous les SCCs du Pareto-optimal appartiennent à l'intervalle [0.7, 1]. Elle montre aussi la solution finale à fournir au client selon les trois différentes préférences du client. Afin de justifier l'utilisation de l'aspect multi-agents, des tests de comparaison sont effectuées entre l'approche SCwA-CCS (multi-agents) avec l'approche SCw-CCS (sans agents).

La figure 4 montre la performance de l'aspect multi-agents en termes de temps d'exécution. En particulier lorsque le nombre d'itérations augmente.

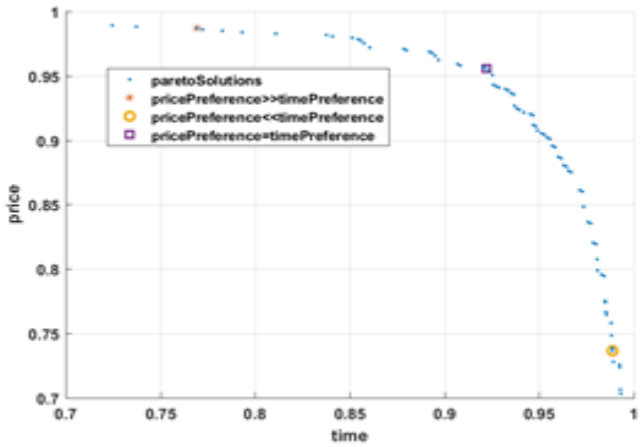


Figure 3. Pareto-optimal et solutions finales selon les différentes préférences du client (contraintes : temps ϵ [0.7, 1]et prix ϵ [0.7, 1]).

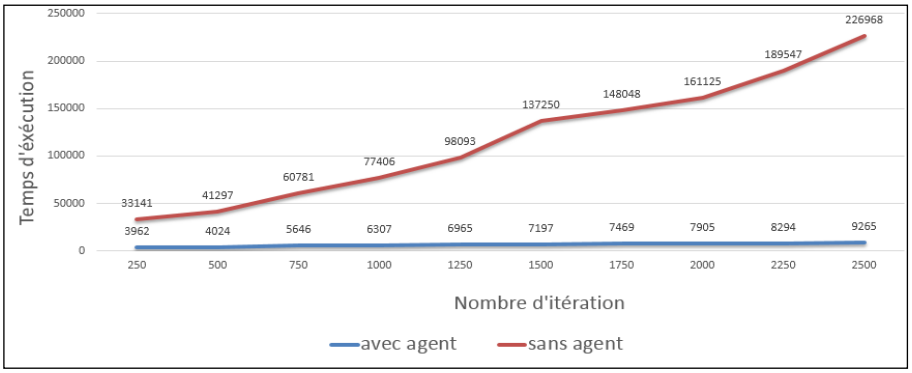


Figure 3. Temps d'exécution de SCw–CCS vs SCwA–CCS.

5. Conclusion

Cet article présente une nouvelle approche sémantique et dynamique à base d'agents afin de composer des services Cloud pour obtenir un ERP personnalisé. En effet, nous avons proposé une ontologie afin de guider et d'adapter le processus de composition dynamique. Cette ontologie est basée sur le potentiel des règles d'inférences pour créer des processus d'affaires virtuels satisfaisant les besoins fonctionnels du client. Ainsi, un algorithme à deux phases est proposé pour l'optimisation du processus d'affaire en termes de QoS. La première phase consiste à engendrer un ensemble de SCCs (Pareto-optimal) qui respectent les contraintes des clients. Cette phase utilise NSGA-II à base d'agents en introduisant un nouveau critère de dominance basé sur la violation des contraintes. La deuxième phase sert à sélectionner le SCC parmi les SCCs obtenus à partir de la première phase afin de le fournir au client selon ses préférences. Dans cette phase, on a utilisé la somme pondérée des attributs de QoS pour évaluer par score les différents SCCs du Pareto-optimal afin de sélectionner le SCC ayant le meilleur (maximum) score. Une nouvelle technique est proposée pour calculer les poids des attributs de QoS. Les résultats de l'expérimentation montrent l'efficacité de l'approche proposé en termes des contraintes et préférences du client, ainsi que le facteur multi-agents améliore clairement le temps d'exécution du système. Dans ce travail nous avons utilisé des valeurs agrégées de QoS déterministes (i.e. la moyenne). Dans les futurs travaux, nous envisageons d'étendre ce travail par l'intégration de l'aspect incertain dans la mesure des QoS, ce qui rend les résultats plus précis.

Bibliographie

Alti A., Laborie S., Roose, P. (2014). Dynamic semantic- based adaptation of multimedia documents. *Transactions on Emerging Telecommunications Technologies*, vol. 25, n° 2, p. 239-258.

Asghari S., Navimipour N. J. (2016). Review and Comparison of Meta-Heuristic Algorithms for Service Composition in Cloud Computing, Majlesi, *Journal of Multimedia Processing*, vol. 4, 2016.

Chang H., Liu H., Leung Y.W., Chu X. (2014). Minimum latency server selection for heterogeneous cloud services. In IEEE Global Communications Conference (GLOBECOM), p. 2276-2282, December 2014.

Chen C.S., Liang W.Y , Hsu H.Y. (2015). A cloud computing platform for ERP applications. *Applied Soft Computing* , vol. 274, p. 127-136.

Deng S., Huang L., Wu H., Wu, Z. (2016). Constraints-Driven Service Composition in Mobile Cloud Computing., *IEEE International Conference on Web Services (IEE ICWS 2016)*, San Francisco, CA, USA, June 27 - July 2, 2016. p. 228-235.

Huang, X., Lei, X., & Jiang, Y. (2012). Comparison of Three Multi-Objective Optimization Algorithms for Hydrological Model. *In Computational Intelligence and Intelligent Systems*, p. 209-216, Springer, Berlin, Heidelberg.2012.

Johansson B., Ruivo, P. (2013). Exploring factors for adopting ERP as SaaS. *Procedia Technology*, vol. 9, p. 94-99.

