INFORSID 2016

34e édition - Grenoble

INFORSID 2016

Actes du Congrès INFORSID, 34e édition

Grenoble, 31 mai - 3 juin 2016

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 : Philippe ROOSE, LIUPPA, Université de Pau et des Pays de l'Adour, IUT de Bayonne
Secrétaire : Agnès FRONT, LIG, Université Grenoble Alpes
Chargé de communication : Elöd EGYED-ZSIGMOND, LIRIS, Université de Lyon, INSA de Lyon

Présidents d'honneur

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

Préface

Stratégiques à la compétitivité des entreprises et des organisations, vecteurs de diffusion des innovations numériques, les systèmes d'information concentrent un grand nombre des problématiques portées par la communauté scientifique autour des STIC. C'est donc avec grand plaisir que j'écris cette préface pour l'édition des actes de la trente-quatrième édition du congrès INFORSID (INFormatique des ORganisations et Systèmes d'Information et de Décision). A la croisée de différents domaines de l'informatique, ce congrès annuel est un temps fort de notre communauté où les échanges et présentations favorisent les enrichissements mutuels et la mise en perspective de nos recherches.

En 2016 le congrès se tient conjointement à la conférence internationale RCIS (Research Challenges in Information Science). Ce rapprochement est l'occasion de nouvelles interactions et une opportunité pour renforcer la synergie entre nos communautés.

Trois conférenciers invités, communs aux deux conférences, nous ont fait l'honneur d'accepter notre invitation. Il s'agit de Barbara Pernici (Politecnico di Milano, Italy), Franck Barbier (Université de Pau, LIUPPA) et Laurent Lefèvre (Inria, Lyon).

Des ateliers mettent le focus sur des problématiques prégnantes *Ville digitalisée contributive*, *Sécurité des systèmes d'information* et les *SI pour l'aide à la décision et la diffusion d'alerte*, tandis que l'atelier *Enseignement des SI* vise au partage des pratiques d'enseignements liées à ces systèmes. Une table ronde autour de *l'innovation par les SI dans l'écosystème* sera l'occasion d'échanges entre industriels et chercheurs issus d'horizons différents.

Ce congrès a, entre autres, pour objectif de favoriser l'intégration des jeunes chercheurs. C'est pourquoi, tous les deux ans, le forum *Jeunes Chercheurs*, présidé cette année par Cécile Favre, met en exergue les travaux des jeunes chercheurs en première ou deuxième année de doctorat à la fois sous la forme de courtes présentations et de posters exposés le temps de la conférence. Il faut noter en 2016 un taux important de soumissions qui témoigne de la forte activité de la communauté.

Cette année, le congrès INFORSID a reçu 42 soumissions d'articles dont les auteurs sont issus de différents pays (France, Algérie, Belgique, Canada, Colombie, Espagne, Finlande, Maroc, Suisse, Tunisie). Vingt articles ont été acceptés. Comme chaque année, les articles couvrent un large panel des problématiques liées aux systèmes d'information, des exigences aux mises en œuvre, des données aux processus métier, du génie logiciel à l'intelligence artificielle. Quatre des articles acceptés ont fait l'objet d'un accompagnement pour aboutir à la version présente dans ces actes. Ce travail des auteurs et des accompagnateurs fait écho à la volonté de la communauté INFORSID de participer au "passage de la science informatique". Douze des articles soumis étaient officiellement en double soumission à RCIS et INFORSID ; nous en avons accepté huit qui ont tous été également acceptés à RCIS, témoignant ainsi de l'implication de la communauté en systèmes d'information à l'international. Les articles acceptés à RCIS et à INFORSID sont officiellement publiés à RCIS et apparaissent dans ces présents actes sous la forme d'un résumé.

Le processus de sélection des articles s'est déroulé en plusieurs phases. Dans un premier temps, chacun des articles soumis a été évalué par trois membres du Comité de Programme. Puis, les membres du Conseil du Comité de Programme ont, sur chaque article dont les avis divergeaient, dirigé des discussions entre les membres du Comité de Programme. Enfin, une réunion plénière entre les membres du Conseil du Comité de Programme a permis de sélectionner les articles acceptés pour présentation lors du congrès.

Avant de clôturer cette préface, je tiens à remercier les membres du bureau de l'association INFORSID, sous la présidence de Régine Laleau, pour m'avoir confié l'organisation scientifique du congrès et pour leur assistance et implication tout au long de cette année.

Je voudrais également remercier chaleureusement tous ceux qui ont contribué à l'organisation de ce congrès :
- les auteurs pour l'énergie mise dans la rédaction des articles ; les conférenciers venus nous présenter les articles sélectionnés ;
- les membres du comité de programme et les relecteurs additionnels qui, par la richesse de leurs retours, contribuent à dynamiser nos recherches ; les "bergers" pour leur investissement dans l'aide à une meilleure diffusion des résultats de recherche ;
- les membres du conseil du comité de programme pour leurs conseils avisés à toutes les étapes de la mise en place du congrès ;
- les conférenciers invités pour avoir accepté de nous faire partager leur savoir et leurs expériences ;
- les porteurs des ateliers et de la table ronde pour leur investissement et leur dynamisme à organiser ces rencontres enrichissantes et originales ;
- Cécile Favre pour avoir pris en charge avec une grande efficacité le forum Jeunes Chercheurs, et avoir ainsi contribué à l'accueil des jeunes chercheurs dans notre communauté ;
- les participants au congrès pour faire vivre et promouvoir notre communauté.
Je suis spécialement reconnaissante envers Guillaume Cabanac qui, par son travail autour de notre communauté, m'a aidée à construire le comité de programme en tenant compte de la longue histoire d'INFORSID.

Enfin, je remercie très chaleureusement le comité d'organisation de nous recevoir sur Grenoble avec toute la complexité d'une double conférence, la prise en charge d'un forum jeunes chercheurs, dans un contexte de fusion des universités ! Un merci particulier à
- Gaëlle Blanco-Laine, Eric Ceret, Sophie Dupuy-Chessa, Akram Idani, et Claudia Roncancio qui ont travaillé activement et "avec agilité" à la préparation de ce congrès ;
- Marlène Villanova-Oliver pour la publication des actes ;
- Cyril Labbe pour sa gestion "quotidienne" du site web ;
- Agnès Front pour avoir su si bien partager son expérience.
Un grand MERCI à Dominique Rieu, dynamique, joviale et efficace présidente de l'organisation.

Je souhaite à tous un excellent congrès INFORSID 2016 !

Mireille Blay-Fornarino
Présidente du comité de Programme

Ces journées ont été gérées en utilisant l'outil easychair.

Comités

Le comité de la 34e 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. Les président(e)s sont mentionné(e)s par une étoile (*).

Comité d'organisation

Gaëlle BLANCO-LAINE	Université Grenoble Alpes
Eric CERET	LIG, Université Grenoble Alpes
Sophie DUPUY-CHESSA	LIG, Université Grenoble Alpes
Agnès FRONT	LIG, Université Grenoble Alpes
Akram IDANI	LIG, Université Grenoble Alpes
Cyril LABBE	LIG, Université Grenoble Alpes
Dominique RIEU *	LIG, Université Grenoble Alpes
Claudia RONCANCIO	LIG, Université Grenoble Alpes
Marlène VILLANOVA-OLIVER	LIG, Université Grenoble Alpes

Conseil du comité de programme

Sylvie Calabretto	INSA Lyon, LIRIS
Corine Cauvet	LSIS, Université Aix-Marseille
Thierry Delot	LAMIH, Université de Valenciennes
Rebecca Deneckere	Université Paris 1 Panthéon-Sorbonne, CRI
Régine Laleau	Université Paris-Est Créteil, LACL
Kathia Marcal de Oliveira	LAMIH, Université de Valenciennes
Franck Ravat	IRIT, Université Toulouse I Capitole
Dominique Rieu	LIG, Université Grenoble Alpes
Florence Sèdes	IRIT, Université Paul Sabatier Toulouse III

Comité de programme

Judith Barrios	Université de Los Andes, GIDyC, Mérida Venezuela
Sadok Ben Yahia	Faculty of Sciences, Tunis
Reda Bendraou	UPMC, LIP6
Antoine Beugnard	Telecom Bretagne
Mireille Blay-Fornarino *	Université Nice Sophia Antipolis, I3S
Célia Da Costa Pereira	Université Nice Sophia Antipolis, I3S
Jean-Christophe Desconnets	Institut de Recherche pour le Développement (UMR ESPACE-DEV)
Sophie Dupuy-Chessa	Université Grenoble Alpes, LIG
Sophie Ebersold	Université Toulouse II Jean Jaures, IRIT
Cyril Faucher	Université de La Rochelle, L3i