Jula A., Zalinda, O., Sundararajan, E. (2013). A hybrid imperialist competitive gravitational attraction search algorithm to optimize cloud service composition. *In IEEE Workshop Memetic Computing (MC)*, p. 37–43.

Jun L., Weihua G. (2009). An environment-aware particle swarm optimization algorithm for services composition. International Conference on Computational Intelligence and Software Engineering, p. 1-4.

Li B., Li M. (2009). Research and design on the refinery ERP and eERP based on SOA and the component-oriented technology. *IEEE International Conference on Networking and Digital Society, ICNDS'09*., vol. 1, p. 85-88., 2009.

Reffad H., Alti A., Roose P. (2016). Cloud-based Semantic Platform for Dynamic Management of Context-aware mobile ERP applications. ACM MEDES International Conference – DOI: 10.1145/3012071.3012076, 2016, Hendaye, France.

Rosenberg F., Leitner P., Michlmayr A., Celikovic P., Dustdar, S. (2009). Towards Composition as a Service -A Quality of Service Driven Approach, *IEEE 25th International Conference on Data Engineering(ICDE)*, p. 1733 -1740

Sasikaladevi N., Arockiam L. (2012). Genetic approach for service selection in composite web service. *International Journal of Computer Applications*, vol. 44, n° 4, p. 22-29.

Tarantilis C.D., Kiranoudis C.T., Theodorakopoulos N.D. (2008). A web-based ERP system for business services and supply chain management: Application to real-world process scheduling. European Journal of Operational Research, vol.187, n° 4, p.1310-1326.

Pop, C.B., Vlad, M., Chifu, V.R., Salomie, I., Dinsoreanu, M. (2011). A tabu search optimization approach for semantic web service composition. In 10th IEEE International Symposium Parallel and Distributed Computing (ISPDC'2011), p. 274-277.

Yu Q., Chen L., Li B. (2015). Ant colony optimization applied to web service compositions in cloud computing. Computers & Electrical Engineering, vol.41, p.18-27.

Méta modèle de la sécurité des systèmes d'information

Enrichissement par le contexte

Jacky Akoka[1,2], Nabil Laoufi[3], Nadira Lammari[1]

1. CEDRIC-CNAM
292 Rue Saint-Martin, 75003 Paris, France
lammari@cnam.fr, jacky.akoka@lecnam.net

2. Institut Mines Télécom- TEM
9 rue Charles Fourier, 91011 Evry, France
jacky.akoka@telecom-em.eu

3. Ecole militaire polytechnique
BP 17, Bordj el Bahri ,16111, Alger, Algérie
nabil.laoufi@gmail.com

RESUME. Les entreprises sont confrontées de plus en plus aux problèmes induits par leur dépendance vis-à-vis des systèmes d'information. Elles se voient ainsi contraintes à mettre en œuvre un processus de dérivation des exigences de sécurité à partir de l'analyse des risques encourus. Ce processus requiert au préalable une analyse approfondie du contexte organisationnel. Le but de cet article est de proposer un méta modèle de sécurité enrichi par une ontologie du contexte. A cette fin, nous proposons (i) le développement d'une ontologie du contexte fondée sur la norme de sécurité ISO/CEI 27000 : 2018, (ii) une démarche d'enrichissement du méta modèle de sécurité par l'ontologie du contexte. Cet enrichissement est réalisé en deux phases. La première est relative à l'identification et à l'extraction des éléments du contexte de l'entreprise. La seconde concerne la détermination des critères de sécurité des actifs de l'organisation à protéger et (iii) l'application à un cas réel qui sert aussi de première étape dans la validation de notre démarche.

Mots-clés : Systèmes d'information, sécurité, ontologie, contexte, actifs, méta modèle

ABSTRACT. Companies are increasingly confronted with the problems caused by their reliance on information systems. They are thus forced to implement a process of security requirements derivation starting from risks analysis. This process requires a thorough analysis of the organizational context. The purpose of this article is to propose a security meta model enriched by an ontology of the context. To this end, we propose (i) the development of a context ontology based on the ISO / IEC 27000: 2018 security standard, (ii) an approach to enrich the security meta model with context ontology. This enrichment is carried out in two phases. The first is related to the identification and extraction of elements of the context of the enterprise. The second concerns the determination of the security criteria of the assets of the

organization to be protected and (iii) the application to a real case which also serves as a first step in the validation of our approach.

KEYWORDS: Information systems, security, ontology, context, assets, meta model.

1. Introduction

Les systèmes d'information (SI) doivent faire face à de nombreuses menaces susceptibles d'exploiter leurs vulnérabilités. Le but d'une politique de la sécurité est de limiter les impacts résultant de ces vulnérabilités. Cette politique est fondée sur la capacité des organisations à mettre en œuvre une procédure de dérivation des exigences de sécurité fondée sur l'analyse des risques qui ciblent le système d'information. Plusieurs méthodes permettent de dériver les exigences de sécurité à partir de l'analyse des risques (Lammari et al., 2011 ; Laoufi, 2017; Vasquez et al., 2012). Toutefois, la plupart n'intègre pas les contextes interne et externe des organisations.

Le concept de « contexte » ne fait pas l'objet d'une définition unanime. Cela est sans doute dû au fait qu'il n'existe pas un contexte déterminé par avance. (IFIP-IFAC Task Force, 2003) indique que « le contexte dépend des conditions interdépendantes dans lesquelles un événement, une action, etc. a lieu ». La définition la plus répandue est due à (Abowd et al., 1999) qui précisent que : « le contexte représente toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un objet ou un endroit considéré comme pertinent pour l'interaction entre un utilisateur et une application, y compris l'application et l'utilisateur ». En d'autres termes, le contexte peut être décrit comme étant constitué d'un ensemble d'attributs ayant un lien avec une finalité pour laquelle le contexte est utilisé.

Les linguistes et les chercheurs en langage naturel utilisent le concept de contexte pour interpréter le sens des phrases. Par exemple, la phrase « Je tiens à jouer avec ma sœur », indique que ma sœur et moi jouons ensemble et non qu'elle est un jouet. Autrement dit, le contexte social rétrécit l'interprétation correcte d'une expression (Leech, 1981).

Le contexte peut servir à limiter l'espace des solutions pour des problèmes liés au raisonnement automatique (Brézillon and Abu-Hakima, 1995). À titre d'exemple, les moteurs de recherche Web filtrent l'information en estimant sa pertinence dans des contextes déterminés d'interprétation, telle que la popularité des pages (Yahoo, Google), les catégories (recherches), les zones géographiques (France), etc.

En informatique, le contexte est généralement lié aux conditions dans lesquelles les utilisateurs sont immergés (poste de travail, réseau, communication, bande passante, sécurité). Dans le domaine de la sécurité des informations, le contexte joue un rôle primordial, qu'il soit un contexte interne ou un contexte externe. Le risque et les mesures de protection changent d'après le contexte. Ainsi, on ne protège pas un serveur de banque de la même façon qu'un simple serveur de messagerie d'une petite entreprise.

Le but de cet article est de proposer un méta modèle de la sécurité qui intègre le contexte des organisations. Le méta modèle de base repose sur des modèles ontologiques fondés sur les concepts relatifs à l'analyse des risques et à la dérivation des exigences de sécurité. Notre proposition permet l'enrichissement de ce méta modèle grâce à une ontologie du contexte. À cette fin, nous avons analysé les modèles de contexte existants. Les concepts sous-jacents permettent de peupler l'ontologie du contexte. Le reste de cet article est organisé comme suit. La section 2 présente un état de l'art sur les ontologies du contexte. La section suivante est précisément consacrée à la construction de l'ontologie de contexte. Nous présentons à la section 4 le méta modèle de la sécurité et son enrichissement par l'ontologie du contexte. A l'issue de cette phase nous obtenons un méta modèle de la sécurité qui intègre le contexte notamment dans sa relation avec les actifs à protéger. La section 5 est consacrée à l'illustration de notre démarche par une étude de cas réel. La conclusion et à la présentation de quelques voies de recherche future font l'objet de la dernière section.

2. Ontologie du contexte : un état de l'art

Le terme contexte, bien que présent dans une multitude de travaux de recherche, ne fait pas l'objet d'une définition consensuelle (Brazire and Brézillon, 2005). Son interprétation et son éventuelle formalisation dépend du domaine d'utilisation. A titre d'exemple, dans le « mobile-learning », le contexte peut aider et soutenir le processus d'apprentissage en fournissant des informations pertinentes ou des services dont l'apprenant peut avoir besoin. (Ardila, 2013). Dans ce contexte d'apprentissage, (Christopoulos, 2008) propose un modèle de contexte à cinq dimensions (information temporelle de l'utilisateur, lieu, artefact, temps et conditions physiques). Ce modèle a, par la suite, été enrichi dans (Gomez et al., 2014) par l'introduction de nouvelles dimensions et de nouveaux éléments de contexte caractérisant ces dimensions. Ces éléments sont décrits dans une taxonomie présentée dans (Ardila, 2013).

Dans le cadre de l'informatique orienté service, (Cabrera et al., 2014) adoptent la définition du contexte proposée dans (Dey, 2001) pour le développement d'applications sensibles au contexte. A l'issue d'un état de l'art sur la modélisation du contexte fondée sur les ontologies (Aguilar et al., 2017), ces auteurs proposent une description du contexte (sous forme d'une ontologie) à partir de onze éléments de contexte (concepts) identifiés : le temps, le lieu, l'activité, l'environnement, le facteur humain, la ressource, la politique, la préférence, le rôle, le profil et l'infrastructure). Des synonymes ont été associés aux différents concepts.

Dans le cadre de la modélisation d'applications sensibles au contexte dans les nuages (context-aware application in the cloud), (Aguilar et al., 2017) proposent une ontologie nommée CAMeOnto, associée à leur outil CARMiCLOC, dans l'objectif de construire un contexte et de le partager. L'ontologie proposée est fondée sur le principe des 5W (Who, When, What, Where et Why). Elle réutilise les deux ontologies CONON et MAont présentées respectivement dans (Guermah et al., 2014) et (Zhong-Jun et al., 2016). Cette ontologie regroupe six concepts, reliés entre

eux, et décrivant le contexte : l'utilisateur, l'activité, le temps, le lieu, le service et le dispositif.

Pour caractériser le contexte dans lequel une application d'entreprise doit fournir ses services, (Nadoveza et al., 2014) proposent de distinguer, dans leur ontologie, (1) les informations concernant les caractéristiques de l'application, (2) celles concernant le contexte métier telles que les activités en cours associées au métier (3) celles qui décrivent le contexte utilisateur tels que ses préférences et le dispositif qu'il utilise pour accéder.

A côté des travaux de recherche sur la description et la modélisation du contexte, on trouve aussi un certain nombre de normes faisant référence à ce terme et à ses composants sans pour autant offrir une modélisation correspondante. A titre d'exemple, pour la mise en place d'un système de management de la qualité (SMQ), la norme ISO 9001 :2015[1] exige des organisations, à travers la clause « Context of the organisation », la compréhension du contexte interne et externe à l'organisation. La compréhension du contexte externe, tel que le mentionne la norme, peut être facilitée par l'examen des problèmes pouvant découler des environnements juridiques, technologiques, concurrentiels, culturels, etc. La compréhension du contexte interne peut être facilitée par l'examen de questions liées aux valeurs, à la culture, aux connaissances et aux performances de l'organisation.