Nicolas Ferry	SINTEF, Norway
Ollivier Haemmerlé	Université Toulouse II Jean Jaures, IRIT
Slim Hammadi	Ecole centrale de Lille, CRIStAL
Marianne Huchard	Université de Montpellier, LIRMM
Abdelaziz Khadraoui	University of Geneva, ISS
Léa Laporte	INSA Lyon, LIRIS
Eric Leclercq	Université de Bourgogne, LE2I
Yves Ledru	Université Grenoble Alpes, LIG
Emmanuel Letier	University College London
Philippe Lopisteguy	Université de Pau et des Pays de l'Adour, LIUPPA
Philippe Merle	INRIA Lille
Elisabeth Murisasco	Université de Toulon, LSIS
André Péninou	Université Toulouse II Jean Jaures, IRIT
Thomas Polacsek	ONERA Toulouse
Christophe Ponsard	CETIC
Pierre-Edouard Portier	INSA Lyon, LIRIS
Christophe Rey	Université Clermont Auvergne, LIMOS
Philippe Roose	Université de Pau et des Pays de l'Adour, LIUPPA
Malika Smail-Tabbone	Université de Lorraine, LORIA
Chantal Soulé-Dupuy	Université Toulouse 1 Capitole, IRIT
Carine Souveyet	Université Paris 1, CRI
Dalila Tamzalit	Université de Nantes, LINA
Jean Vanderdonckt	Université catholique de Louvain, Louvain Interaction Lab
Christine Verdier	Université Grenoble Alpes, LIG
Herve Verjus	Université Savoie Mont Blanc - LISTIC

Relecteurs additionnels

Jean-Christophe Bach, Patrice Bellot, Amine Benelallam, Eric Bourreau, Sandra Bringay, Isabelle Mougenot, Touzani Mounir, Dagorret Pantxika, Patrick Etcheverry, Romain Rouvoy, Boukhedouma Saida

Table des matières

Session Modélisation de Données No-SQL

Document-oriented data warehouses: models and extended cuboids

Modèle orienté-document et cuboïdes étendus

Max Chevalier[1], Mohammed El Malki [1,2] Arlind Kopliku, Olivier Teste, Ronan Tournier

1. Université de Toulouse, IRIT (UMR 5505, www.irit.fr), Toulouse, France
prenom.nom@irit.fr

2. Capgemini (www.capgemini.com) Toulouse France
prenom.nom@capgemini.com

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS

ABSTRACT. Within the Big Data trend, there is an increasing interest in Not-only-SQL systems (NoSQL). These systems are promising candidates for implementing data warehouses particularly due to the data structuration/storage possibilities they offer. In this paper, we investigate data warehouse instantiation using a document-oriented system (a special class of NoSQL systems). On the one hand, we analyze several issues including modeling, querying, loading data and OLAP cuboids. We compare document-oriented models (with and without normalization) to analogous relational database models. On the other hand, we suggest improvements in order to benefit from document-oriented features. We focus particularly on extended versions of OLAP cuboids that exploit nesting and arrays. They are shown to work better on workloads with drill-down queries.

KEYWORDS: NoSQL, document-oriented system, big data warehouse, OLAP cuboid.

RESUME: Parmi les nouvelles technologiques du Big Data, il y a un intérêt croissant pour les systèmes Not-Only-SQL (NoSQL). Ces derniers sont des candidats prometteurs pour l'implantation d'entrepôts de données grâce aux possibilités qu'ils offrent en termes de stockage et de structuration des données. Dans ce papier, nous étudions l'instanciation d'entrepôts de données avec les systèmes orientés document, une des catégories les plus répandues des systèmes NoSQL. Dans un premier temps, nous étudions les enjeux primaires des entrepôts tels que la modélisation, l'interrogation, le chargement des données, et les cuboïdes OLAP. Dans un deuxième temps, nous proposons des améliorations qui sont spécifiques aux systèmes orientés document. En particulier, nous proposons des versions étendues des cuboïdes OLAP qui exploitent l'imbrication et les tableaux. Nous montrons que ces cuboïdes répondent plus rapidement à des charges de travail composées de requêtes OLAP de type "drill-down ».

MOTS CLES: NoSQL, système orienté-document, entrepôts de données big data, cuboïde OLAP

Processus de transformation MDA d'un schéma conceptuel de données en un schéma logique NoSQL

Fatma ABDELHEDI [(1) et (3)], Amal AIT BRAHIM [(1)], Faten ATIGUI [(2)], Gilles ZURFLUH [(1)]

1. IRIT - Université Toulouse Capitole - France
<prenom.nom>@irit.fr
2. CEDRIC – CNAM Paris – France
<prenom.nom>@cnam.fr
3. CBI² - Sté TRIMANE Saint Germain en Laye – France
<prenom.nom>@gmail.com

RESUME : *La transformation digitale des entreprises et plus largement celle de la société, entraine une évolution des bases de données vers le Big Data. Nos travaux s'inscrivent dans cette mutation et concernent plus particulièrement les mécanismes d'implantation d'une base de données sur une plateforme NoSQL. Pour automatiser ce processus d'implantation, nous avons utilisé l'architecture MDA qui offre un cadre formalisé aux mécanismes de transformation des schémas. A partir d'un schéma conceptuel décrivant une base d'objets complexes, nous proposons des règles de dérivation pour générer, in fine, un schéma d'implantation destine à une plateforme NoSQL orientée colonnes. Nous introduisons un schéma intermédiaire de niveau logique afin de limiter les impacts liés aux évolutions techniques des plateformes NoSQL. Une expérimentation du processus de transformation a été réalisée sur une application médicale.*

ABSRACT: Recent years have seen a real explosion of volume of data available in business and on the web. In this paper, we consider the automatic transformation of Big Data conceptual schema within NoSQL systems. For this, we use the Model Driven Architecture (MDA) that provides a framework for models automatic transformation. Starting from a conceptual model that describes a set of complex objects, we propose transformation rules to generate, ultimately, two NoSQL models: columns-oriented model and documents-oriented model. To ensure efficient automatic transformation, we use a logical model that limits the impacts related to technical developments of NoSQL platforms. We provide experiments of the QVT model transformations in the context of health area.

Mots-clés : Big Data, systèmes NoSQL, transformation de schémas, architecture MDA.

Keywords: Big Data, NoSQL systems, schema transformation, MDA.

1. Introduction

Pendant trois décennies, les systèmes relationnels ont représenté l'outil majeur pour l'exploitation des bases de données. Mais le modèle relationnel connaît des limites face aux nouvelles applications qui apparaissent sur le Web. Ces dernières années ont connu une véritable explosion du volume des données disponibles dans les entreprises et sur le Web. Ces nouvelles problématiques sont désignées par l'expression « Big Data » [5] et caractérisées par la règle dite des « 3V » [10]. Il s'agit du Volume (des masses considérables des données à gérer), de la Variété (des données complexes) et de la Vélocité (en référence à la collecte et au traitement en temps-réel de ces données). Les approches classiques basées principalement sur le paradigme relationnel, ne peuvent pas répondre à ces objectifs et exigent de nouvelles approches de stockage et de manipulation des données. Regroupées sous le terme NoSQL [8], ces approches permettent une plus grande adaptabilité dans des contextes fortement distribués, ainsi qu'une gestion performante des données complexes [7]. Les développeurs d'applications Big data se trouvent notamment confrontés à la problématique du stockage des données avec des systèmes NoSQL. L'objectif de nos travaux est donc de faciliter le processus d'implantation de bases d'objets sur des plateformes NoSQL. En raison de la complexité des schémas des bases de données à implanter et de la spécificité des modèles NoSQL, nous proposons un processus de transformation des modèles basé sur l'approche de l'Ingénierie Dirigée par les Modèles [3].

Dans la section 2 suivante, nous présentons l'application médicale qui justifie l'intérêt de nos travaux. La section 3 décrit le contexte de notre étude ainsi que notre problématique de recherche visant à implanter des données sur une plateforme NoSQL. La section 4 présente notre contribution qui consiste à formaliser avec MDA le processus de transformation d'un schéma conceptuel de données en un schéma logique NoSQL. La section 5 décrit une expérimentation du processus proposé à partir de notre application médicale. Enfin, la section 6 positionne nos travaux par rapport à l'état de l'art.