Aussi, une des lignes directrices relatives à la gestion des risques en sécurité de l'information mentionnées dans la norme ISO/IEC 27005:2011[2] est l'établissement du contexte de l'organisation. La norme ISO/CEI 27000 :2018[3] qui comprend les termes et les définitions d'usage courant dans la famille des normes SMSI (Systèmes de Management de la Sécurité de l'Information) dont fait partie la norme ISO/IEC 27005:2011, distingue le contexte interne du contexte externe. Elle définit ce dernier comme étant l'environnement extérieur dans lequel l'organisation cherche à atteindre ses objectifs. Cela peut inclure : (i) les aspects culturels, sociaux, politiques, juridiques, réglementaires, financiers, technologiques, économiques, naturels ainsi que l'environnement concurrentiel, qu'il soit international, national, régional ou local ; (ii) les principales tendances ayant un impact sur les objectifs de l'organisation ; (iii) les perceptions et les valeurs des parties prenantes externes. Le contexte interne, quant à lui, est défini, dans cette même norme, comme intégrant : (i) la gouvernance, la structure organisationnelle, les rôles et les responsabilités; (ii) les politiques, les objectifs et les stratégies qui sont en place pour les atteindre ; (iii) les capacités, exprimées en termes de ressources et de connaissances (par exemple le capital, le temps, les gens, les processus, les systèmes et les technologies); (iv) les systèmes d'information, les flux d'information et les processus de prise de décision (à la fois formel et informel) ; (v) les relations avec et les perceptions et les valeurs des parties prenantes internes ; (v) la culture de l'organisation ; (vi) les normes,

1 https://www.iso.org/fr/standard/62085.html
2 https://www.iso.org/standard/56742.html
3 https://www.iso.org/standard/73906.html

directives et modèles adoptés par l'organisation ; et (vii) la forme et l'étendue des relations contractuelles.

Des méthodes de gestion des risques informatiques, telles que EBIOS (ANSSI, 2010) et MAGERIT (Amutio et al., 2014) soulignent aussi l'importance de l'établissement du contexte et décrivent textuellement le contexte d'une organisation. La méthode EBIOS considère que le contexte est composé de deux catégories. L'une est externe et comprend l'environnement social, culturel, politique, légal, réglementaire, financier, technologique, économique, naturel et concurrentiel, tant au niveau international, national, que régional ou local. Elle considère aussi les facteurs et les tendances ayant un impact déterminant sur les objectifs ainsi que les relations avec les parties prenantes externes, leurs perceptions et leurs valeurs. L'autre catégorie, interne, comprend la description générale de l'organisme, les aptitudes en matière de gestion des ressources, les missions, les valeurs, les métiers, etc. La méthode MAGERIT fait la différence entre le contexte interne et les autres concepts qui constituent l'ensemble du contexte. Le contexte interne est composé des politiques internes ainsi que des intervenants internes et du management ou de ses représentants.

3. Construction de l'ontologie du contexte

Une ontologie est une relation de concepts permettant de partager un ensemble de connaissances d'un domaine donné. Exploitable par les systèmes informatiques, elle permet d'expliciter et d'interpréter les termes nécessaires pour partager la connaissance liée à ce domaine. Deux types de conception existent: la conception manuelle et celle fondée sur des apprentissages. Le premier type est coûteux et surtout pose des problèmes de maintenance et de mise à jour. La conception reposant sur des apprentissages utilise des procédés de construction plus automatiques qui mènent généralement à la conception d'ontologies dites légères. Dans (Maedche et al., 2001) différents types d'approches sont distingués. Les méthodologies de construction d'ontologies les plus connues sont : KACTUS (modelling Knowledge About Complex Technical systems for multiple USe) (Kactus, 1996) et METHONTOLOGY (Fernández-López et al., 1997) qui s'applique à clarifier les différentes phases de la construction en respectant les activités de gestion de projets, de développement et des activités de support. Les activités de gestion de projet correspondent aux activités qui initient et suivent le projet. Les activités de support assurent la réussite du projet. Parmi ces activités de support, on peut citer l'activité d'acquisition des connaissances des experts du domaine ou encore à partir de la documentation fournit par des experts du domaine. Dans les activités de développement, on retrouve les activités de spécification des besoins, de conceptualisation et de formalisation de l'ontologie. Citons aussi la méthode On-To-Knowledge (Staab, et al., 2001) qui tient compte du domaine d'application de l'ontologie en construction. Enfin, Neon (Suárez-Figueroa et al., 2012) propose un cadre méthodologique fondé sur neuf scénarios de construction d'ontologies.

Pour la construction de notre ontologie de contexte, nous avons choisi de mettre en œuvre les activités de développement décrits dans Methonthology. Dans ce cadre nous avons d'abord réuni l'ensemble des sources d'information utiles à sa construction (activité d'acquisition des connaissances) sachant que notre objectif est de l'utiliser pour récupérer le contexte d'une entreprise dans le cadre d'un processus de dérivation des exigences de sécurité fondée sur l'analyse des risques (spécification des besoins). De notre état de l'art, nous avons retenu comme source d'information les documents décrivant les méthodes d'analyse des risques informatiques (EBIOS et MAGERIT en font partie) et les normes ISO citées dans notre état de l'art.

L'activité de conceptualisation a été menée de façon incrémentale. Le premier incrément a consisté à extraire de la norme ISO/CEI 27000 : 2018, tous les termes décrivant le contexte ainsi que les liens pouvant exister entre eux et d'en faire une abstraction sous forme de digramme des classes UML. Chaque incrément qui a suivi a permis de traiter une source d'information parmi celles sélectionnées dans la phase d'acquisition des connaissances. Ce traitement a consisté à extraire les termes relatifs au contexte afin d'enrichir, si besoin est, le modèle conceptuel issu de l'incrément précédent. Ceci nous a permis d'obtenir le modèle conceptuel de l'ontologie du contexte ci-dessous (Figure 1).

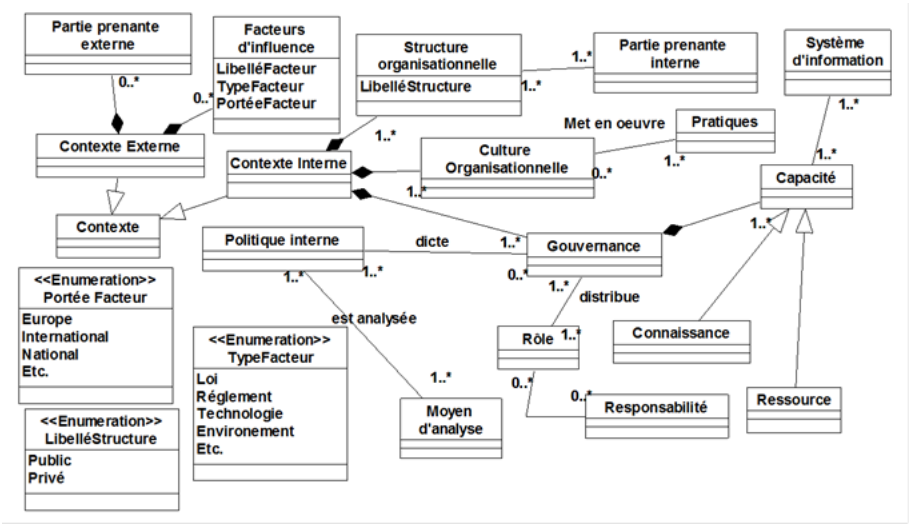


Figure 1. Ontologie du contexte

4. Processus d'enrichissement du méta modèle de la sécurité

Nous présentons ci-dessous (Figure 2) une démarche en deux phases qui permet d'instancier le méta modèle de sécurité (Figure 3) qui sert de base au processus de dérivation des objectifs de sécurité fondée sur l'analyse des risques.

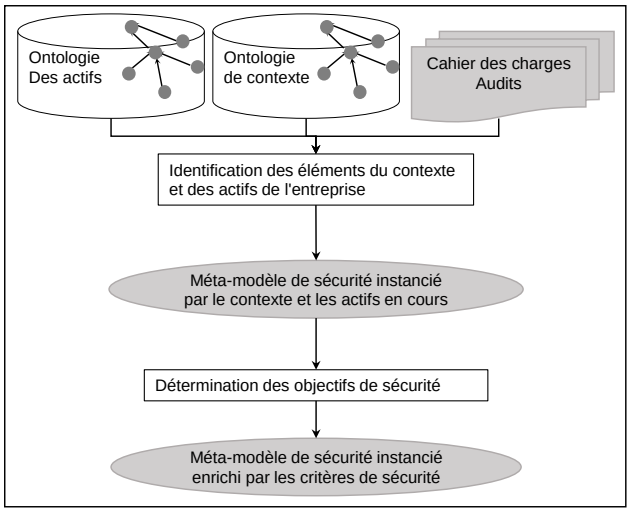


Figure 2. Processus d'enrichissement du méta modèle de la sécurité.

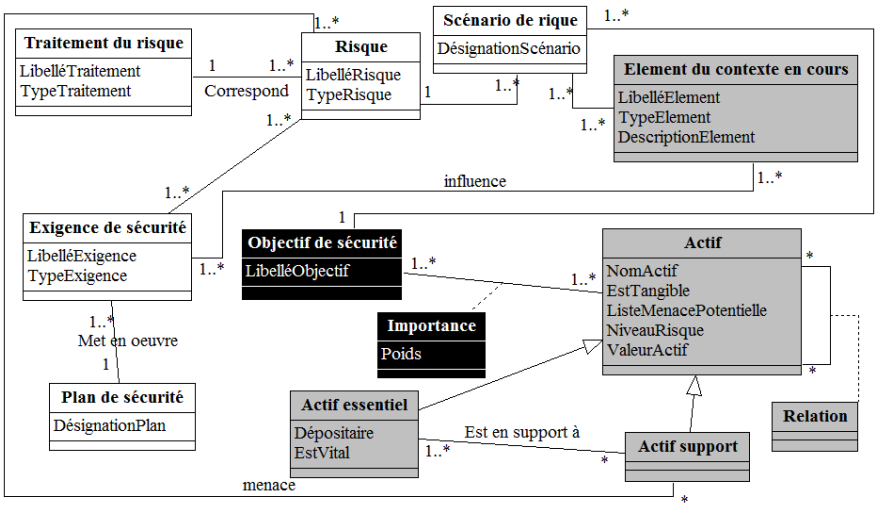


Figure 3 : Méta modèle de la sécurité.

Notre méta modèle a été construit sur la base de nos travaux antérieurs (Vasquez et al., 2012), (Laoufi, 2017). Il est composé de trois groupes de concepts. Le premier groupe (en noir) est constitué des concepts relatifs aux objectifs de sécurité. Le second groupe (en blanc) est relatif aux concepts utilisés dans la phase de dérivation des exigences de sécurité. Le dernier groupe (en gris) est composé des concepts relatif au contexte qui comporte aussi ceux des actifs. Les vulnérabilités sont des propriétés de ces derniers. Toutefois, il faut noter que les vulnérabilités sont une des

composantes du risque. Il y a une relation entre le concept de risque et le concept d'actif support qu'on dénomme menace. Les critères de sécurité sont proposés pour les actifs de l'organisation, comme le propose le modèle des actifs de la méthode ISSRM (Naudet, 2016). Chaque relation possède un poids défini d'après la situation d'exploitation de l'actif. Cette valeur est tirée des méthodes utilisées généralement pour la construction de l'ontologie des actifs (Laoufi, 2017). Nous considérons que les actifs font partie du contexte. L'ensemble des actifs sont inclus dans l'ensemble du contexte, si tous les éléments des actifs sont aussi éléments du contexte.

Le contexte joue un rôle prépondérant dans la proposition des exigences de sécurité. Il existe un lien et une certaine interdépendance entre le contexte et les exigences de sécurité. Du fait que ces deux concepts sont rarement formalisés dans les démarches orientées sécurité et afin de formaliser le lien entre le contexte et les exigences de sécurité, nous rajoutons deux concepts du contexte (ressources, connaissances) et nous proposons une association dénommée "influence" dans le méta-modèle de la sécurité.

4.1. Identification des éléments du contexte et des actifs de l'entreprise

Cette phase requiert en entrée des informations concernant l'entreprise (cahiers des charges pour la conception du SI et/ou des audits s'ils existent) et les ontologies du contexte et des actifs. L'ontologie des actifs est décrite dans (Laoufi, 2017).