2. Cas d'étude

Pour illustrer nos travaux, nous utilisons un cas extrait d'une application médicale dont la base de données est représentée dans le formalisme UML. Cet exemple nous permettra de montrer comment transformer un diagramme UML en un schéma NoSQL. Il s'agit de la mise en place de programmes nationaux ou internationaux pour le suivi de cohortes de patients atteints de pathologies graves. L'objectif majeur d'un tel programme est de collecter des données sur l'évolution temporelle d'une pathologie particulière, d'étudier les interactions de la pathologie avec des maladies opportunes et d'évaluer l'influence des traitements et médications à court et moyen termes. La durée d'un programme est décidée lors de son lancement et peut atteindre trois ans. Les données collectées par plusieurs établissements dans le cadre d'un programme pluriannuel, présentent les caractéristiques généralement admises pour le Big Data (les 3 V). En effet, le volume des données médicales recueillies quotidiennement auprès des patients, peut atteindre, pour l'ensemble des établissements et sur trois années, plusieurs téraoctets. D'autre part, la nature des données saisies (mesures, radiographie, scintigraphies, etc.) est diversifiée et peut varier d'un patient à un autre selon son état de santé. Enfin, certaines données sont produites en flux continu par des capteurs ; elles doivent être traitées quasiment en temps réel car elles peuvent s'intégrer dans des processus sensibles au temps (mesures franchissant un seuil qui

impliqueraient l'intervention d'un praticien en urgence par exemple). Le suivi des patients exige le stockage de données variées telles que l'enregistrement des consultations effectuées par les praticiens, des résultats d'examens, des prescriptions de médicaments et de traitements spécifiques. L'extrait de diagramme UML de la figure 1 montre quelques classes pour un programme médical associé au suivi des patients atteints d'une pathologie particulière.

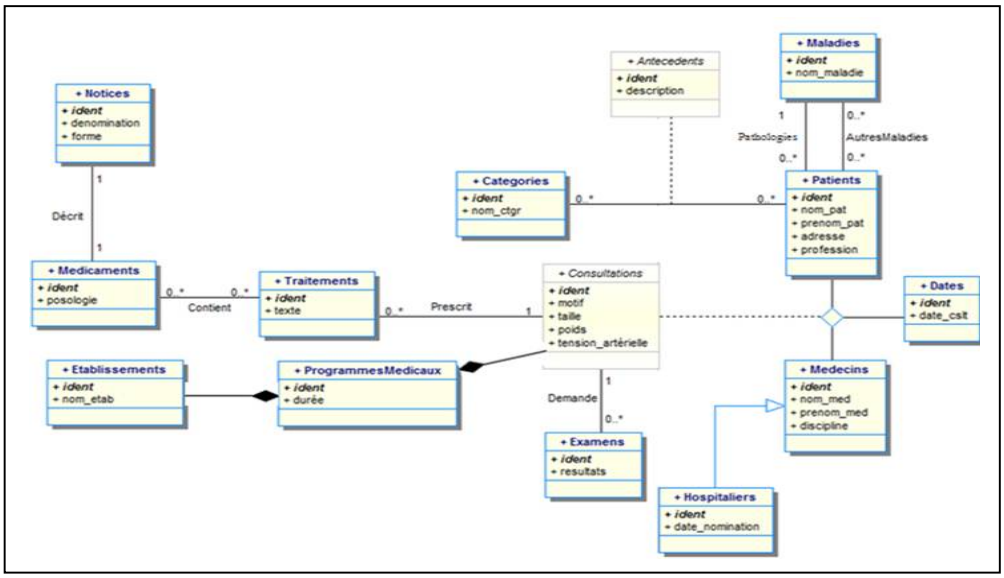


Figure 1. Extrait d'un schéma des données

3. Contexte et Problématique

Notre objectif est de partir d'un schéma de bases de données tel que celui de la figure 1 et de le transformer en un schéma NoSQL. Nous présentons dans cette section le contexte de notre étude en abordant les principes généraux de l'approche IDM. Nous décrivons ensuite la problématique de recherche traitée dans le cadre de cet article.

3.1. Ingénierie Dirigée par les Modèles (IDM)

Pour faire face à la complexité des applications informatiques, l'IDM [3] consiste à adopter les modèles comme éléments centraux dans le processus de développement d'une application et à automatiser la transformation de ces modèles afin d'aboutir au code source. L'Architecture Dirigé par les Modèles (MDA pour Model Driven Architecture) [9] proposée par l'Object Management Group[1] (OMG) est un mécanisme dérivé de l'IDM. MDA propose de décrire séparément les spécifications fonctionnelles et les spécifications

d'implantation d'une application sur une plateforme donnée. Parmi les modèles proposés, nous retenons (1) le modèle d'analyse et de conception (Platform Independent Model – PIM) qui décrit les données en faisant abstraction des aspects techniques liés aux systèmes informatiques et (2) le modèle de code (Platform Specific Model – PSM) qui représente les données en tenant compte des caractéristiques d'une plateforme de stockage particulière.

Dans la suite de l'article, nous considérons deux sortes de PIM : le PIM conceptuel qui décrit les données sous ses seuls aspects métier et le PIM logique qui tient compte en plus du type d'organisation des données choisi. Le passage entre deux modèles MDA se fait via une succession de transformations. Une transformation correspond à l'application d'un ensemble de règles qui décrit comment dériver un modèle cible à partir d'un modèle source. Pour la transformation de modèles, l'OMG a défini le standard QVT[2] (Query/View/Transformation) qui propose des langages d'expression de règles.

3.2. Problématique

Notre problématique consiste à définir des mécanismes permettant d'implanter une base de données de type Big data sur une plateforme NoSQL. A partir d'un diagramme de classes UML décrivant des données complexes, nous devons spécifier les transformations nécessaires à l'élaboration d'un schéma NoSQL. Parmi les principaux types de systèmes NoSQL (Clé-valeur, Colonne, Document, Graphe) [1], nous avons choisi d'implanter les données sur un système orienté colonnes. Ce choix a été dicté par les besoins de nos applications basés sur des requêtes multicritères faisant intervenir simultanément plusieurs attributs. Or les systèmes orientés colonnes offrent des techniques de stockage qui sérialisent les valeurs des colonnes et permettent ainsi d'accélérer l'accès aux données. Le problème consiste donc à passer d'un schéma conceptuel de base de données (DCL – Diagramme de Classes d'UML) vers un schéma physique NoSQL qui fera l'objet d'une implantation. Mais plusieurs systèmes NoSQL orientés colonne coexistent ; les plus connus sont BigTable [6], HBase[3], Cassandra[4] et Accumulo[5]. Ils présentent des spécificités techniques propres qui relèvent essentiellement des techniques d'implantation. Pour faire abstraction de ces spécificités, nous intègrerons le niveau logique dans le processus de transformation des schémas. Autrement dit, nous considérerons les transformations successives : Conceptuel $\longrightarrow$ Logique puis Logique $\longrightarrow$ Physique. Au niveau logique, le schéma décrit l'implantation des données en faisant abstraction de considérations techniques propres à tel ou tel système NoSQL.

3.3. Etat de l'art

Une base de données de type Big data contient des données variées, c'est-à-dire des données de types non standard qualifiés généralement d'objets complexes : textes, graphiques, documents, séquences vidéo. Aujourd'hui, le modèle de données d'UML

[2] http://www.omg.org/spec/QVT/1.2/PDF/.
[3] https://hbase.apache.org/.
[4] http://cassandra.apache.org/.
[5] https://accumulo.apache.org/.

représente une sorte de référence en matière de représentation de schémas de bases de données complexes [2]. Ce modèle conceptuel, permettant de décrire la sémantique des objets métiers dans une application, peut donc être appliqué à la description des bases de données de type Big data.

En ce qui concerne les processus permettant d'implanter des bases de données sur des systèmes NoSQL, plusieurs études ont porté sur la transformation des schémas. Ainsi, dans le contexte des entrepôts de données, les travaux de Chevalier et al. [11] ont défini des règles pour traduire un modèle multidimensionnel en étoile, en deux modèles physiques NoSQL, un modèle orienté colonnes et un modèle orienté documents. Les liens entre faits et dimensions ont été traduits sous la forme d'imbrications. L'article de Li [4] a étudié l'implantation d'une base de données relationnelle dans le système HBase. La méthode proposée est basée sur des règles permettant la transformation d'un schéma relationnel en un schéma HBase ; les relations entre les tables (clés étrangères) sont traduites par l'ajout des familles de colonnes contenant des références. D'autres travaux ont étudié la transformation d'un diagramme de classes UML en un schéma de données HBase avec l'approche MDA [14]. L'idée de base est de construire des méta-modèles correspondant au diagramme de classes UML et au modèle de données orienté colonne HBase puis de proposer des règles de transformation entre les éléments des deux méta-modèles construits. Ces règles permettent de transformer un DCL directement en un schéma d'implantation spécifique au système HBase. Cet état de l'art montre que peu de travaux ont étudié la transformation d'un modèle conceptuel de données complexes vers un modèle NoSQL. Dans l'étude [14] la plus proche de notre problématique, les règles de transformation de schémas qui ont été adoptées, ne sont pas indépendantes d'une plateforme technique.