L'identification des éléments du contexte de l'entreprise requiert le recours à l'ontologie du contexte présentée dans la section 3. A cette fin, nous faisons appel à une technique d'extraction d'information pour capturer, comparer et identifier les éléments du contexte contenus dans les sources fournies par l'entreprise. Rappelons que l'extraction d'informations est un processus par lequel un système automatique est capable de traiter des documents par une approche linguistique (Turenne, 2010). Nous avons choisi les logiciels AutoMap (Carley et al, 2013a) et ORA (Carley et al, 2013b) pour réaliser les opérations d'extraction des informations et de visualisation des résultats. Automap est un outil dédié à l'exploration des textes. Il permet l'extraction d'informations à partir des textes en utilisant des méthodes d'analyse de texte. Il soutient l'extraction de plusieurs types de données, de documents non structurés. L'information qui peut être extraite comprend : le contenu des données analytiques (mots et fréquences), les données du réseau sémantique (réseau de concepts), les données méta réseau (la classification croisée des concepts dans leur catégorie ontologique comme les gens, les lieux et les choses et les liens entre ces concepts classés), et les données de sentiment (attitudes, croyances). Le logiciel ORA sert, quant à lui, à visualiser les données. À noter que cette phase s'exécute automatiquement en utilisant un logiciel d'exploration de texte fondé sur les approches syntaxiques. La mise en œuvre s'appuie pour son application sur un filtrage que nous avons conçu en utilisant les connaissances qui peuplent l'ontologie du contexte.

Cette ontologie contribue à la détection des éléments de contexte se trouvant dans les documents en entrée du processus (cahiers des charges et/ou rapports

d'audit) ainsi que qu'à leur regroupement en concepts génériques. Elle permet aussi de repérer les actifs.

A l'issue de l'identification des éléments du contexte et des actifs de l'entreprise, nous instancions notre modèle de la sécurité (Figure 3). Cette instanciation concerne la partie grisée de notre méta modèle.

4.2. Détermination des objectifs de sécurité

Cette deuxième phase de notre démarche consiste à déterminer les objectifs de sécurité des actifs de l'organisation en vue de leur instanciation dans le méta modèle de la sécurité. Ces objectifs de sécurité sont là pour garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. Chaque actif peut avoir un ou plusieurs objectifs de sécurité selon le contexte d'utilisation et un niveau de gravité selon le scénario de risque. Pour cela, nous avons rajouté comme guidage à l'utilisateur un concept (poids) dans le méta modèle de sécurité qui possède comme attribut le niveau de gravité d'utilisation d'un des quatre objectifs de sécurité qui sont (ISO/IEC 2700:2018) :

- La confidentialité : propriété selon laquelle l'information n'est pas diffusée ni divulguée à des personnes, des entités ou des processus (3.54) non autorisés ;
- La disponibilité : propriété selon laquelle l'information est accessible et utilisable à la demande par une entité autorisée ;
- L'intégrité : propriété selon laquelle l'information est exacte et complète ;

Le résultat de cette étape est l'enrichissement du méta modèle de la sécurité avec les objectifs de sécurité des actifs de l'entreprise en précisant le poids de réalisation de chacun (Figure 5).

5. Cas d'application

L'étude de cas porte sur le système d'information d'une organisation chargée de gérer les dossiers de demandes de pension d'invalidité. Toute demande donne lieu à une étude par une commission chargée de décider, le cas échéant, du taux d'invalidité. Cette décision est alors transmise au centre payeur le plus proche de l'adresse du demandeur. Le traitement des demandes des pensions d'invalidité nécessite l'intervention de plusieurs acteurs, notamment du demandeur, du gestionnaire, de la commission d'attribution des taux et du centre payeur. Le nombre de bénéficiaires est estimé à 300.000 personnes par an. Le délai moyen d'examen d'un dossier et de l'attribution éventuelle d'une pension est d'un an. Le système d'information souffre aussi de graves lacunes relatives à la sécurité. Pour améliorer ce système, l'entreprise a déclenché un audit du système d'information ainsi que de la procédure d'attribution du taux d'invalidité. Cet audit donne lieu à des recommandations quant au fonctionnement du service et à des mesures de sécurité informatique.

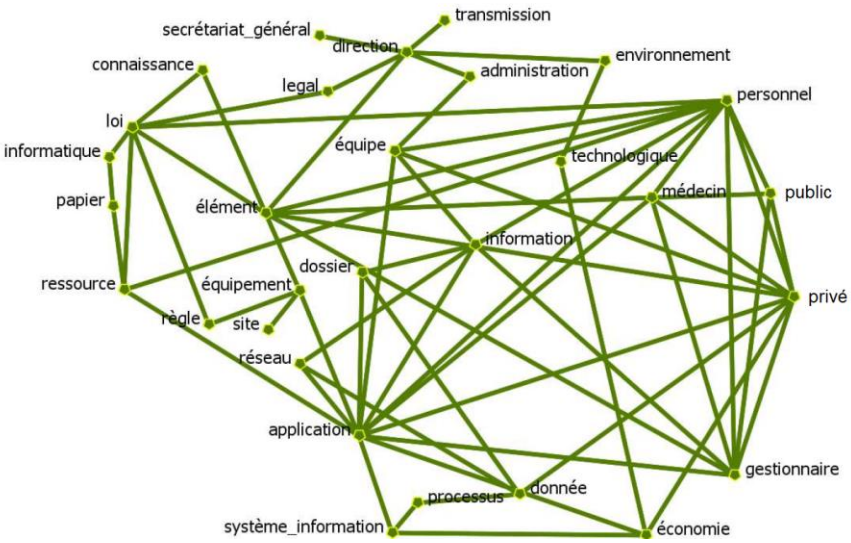


Figure 6. Réseau sémantique des concepts du contexte et des actifs.