4. Transformation des schémas

Nos travaux visent à transformer un schéma conceptuel décrivant une source Big data en un modèle NoSQL orienté colonnes. Pour ce faire, nous proposons une approche dirigée par les modèles (MDA) qui fournit des métamodèles et des règles de transformation permettant le passage automatique du niveau conceptuel au niveau physique. La figure 2 montre un aperçu de notre contribution. D'une manière générale, une approche MDA repose sur les modèles CIM, PIM et PSM ainsi que les transformations automatiques entre ces modèles. Cependant, chaque approche présente ses propres caractéristiques, notamment le nombre et le type des modèles et des transformations [13]. Notre approche repose donc sur les deux niveaux PIM et PSM.

- le niveau PIM comporte deux niveaux différents ; le premier niveau présente le modèle conceptuel (Diagramme de classes UML) et le second présente le modèle logique (Modèle NoSQL orienté colonnes).

- le niveau PSM décrit les modèles physiques correspondants aux plateformes HBase et Cassandra.

Le passage d'un niveau à un autre se fait automatiquement grâce aux transformations M2M (Model-To-Model) formalisées en QVT[6] comme le montre la figure 2.

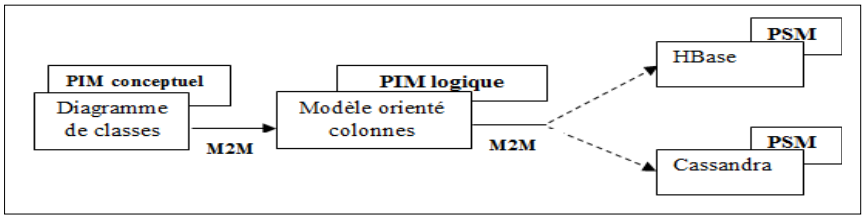


Figure 2. Les niveaux de modélisation

4.1. PIM conceptuel

UML étant le modèle reconnu par la communauté des bases de données pour représenter des objets complexes, nous décrivons le PIM conceptuel sous la forme d'un DCL. Avant d'assurer un passage automatique d'un DCL vers un modèle logique, nous devons préalablement formaliser les concepts présents dans le modèle de données d'UML. Un DCL contient un ensemble de classes. Chaque classe représente des objets ayant une sémantique et des propriétés communes ; elle est définie par son nom, ses attributs et ses opérations (dans cet article, nous prenons en compte uniquement la partie structurelle à l'exclusion des opérations). On distingue principalement quatre types de liens entre les classes : l'association, l'agrégation, la composition et l'héritage. A partir du métamodèle d'un DCL défini par l'OMG [15] et adapté à notre PIM conceptuel, nous présentons ces différents concepts à travers un métamodèle (Figure 3) que nous avons implanté sur la plateforme Eclipse Modeling Framewok[7] (EMF).

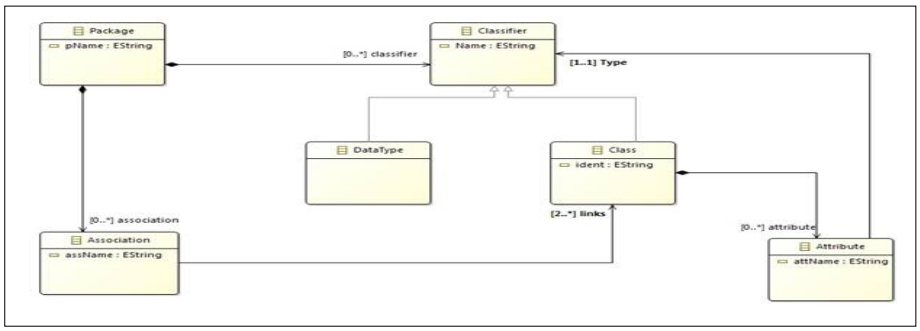


Figure 3. Méta-modèle source

4.2. Transformation automatique

4.2.1. PIM logique

4.2.1.1. Définition d'un modèle orienté colonnes

[6] http://www.omg.org/spec/QVT/1.2/PDF/.
[7] http://www.eclipse.org/modeling/emf/

Dans le niveau logique de description d'une base de données, les choix d'implantation ne sont pas complètement spécifiés. Les principes d'organisation des données sont précisés mais il est fait abstraction du SGBD utilisé pour implanter la base (ce choix se fait au niveau physique) ; seul le type du SGBD est pris en compte. Nous avons retenu un système NoSQL de type orienté colonnes. Selon ce modèle, une base de données (BD) est constituée d'un ensemble de tables. Une table permet de regrouper des objets de taille variable sous forme de lignes ; chacune d'elles est identifiée par un identificateur unique dont le type est noté clé-ligne. Généralement, on regroupe dans une table les objets fortement liés ; par exemple les employés, les services auxquels ils appartiennent et les projets auxquels ils participent. Par défaut, nous stockerons la base de données dans une table unique. Cette table, notée T, est associée à un ensemble de familles de colonnes F : $\{f_1, \ldots, f_p\}$. Une famille regroupe un nombre variable de colonnes f : $\{cl_1, \ldots, cl_q\}$ chacune d'elles est composée d'un nom, d'un type, d'une valeur et d'un horodatage (Timestamp) pour stocker plusieurs versions de la même donnée. Dans cet article, nous ne considérons pas l'horodatage des données. Nous présentons les concepts du PIM logique orienté colonnes à travers le métamodèle de la figure 4.

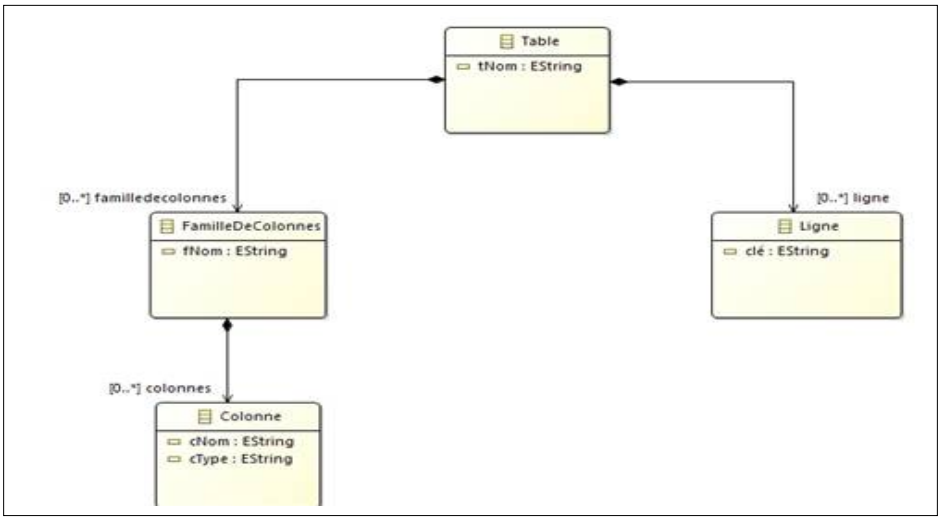


Figure 4. Méta-modèle cible

4.2.1.2. Règles de transformation

Nous proposons les règles suivantes en mettant en vis-à-vis un schéma conceptuel UML et sa traduction dans le PIM logique. Plusieurs solutions de transformation en PIM logique sont parfois possibles ; nous avons opté pour celles qui s'adaptent le mieux aux manipulations prévues dans notre application médicale.

- R1 : Package → Table, par défaut, les objets fortement liés sont regroupés dans une table ; c'est par exemple le cas des patients, des consultations médicales et des traitements prescrits par les médecins. Par analogie, la notion de Table est ici à rapprocher du concept

de paquetage dans UML qui permet de regrouper des éléments (classes, associations, composants, etc.) dans le but de constituer des ensembles d'objets sémantiquement liés.

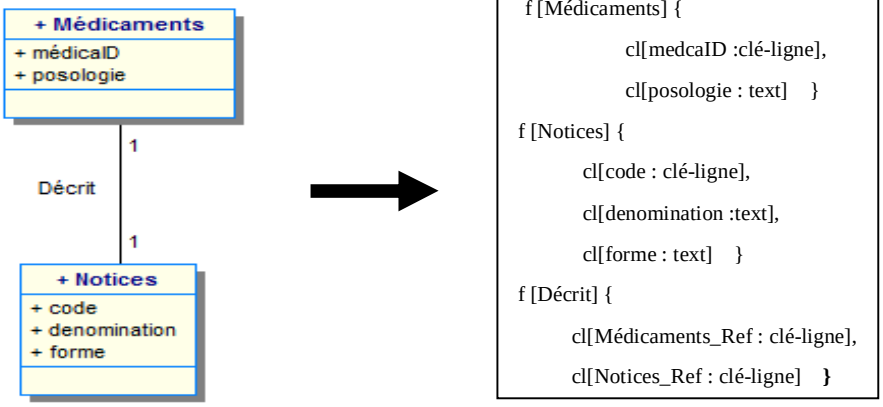
- R2 : Class → ColumnsFamily, les attributs de la même classe sont regroupés dans une seule famille ; le nom de la famille correspond au nom de la classe d'origine. Le stockage des données orienté colonnes sur le disque étant organisé par famille de colonnes, cette solution privilégie les requêtes qui utilisent simultanément les attributs de la classe.

- R3 : Oid →clé-ligne, les lignes décrites par une famille de colonnes représentent des objets (instances) de la classe correspondante. Ainsi, un identificateur de type « Oid » qui permet d'identifier les objets d'une classe est traité comme un identificateur de type « clé-ligne » qui permet d'identifier les lignes décrites par une famille de colonnes.

- R4 : Classe-Associations → ColumnsFamily, comme toute classe, une classe d'association est transformée en une famille de colonnes ; chaque colonnes correspond à un attribut de la classe soit à un attribut de type « Oid » qui référence une des classes liées.

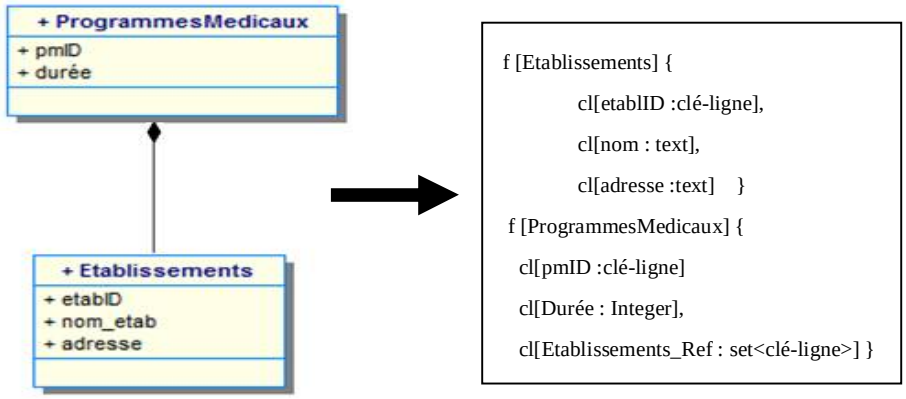
- R5 : Association →ColumnsFamily, chaque lien d'association n-aire se traduit par une nouvelle famille de n colonnes. Chaque colonne est associée à un type clé-ligne : une telle colonne référence une famille cible (classe liée). Ce principe de transformation s'applique à tout lien d'association quel que soit son degré (nombre de classes participantes) et quel que soit ses cardinalités. Nous avons eu ici le souci de généraliser le processus de traduction des liens d'association.

Exemple :



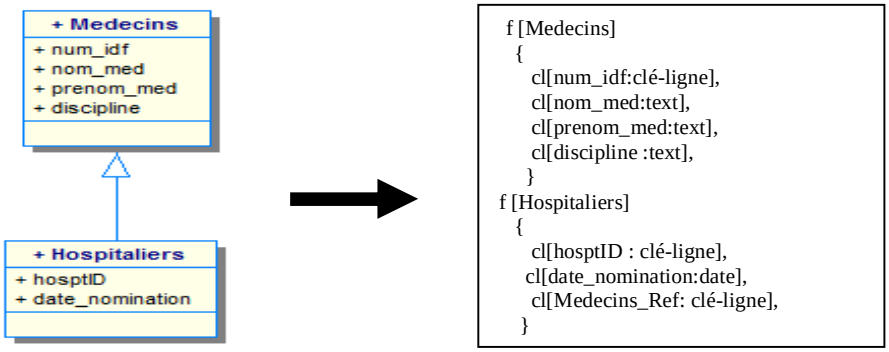
- R6 : Composition →NewColumn, une relation de composition est traduite soit par une imbrication soit par l'utilisation d'une référence. Comme le modèle orienté colonnes ne permet pas la présentation des données imbriquées (par exemple une colonne contenant d'autres colonnes), nous avons choisi l'utilisation des références. La composition/agrégation associe une classe composite et des classes composantes, tel que toute classe composante appartient à une et une seule classe composite. C'est donc une association 1..* (voire 1..1), nous la transformons alors comme suit : Tout lien de composition ou d'agrégation entre une classe composite et des classes composantes se traduit par l'ajout d'une nouvelle colonne de type « set clé-lignes » référençant les classes composantes dans la famille correspondante à la classe-composite.

Exemple :



- R7 : Héritage → NewColumn, nous proposons de transformer chaque lien d'héritage par l'ajout d'une nouvelle colonne de type « clé-ligne » dans la famille correspondante à la sous-classe ; cette colonne a pour rôle de référencer la super-classe.

Exemple :



La formalisation des règles précédentes avec QVT (Query/View/Transformation) permet un passage automatique entre le schéma conceptuel et le schéma logique de notre démarche. QVT est un langage déclaratif standardisé par l'OMG. Une transformation QVT entre deux modèles candidats est spécifiée grâce à un ensemble de relations. Chaque transformation est composée des éléments suivants : « Domains », « Relation Main », une clause « When » et une clause « Where ». Nous présentons ci-dessous les principales règles de transformation en QVT graphique.

Relation « Main ». Cette relation est le point d'entrée du processus de transformation. La partie gauche de la figure 4 montre les éléments du modèle source (uml : UML) transformés en éléments du modèle logique Orienté Colonne (OC : OrientéColonne) présenté par la partie droite de la figure. Un package est transformé en une table de même nom. La clause "where" spécifie que les classes et les associations sont transformées en familles de colonnes.

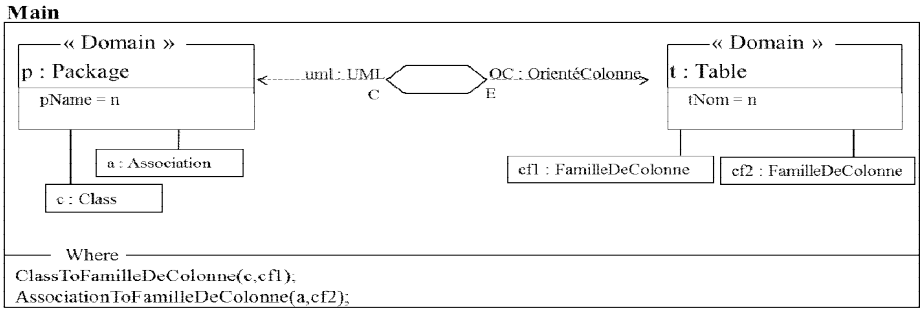


Figure 5. Relation Main de la transformation du PIM conceptuel en PIM logique

Relation « Classe en Famille de colonnes ». Une classe est transformée en une famille de colonnes dont le nom correspond au nom de la classe. Tous les attributs de la classe sont transformés en colonnes de la famille de colonnes en appliquant la relation « AttributeToColonne ». Les associations auxquelles participe la classe sont transformées en famille de colonnes.

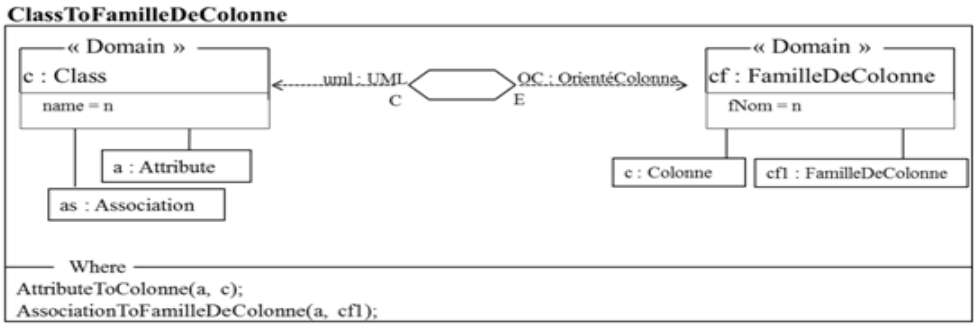


Figure 6. Relation de transformation de classe en famille de colonnes

Relation « Attribut en colonne ». Après avoir transformé les classes en familles de colonnes (relation "ClassToFamilleDeColonne" de la clause when), tous les attributs de cette classe sont transformés en colonnes de la famille ayant le même nom. Les types de ces attributs correspondent aux types des colonnes.