Dans une première étape, nous procédons à l'extraction des concepts du contexte liés à l'organisme en utilisant le rapport d'audit. A cette fin, nous utilisons le logiciel Automap pour l'extraction et le logiciel ORA pour la visualisation du résultat. On obtient ainsi un réseau sémantique qui représente les relations existantes entre les concepts du contexte de l'organisation concernée (Figure 6). Dans cette figure, la fréquence des concepts ontologiques correspond à l'épaisseur du trait, ici peu visible.

Puis nous procédons à l'enrichissement du méta modèle de la sécurité avec les résultats obtenus lors de l'extraction précédente. Cet enrichissement est facilité par la mise en correspondance des éléments du contexte avec les concepts génériques de l'ontologie du contexte (Tableau 1).

Elément de contexte de l'organisation	**Concepts de l'ontologie du contexte**
Environnement, Technologique, Loi, Légal	Contexte externe
Direction, Secrétariat général, Règle, Administration, Informatique	Gouvernance
Donnée, Equipe, Dossier, Equipement, Information Réseau, Transmission, Application, Personnel Médecin, Gestionnaire, Public, Civil, Elément Papier, Connaissance, Processus, Ressource	Ressources
Système information	Système information
Site	Site
Donnée, Equipe, Dossier, Equipement, Information Réseau, Transmission, Application, Personnel Médecin, Gestionnaire, Public, Civil, Elément, Papier Connaissance, Processus, Ressource, Système information	Contexte interne

Site, Direction, Secrétariat général, Règle, Administration Informatique	

Tableau 1. Eléments de contexte de l'organisation et concepts de l'ontologie du contexte

Nous procédons de la même façon pour les actifs de l'organisation en les mettant en correspondance avec les concepts génériques de l'ontologie des actifs qui font partie de l'ontologie du contexte.

Actifs de l'organisation	**Concepts de l'ontologie des actifs**
Personnel, Public, Médecin, Gestionnaire, Elément Civil	Personnel
Application, Donnée, Information, Dossier Système information	Application
Equipe, Electronique	Matériel
Réseau	Réseau
Site	Site
Réseau, Application, Donnée, Information, Dossier Système information, Equipe, Electronique	Actif support (technique)

Tableau 2. Eléments des actifs de l'organisation et concepts de l'ontologie des actifs

La dernière phase consiste à déterminer les objectifs de sécurité. A cet effet, nous associons à chaque concept des actifs un ou plusieurs objectif(s) de sécurité en appliquant notre démarche de construction des scénarios des risques (Vasquez et al., 2012). Nous obtenons le tableau ci-dessous (Tableau 3).

Actif	**Contexte**	**Scénarios des risques**	**Objectifs de sécurité**
Hardware Software Site	Contexte externe	Destruction ou altération de ressources techniques, de supports de stockage, de documents ou de locaux du système, par un phénomène naturel majeur.	Disponibilité Intégrité Confidentialité
Software	Contexte externe	Traitement illicite des données personnelles, utilisation des données personnelles à d'autres fins que celles autorisées par la législation ou un règlement.	Confidentialité
Hardware	Gouvernance	Destruction ou altération d'un équipement ou d'un support de stockage d'une plate-forme du système, due à un accident ou une négligence ou encore à un acte délibéré, par une personne ayant accès à cet élément.	Disponibilité Intégrité Confidentialité
Hardware	Ressources	Arrêt ou dysfonctionnement de la climatisation dans les locaux d'une plate-forme, de ceux de support de stockage, de documents ou de d'équipements, suite à une panne ou un acte volontaire.	Disponibilité
Réseaux	Ressources	Au niveau des réseaux ou des supports de communication utilisés, interception des échanges entre un utilisateur et le système, entre deux plates-formes du système, entre deux équipements d'une même plate-forme.	Disponibilité Intégrité Confidentialité
Software	Site	Vol de documents du système, vol ou substitution	Confidentialité

Hardware		d'un support de stockage d'informations dans un site du système, dans un site de stockage.	
Software	Ressources	Personne interne à l'organisme qui, par negligence, diffuse de l'information à d'autres personnes de l'organisme ne devant pas à en connaître, ou à l'extérieur. Personne diffusant consciemment de l'information à d'autres personnes de l'organisme ne devant pas en connaître,	Confidentialité

Tableau 3. Détermination des critères de sécurité (Extrait)

Le résultat obtenu permet d'instancier, et donc d'enrichir, le méta modèle de la sécurité. A noter que nous avons attribué le même poids pour chaque objectif de sécurité.

6. Conclusion et future recherche

Les principales contributions de cet article sont :

- la proposition d'un méta modèle de sécurité qui sert de base à une démarche de dérivation des exigences de sécurité à partir d'une analyse des risques,
- le développement d'une ontologie du contexte fondée sur la norme de sécurité ISO/CEI 27000 : 2018,
- Une démarche d'enrichissement du méta modèle de sécurité par l'ontologie du contexte. Cet enrichissement est réalisé en deux phases. La première est relative à l'identification et à l'extraction des éléments du contexte de l'entreprise et des actifs. La seconde concerne la détermination des objectifs de sécurité des actifs de l'organisation à protéger.
- L'application à un cas réel, et qui sert d'une première étape dans la validation de notre démarche.

Plusieurs axes de recherche future sont possibles. Citons notamment l'enrichissement de l'ontologie du contexte et des règles de correspondance associées, l'application de la démarche à plusieurs exemples, l'exploitation de l'aspect « contexte externe » ainsi que des scénarios de risques associés, ainsi qu'une validation plus large des concepts et des relations ontologiques.

Bibliographie

Abowd G.D., Dey A.K., Brown P.J., Davies N., Smith M., Steggles P. (1999) *Towards a Better Understanding of Context and Context-Awareness*. HUC 1999 (Handheld and Ubiquitous Computing). LNCS 1707. Springer, Berlin, Heidelberg.

Aguilar, J., Jerez, M., Rodriguez, T. (2017). CAMeOnto: Context awareness meta ontology modeling, Applied Computing and Informatics, 2017,ISSN 2210-8327,https://doi.org/10.1016/j.aci.2017.08.001.

Amutio, M., Candau, J. (2014). *Magerit V3 : Methodology for Information Systems Risk Analysis and Management, Book I - The Method*. Ministerio de Hacienda y Administraciones Públicas . http://administracionelectronica.gob.es/.