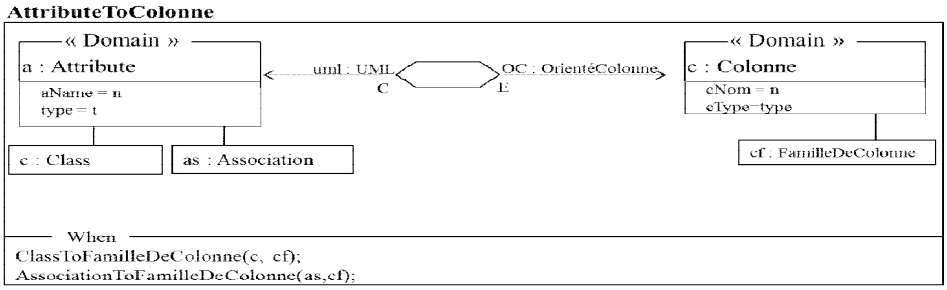


Figure 7. Relation de transformation d'attributs en colonnes

Relation « Associations en Famille de colonnes ». Une fois les classes qui participent à l'association transformées en famille de colonnes (la précondition « ClassToFamilleDeColonne » de la clause « When »), l'association est convertie en une famille de colonnes ayant le même nom. Les classes qui participent à cette association sont par la suite traduites en colonnes de la nouvelle famille de colonne correspondant à l'association (relation "ClassToColumn" de la clause "Where"). De même, les attributs de l'association sont aussi convertis en colonnes de la famille de colonnes.

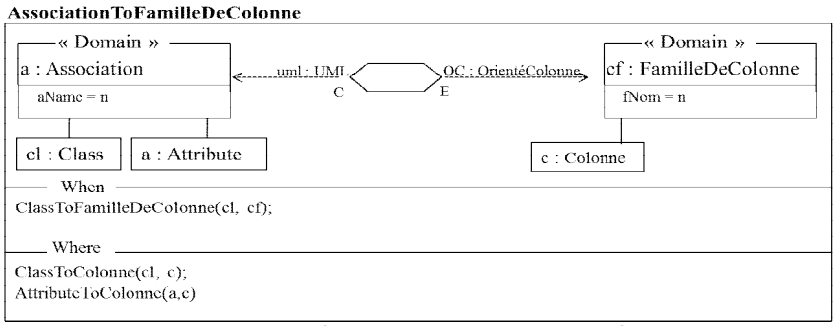


Figure 8. Relation de transformation d'association en famille de colonnes

4.2.2. Modèles physiques

Un PIM logique permet de générer plusieurs PSM associés à des plateformes distinctes ; ce principe assure l'indépendance du niveau logique face aux spécificités techniques des systèmes NoSQL ainsi qu'à leurs évolutions. Nous présentons brièvement les deux plateformes d'implantation : Cassandra et HBase, qui sont compatibles avec le PIM logique proposé. Mais, dans la mesure où cet article est consacré à la transformation du PIM conceptuel vers le PIM logique, nous ne décrivons pas le passage du PIM logique vers les PSM.

4.2.2.1. PSM HBase

HBase est un système NoSQL orienté colonnes et développé au-dessus du système de fichiers HDFS (Hadoop Distributed File System) de la plateforme Hadoop [12]. Une base

de données HBase est par défaut composée d'une seule table notée HTable (l'administrateur peut modifier ce paramètre pour créer plusieurs tables). Une HTable est associée à un nombre fixe de familles de colonnes devant être spécifiées à la création de la HTable. Seul le nom de la famille est précisé sans mention des noms de colonnes. Chaque famille est un regroupement logique de colonnes qui seront ajoutées au moment de l'insertion des données. Chaque ligne (ou enregistrement) au sein d'une HTable est identifiée par une clé notée RowKey et choisie par l'utilisateur. Au triplet (RowKey, famille de colonnes, colonne) correspond une cellule unique qui contiendra une valeur.

4.2.2.2. PSM Cassandra

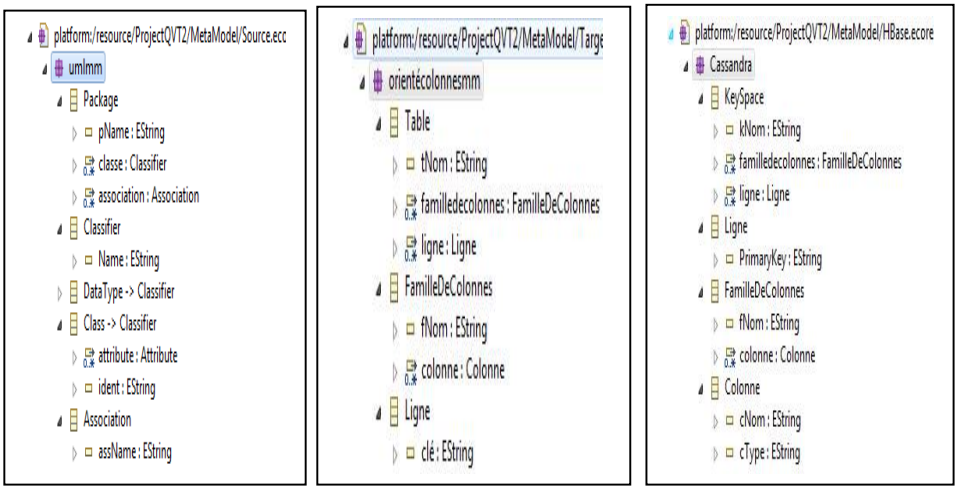
Cassandra est un SGBD NoSQL orienté colonnes, initialement basé sur le modèle BigTable de Google, mais qui emprunte également des caractéristiques au système Dynamo d'Amazon[8]. Une base de données Cassandra est par défaut composée d'un seul conteneur de données noté KeySpace. Ce dernier est associé à une ou plusieurs familles de colonnes, chacune d'elles est un regroupement logique de lignes. Une ligne est composée d'un ensemble de colonnes et est identifiée par une clé notée PrimaryKey. Chaque colonne est représentée par un quadruplet correspondant à un nom, un type, une valeur et un timestamp.

Les concepts « Table » et « colonne de type clé-ligne » vont correspondre respectivement aux concepts HTable et RowKey sous HBase et par KeySapce et PrimaryKey sous Cassandra.

5. Implantation

Dans cette section, nous décrivons les techniques que nous avons utilisées pour mettre en œuvre la démarche présentées dans la figure 2. Etant donné que notre approche est dirigée par les modèles, nous avons utilisé un environnement technique adapté à la modélisation, la métamodèlisation et la transformation des modèles. Nous avons eu recours à la plateforme Eclipse Modeling Framework (EMF) qui utilise Ecore pour créer et manipuler les modèles. La figure 9 illustre les métamodèles Ecore utilisés par notre module de transformation : PIM conceptuel (a), PIM logique (b) et PSM Cassandra (c).

[8] https://aws.amazon.com/fr/documentation/dynamodb/



(a) ***(b)*** ***(c)***

Figure 9. Métamodèles implantés avec Ecore

Notre choix du langage de transformation a été fondé sur des critères spécifiques à notre démarche. En effet, l'outil doit être intégré dans l'environnement EMF pour qu'il soit utilisé aisément avec les outils de modélisation et de métamodélisation. Ainsi, nous avons utilisé le langage QVT opérationnel. La figure 10 montre un extrait de code QVT assurant la génération du PIM logique et du PSM Cassandra à partir du PIM conceptuel ; les commentaires figurant dans le code précisent les règles utilisées.

```
modeltype UML uses "http://umlmm.com";
modeltype OrientéColonnes uses "http://orientécolonnesmm.com";
transformation TransformationUmlToNoSQL(in Source: UML, out Target: OrientéColonnes);

main() {
Source.rootObjects()[Package] -> map PackageToTable();
}

mapping Package::PackageToTable():Table{
tNom := self.pName;

familledecolonnes:=self.classes -> map toColumnFamilyC();

familledecolonnes:=self.association -> map toColumnFamilyA();
} -- Transformation de Package en Table

 mapping UML::Class::toColumnFamilyC():OrientéColonnes::FamilleDeColonnes{
     fNom:=self.Name;
     colonne:=self.attribute -> map toColumnC();
}
 mapping UML::Attribute::toColumnC():OrientéColonnes::Colonne{
     cNom:=self.attName;
     cType:=self.attType;
} -- Transformation de Classes en Familles de colonnes

mapping UML::Association::toColumnFamilyA():OrientéColonnes::FamilleDeColonnes{
     fNom:=self.assName;
     colonne := self.links -> toColumnA()
}
mapping UML::Class::toColumnA():OrientéColonnes::Colonne{
     cNom:=self.Name + ".Ref";
     cType:="clé-ligne";
} -- Transformation d'Associations n-aire en Familles de colonnes
```

Figure 10. Extrait du code QVT

L'étape suivante vise à instancier le métamodèle du PIM conceptuel ; un exemple de cette instance est présenté dans la figure 11.

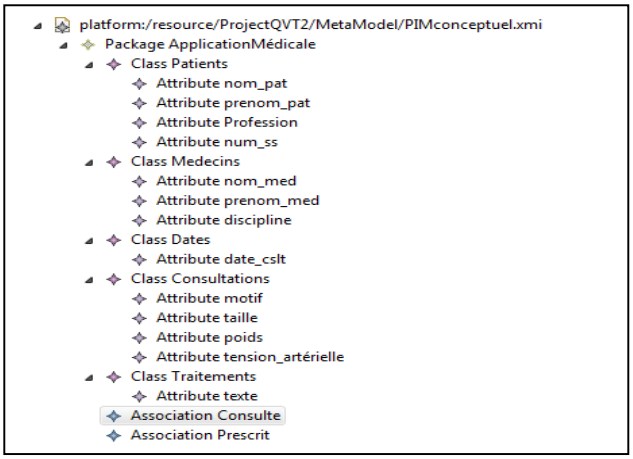
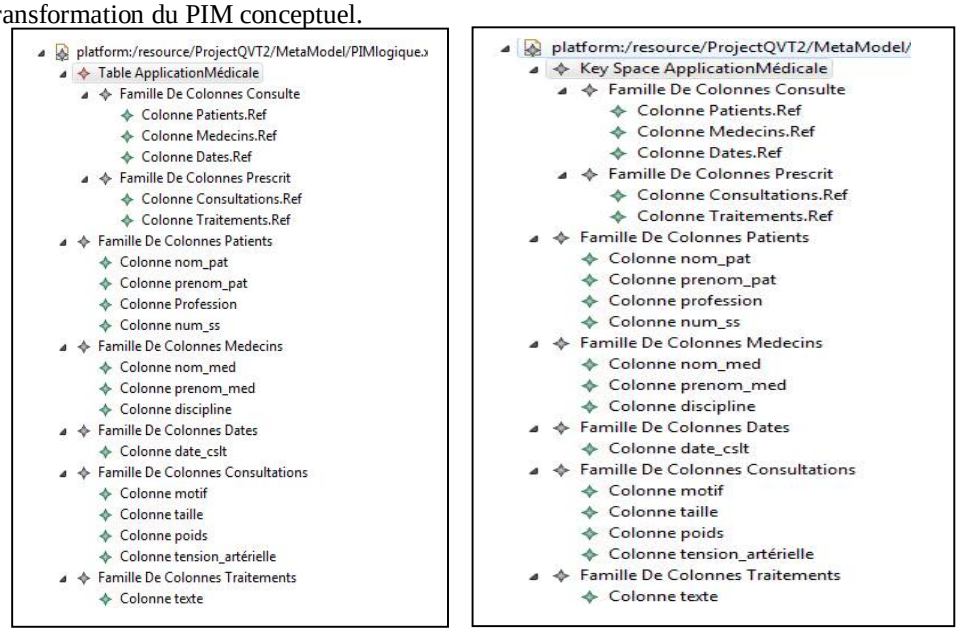


Figure 11. PIM conceptuel entré par l'utilisateur

La figure 12 montre le PIM logique (a) et le PSM Cassandra (b) obtenus par transformation du PIM conceptuel.



(a) ***(b)***

Figure 12. Modèle générés par le système : PIM logique (a) et PSM Cassandra (b)

6. Positionnement de nos travaux

Nos travaux traitent de la transformation d'un schéma conceptuel des données, représenté par un DCL d'UML, en un schéma NoSQL orienté colonnes. Nous positionnons nos travaux au regard de trois articles de recherche dont les problématiques et/ou les solutions proposées sont proches des nôtres. L'article de Chevalier et al. [11] s'inscrit dans le contexte de l'entreposage des données puisqu'il étudie les règles de passage d'un schéma multidimensionnel en un schéma physique ; deux plateformes NoSQL ont été retenues : le système orienté colonne HBase et le système orienté document MongoDB. Bien que le point de départ du processus (un schéma multidimensionnel) se situe au niveau conceptuel, ce schéma ne présente pas les mêmes caractéristiques qu'un DCL d'UML ; notamment, il comporte exclusivement des classes Faits et Dimensions et un type de lien unique entre ces deux classes. L'article de C. Li [4] traite de la transformation d'un schéma relationnel en un schéma orienté colonne HBase. Ces travaux répondent bien aux attentes concrètes des entreprises qui, face aux évolutions récentes de l'informatique, souhaitent stocker leurs bases de données actuelles dans des systèmes NoSQL. Mais la source du processus de transformation, ici un schéma relationnel, ne présente pas la richesse sémantique que l'on peut exprimer dans un DCL (notamment grâce aux différents types de liens entre classes : agrégation, composition, héritage, …). Les travaux de Y. Li et al. présentés dans [14] ont pour objet de spécifier un processus de transformation MDA d'un schéma conceptuel (DCL) vers un schéma physique HBase. Ce processus ne propose pas un niveau intermédiaire (le niveau logique) qui permettrait de rendre le résultat indépendant d'une plateforme particulière.

7. Conclusion

Nos travaux s'inscrivent dans le cadre de l'évolution des bases de données vers les Big Data, ceci pour prendre en compte le volume, la variété et la vélocité des données présentes dans les nouvelles applications liées à la transformation digitale des entreprises. Nos études portent actuellement sur les mécanismes de stockage des données dans des systèmes NoSQL. Dans cet article, nous avons traité du processus de transformation d'un schéma conceptuel représenté par un DCL d'UML en un schéma physique NoSQL orienté colonne. Pour automatiser ce processus, nous avons utilisé l'approche MDA pour créer des transformations successives entre un DCL, un schéma logique NoSQL et un schéma physique spécifique à une plateforme NoSQL. Selon notre approche, le schéma logique constitue un niveau intermédiaire qui fait abstraction de considérations techniques propres aux plateformes d'implantation et qui apparaîtront uniquement dans le schéma physique ; ce principe permet de rendre le niveau logique indépendant des évolutions technologiques des plateformes. Le point de départ du processus de transformation MDA est un métamodèle de DCL proposé par l'OMG. Les règles de transformation basées sur ce métamodèle et qui permettent de produire un schéma logique NoSQL, ont été exprimées en langage QVT. Nous avons expérimenté notre démarche et nos modèles sur une application du domaine médical qui porte sur des programmes pluriannuels de suivi de pathologies. Nous avons automatisé le processus de transformation d'un DCL décrivant une base de

données en un schéma NoSQL orienté colonne. Ce schéma a été implanté sur les systèmes HBase et Cassandra. Actuellement, nous poursuivons nos travaux en prenant en compte les spécificités du modèle de données dans les systèmes NoSQL orientés graphes.

Bibliographie

[1] Abhinay B. Angadi, Akshata B. Angadi, Karuna C. Gull. “Growth of New Databases & Analysis of NOSQL Datastores”. International Journal of Advanced Research in Computer Science and Software Engineering. 2013.

[2] A. Tanasescu, O. Boussaïd, F. Bentayeb. “Preparing Complex Data for Warehousing”. 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 05). Cairo, Egypt, January 2005.

[3] B.Combemale. “Ingénierie Dirigée par les Modèles (IDM) - Etat del'art” .2008.

[4] C. Li. “Transforming relational database into HBase : A case study”. In International Conference on Software Engineering and Service Sciences (ICSESS), pp. 683–687. IEEE.2010.

[5] C. L. P. Chen, C. Zhang. “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data”. Inf. Sci., 275, 2014.

[6] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T, Chandra, A. Fikes, R.E. Gruber. “Bigtable: a distributed storage system for structured data,”. ACM Trans. Comput. Syst. 26(2), 2008.

[7] J. Darmont, O. Boussaid, J. Ralaivao, K. Aouiche. “An Architecture Framework for Complex Data Warehouses”. 7th International Conference on Enterprise Information Systems (ICEIS 05), Miami, USA, May 2005.

[8] J.Han, E. Haihong, G. Le, J. Du.“Survey on NoSQL Database”. Pervasive Computing and Applications (ICPCA), 6th International Conference on, 2011.

[9] J. Miller, J. Mukerji. Model Driven Architecture (MDA) 1.0.1 Guide. Object Management Group, Inc., June 2003.

[10] META Group (devenu Gartner). “3D Data Management: Controlling Data Volume, Velocity, and Variety”. February 2001.

[11] M. Chevalier, M. El Malki, A. Kopliku, O. Teste, R. Tournier. “Entrepôts de données multidimensionnelles NoSQL”. EDA 2015, 161-176.

[12] M. Grover, T. Malaska, J. Seidman, G. Shapira. “Hadoop Application Architectures”. O'Reilly, 2015.

[13] X. Blanc, O. Salvatori "MDA En Action : Ingénierie Logicielle Guidée Par Les Modèles". Paris : Eyrolles, 2005.

[14] Yan Li, Ping Gu, Chao Zhang. “Transforming UML Class Diagrams into HBase Based on Meta-model. Information Science”. Electronics and Electrical Engineering (ISEEE), 2014.

[15] http://www.omg.org/spec/QVT/1.1/.