ANSSI. (2010). https://www.ssi.gouv.fr/uploads/2011/10/EBIOS-1-GuideMethodologique-2010-01-25.pdf.

Ardila, S.E.G. (2013). Learning Design Implementaion in Context-Aware and Adaptive Mobile learning, Thèse Université de Girona, Catalonia, Spain, 2013.

Bazire, M., Brézillon, P. (2005). Understanding Context before Using It. CONTEXT 2005: 29-40.

Brézillon, P., Abu-Hakima, S. (1995). *Using knowledge in its context*: *Report on the IJCAI-93 Workshop*. The AI Magazine, 16(1) pp. 87-91.

Bulcao Neto, R. F., Pimentel, M. G. C. (2005). *Toward a domain-independent semantic model for context-aware computing*. In Proceedings of the 3rd Latin American Web Congress (LA-WEB'05), pages 61–70, Buenos Aires, Argentina, 2005.

Cabrera, O. Franch, X., Marco, J. (2014). *A Context Ontology for Service Provisioning and Consumption*. IEEE RCIS 2014.

Carley, K. M., (2013a). Dave Columbus, D., Landwehr, P. (2013). *AutoMap User's Guide 2013*. Technical Report CMU-ISR-13-105. Institute for Software Research. Carnegie Mellon University. http://www.casos.cs.cmu.edu/projects/automap/CMU-ISR-13-105.pdf.

Carley, K.M., (2013b). *ORA: Quick Start Guide*. http://netanomics.com/wp-content/uploads/2017/03/ORA-QuickStart-Guide-2016.pdf

Chen, H., Finin, T., Joshi, A. (2003). *Using OWL in a Pervasive Computing Broker*. Workshop on Ontologies in Open Agent Systems (AAMAS 2003).

Christopoulou, E. (2008). Context as a necessity in mobile applications. In: Klinger, K. (Ed.), User Interface Design and Evaluation for Mobile Technology, pp. 187–204, 2008.

Dey, A.K. (2001). Understanding and Using Context, Personal Ubiquitous Comput., vol. 5, pp. 4-7, 2001.

Ejigu, D., Scuturici, M., Brunie, L. (2007). *An Ontology-Based Approach to Context Modeling and Reasoning in Pervasive Computing*. CoMoRea Workshop de PerCom'07, White Plains, NY, 2007, pp. 14-19.

Fernández-López, M. and Gómez-Pérez, A. and Juristo, N. (1997). *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. Actes de AAAI-97, 1997, Stanford University.

Gomez S., Zervas, P., Sampson, D.G., Fabregat, R. (2014). Context-aware adaptive and personalized mobile learning delivery supported by UoLmP, Journal of King Saud University - Computer and Information Sciences, Volume 26, Issue 1, Supplement, 2014, Pages 47-61, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2013.10.008.

Guermah, H., Fissaa, T., Hafiddi, H. Nassar, M., Kriouile,A. (2014). An ontology oriented architecture for context aware services adaptation, Int. J. Comput. Sci. 11 (2) (2014).

IFIP-IFAC Task Force. (2003). *Geram: Generalized Enterprise Reference Architecture and Methodology*. Version 1.6.3, Handbook on Enterprise Architecture. Editeurs : Bernus, Peterand Nemes, Laszloand Schmidt. ISBN=978-3-540-24744-9

Kactus. (1996). *The KACTUS Booklet version 1.0*. Esprit Project 8145 KACTUS.

Lammari, N., Bucumi, J. S., Akoka, J., Comyn Wattiau, I. (2011). *A conceptual Meta, Model for Secured Information Systems*.ICSE´11. 2011. DOI: 10.1145/1988630.1988635.

Laoufi, N. (2017). Processus de dérivation des exigences de sécurité à partir de l'analyse des risques, Thèse de doctorat, Conservatoire National des Arts et Métiers, Mars 2017.

Leech, G. (1981). Semantics: The Study of Meaning. Harmondsworth, UK: Penguin, 1981.

Maedche, A., Staab, S. (2001). Ontology Learning for the Semantic Web. IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2), 2001.

Nadoveza, D., Kiritsis, D. (2014). Ontology-based approach for context modeling in enterprise applications. Computers in Industry 65(9): 1218-1231.

Naudet, Y., Mayer, N., Feltus, C. (2016). *Towards a Systemic Approach for Information Security Risk Management*. ARES 2016: 177-186

Staab, S., Schurr, H., Studer, P., Sure, Y. (2001). Knowledge processes and ontologies. IEEE Intelligent Systems 16(1):26-34.

Stumme, G., Hotho, A., Berendt, B. (2006). *Semantic web mining: State of the art and future directions*. Web Semantics: Science, Services and Agents on the World Wide Web, 4(2), pp.124–143.

Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M. (2012). *The NeOn Methodology for Ontology Engineering*, Book Chapter in Ontology Engineering in a Networked World, 2012, Publisher: Springer Berlin Heidelberg, pp. 9-34.

Turenne, N., (2010). *Apprentissage statique pour l'extraction de concepts à partir de textes*, Doctoral Thesis in Computer sciences.

Vasquez, M., Lammari, N., Comyn-Wattiau, I., Akoka, J. (2012). *De l´analyse des risques à l´expression des exigences de sécurité des systèmes d'information*, INFORSID 2012.

Zhong-Jun, L., Guan-Yu, P., Ying, A (2016*). Method of meta-context ontology modeling and uncertainty reasoning in SWoT*, Proceedings of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (2016) 128–135.

INDEX DES AUTEURS

M

N

P

Q

R

S

T

V

Résumé

Ce document contient les actes du trente-sixième congrès INFORSID (INFormatique des ORganisations et Systèmes d'Information de Décision) qui s'est déroulé à Nantes du 28 au 31 mai 2018. Le processus de sélection des articles publiés a été organisé à deux niveaux avec un Conseil du Comité de Programme (CoP) additionnel au Comité de Programme habituel (CP).