Session Processus Métier

Gestion Intégrée du Changement des Modèles de Processus Métier

Mourad Bouneffa, Adeel Ahmad, Henri Basson

Université du Littoral Côte d'Opale
Laboratoire d'Informatique Signal et Image de la Côte d'Opale
50, rue Ferdinand Buisson - BP 719, 62228 CALAIS CEDEX*, FRANCE*
bouneffa,ahmad,basson@lisic.univ-littoral.fr

RÉSUMÉ. *Ces dernières années, les modèles de processus métier ou BPM, ont été utilisés comme des entités servant dans la spécification des différents processus d'une organisation. Des ateliers existent, permettant même de générer des systèmes informatiques exécutables sur des plates-formes distribuées à partir de modèles de processus métier. Ceci a été facilité par la généralisation de la notion d'architectures orientées service ou SOA. Nous proposons une approche à base de graphes et de systèmes de réécriture de graphe pour à la fois modéliser et simuler la propagation de l'impact du changement des différents constituants d'une application basée sur la mise en oeuvre des modèles de processus métier. Nous adoptons également l'utilisation des ontologies dans le but de représenter la sémantique des relations reliant les différents constituants de ce type d'applications. Cette connaissance sémantique sera alors utilisée par un algorithme permettant de générer des règles de réécriture de graphes implémentant la propagation de l'impact du changement de ce type d'applications.*

ABSTRACT. *For the past few years, the Business Process Models or BPMs, have been largely used to specify the different processes of an organisation. The existing frameworks, may also generate executable information systems, deployed on distributed platforms, from the instantiation of business process models. This has been facilitated by the generalization of the so called Service-Oriented Architecture (SOA). We propose a graph-based approach, along with the graph re-writing system, for the modeling and simulation of the change impact propagation among the different constituents of the instantiated business process models. We use ontologies for the semantic representation of the relations linking the different constituents of such applications. This semantic knowledge is then used by an algorithm that may generate the graph re-writing rules to incorporate the change impact propagation in such applications.*

MOTS-CLÉS : *Modèles de Processus Métier (BPM), Propagation de l'impact du changement, Systèmes de réécriture de graphe, Ontologies.*

KEYWORDS: *Business Process Model (BPM), Change impact propagation, Graph re-writing systems, Ontology.*

Congrès INFORSID, 34ème édition, Grenoble, 31 mai – 3 juin 2016

1. Introduction

Les modèles de processus métier (Business Process Models ou BPMs) ont servi, dans un premier temps, comme formalisme dans le cadre du management des organisations et plus particulièrement dans le cadre de la réingénierie des processus métier (Business Process Reengeneering) (Reijers *et al.*, 2016). La dernière décennie a vu l'extension de l'utilisation de ces formalismes au domaine du développement et du déploiement des applications collaboratives souvent distribuées. Ainsi les BPMs sont passés au stade d'abstractions de haut niveau ayant un lien avec des composants informatiques exécutables les implémentant et les déployant sur des plates-formes souvent distribuées. Ceci a fortement été encouragé par l'émergence des architectures dites orientées service (SOA). Ainsi, on assiste à l'emergence d'approches et d'outils pour le développement de logiciels d'entreprise distribués dans lesquels les BPMs sont transformés en entités exécutables. Dans ces approches, les BPMs sont spécifiés à l'aide de notations standards telles que BPMN (Business Process Model Notation) [1] et XPDL [2]. Ils sont, par la suite, transformés en programmes exécutables déployés sur des plates-formes distribuées en forme d'applications multi-tiers (Java J2EE, .NET, etc.). Ces programmes exécutables sont également souvent construits comme des macro-programmes implémentant ce qui est communément appelé *the programming in the large*. Ces programmes contiennent des invocations de services web (Gottschalk *et al.*, 2002) fournis par diverses applications déployées à l'intérieur et parfois à l'extérieur des frontières du système d'information concerné. BPEL (Business Process Execution Language) (Juric, 2006) est l'un des langages les plus connus en matière de *programming in the large.* Il permet la composition ou plutôt l'orchestration de services web dans le but de réaliser une fonctionnalité donnée pouvant correspondre aux actions formant une activité d'un BPM. Ce type d'approches peut contribuer à l'élimination du gap existant entre les BPMs et leur implémentation ou déploiement en termes de composants IT (Information Technology). Il devient alors possible d'imaginer une gestion intégrée des changements d'un BPM en répercutant les effets de ces changements sur les composants IT correspondant et inversement.

La notion de changement est intrinsèquement liée aux applications basées sur le BPM. Comme nous le montrerons dans la section 2, le cycle de vie de ces applications est une succession d'itérations ayant pour but l'amélioration de critères de qualité et de performance formant le tableau de bord des décideurs aussi bien au niveau fonctionnel que technique. Cette question est donc centrale et a conduit à de nombreux travaux reposant généralement sur la notion de patterns ou patrons de changement des BPMs (Kim *et al.*, 2007). Dans ce papier, nous nous intéressons à un problème particulier qui est l'analyse et à la propagation de l'impact du changement de ce type d'applications. Nous privilégions l'utilisation des règles de réécriture de graphes dans le but de pouvoir automatiser leur génération et donc d'automatiser le processus d'identification de l'impact du changement. Les graphes et plus spécifiquement les graphes

1. http://www.omg.org/spec/BPMN/2.0
2. http://www.xpdl.org

typés et attribués ont, depuis longtemps, servi comme structures de prédilection dans le cadre de la gestion de l'évolution du logiciel. Cela est dû à la nature du logiciel qui est constitué d'une grande collection d'artefacts de différents types (représentables par des nœuds) reliés par divers types de relations. Dans ce cadre, les systèmes de réécriture de graphes ont également été expérimentés depuis de nombreuses années aussi bien dans la problématique de propagation de l'impact du changement (Rajlich, Gosavi, 2004 ; Maweed *et al.*, 2005) qu'à celui du refactoring de modèles (Folli, Mens, 2007 ; Mens, 2005), etc. Ces systèmes fournissent un outil de raisonnement, de spécification et de mise en oeuvre visuelle (donc simplifiée) des différentes tâches rentrant dans le cadre de l'évolution du logiciel et nécessitant une prise en compte des différents liens inter-artefacts. Ils permettent, entre autres, d'éviter le phénomène *d'impedence mismatch* qui est généré par une utilisation de structures hétérogènes pour la manipulation des artefacts logiciels. C'est le cas notamment quand ces artefacts sont stockés dans des bases de données relationnelles, transformés après en faits pour être manipulés par un système d'inférence et en graphes pour être visualisés. Notre choix des systèmes de réécriture de graphes pour la gestion de l'évolution d'applications reposant sur les processus métier s'inscrit dans ce cadre. Notre but est également d'assurer une certaine inter-opérabilité entre les travaux que nous avons déjà menés dans le cadre de l'évolution du logiciel et ceux concernant plus spécifiquement les applications basées sur les processus métier. Nous avons donc expérimenté la faisabilité de l'utilisation des systèmes de réécriture de graphes dans le cadre de l'analyse *a priori* et la propagation de l'impact du changement des applications orientées BPM (Bouneffa, Ahmad, 2013b ; 2013a). Dans le but d'une automatisation de la génération des règles de réécriture de graphes et donc d'une automatisation de la génération du processus d'analyse et de propagation du changement, nous avons été confrontés à une limitation des systèmes de réécriture de graphes. En effet, pour une telle automatisation il était nécessaire d'expliciter des connaissances d'ordre sémantique sur les artefacts et plus particulièrement sur les relations inter-artefacts. Pour cela, nous avons introduit, dans ce papier, une ontologie qui explicite certaines connaissances utiles telles que la manière dont une relation conduit l'impact d'un changement, etc. Bien que la notion d'impact soit étroitement liée à la violation de propriétés de cohérence des processus métier, dans ce papier, nous n'abordons pas de façon approfondie la problématique de vérification des propriétés des processus que nous avons déjà traitée dans (Kherbouche *et al.*, 2013).

La suite du papier est organisée comme suit : la section 2 décrit les principaux constituants des applications centrées BPM. La section 3 décrit le méta modèle associé aux BPMs et BPEL et qui est basé sur l'utilisation de la notion de graphes attribués et typés enrichis par des connaissances sémantiques explicités par une ontologie. En effet, notre approche étant basée sur l'utilisation des systèmes de réécriture de graphes pour la mise en oeuvre et l'analyse de l'impact des changements des applications centrées BPM, il était évident que le méta modèle emprunte le vocabulaire issus de ces systèmes et repose sur l'utilisation des types de graphes. La section 4 décrit les opérations de changement, leur exécution et le processus d'identification et de propagation de l'impact de ces changements à l'aide des règles de réécriture de

graphes. Elle décrit également l'utilisation des ontologies dans le cadre de la mise en oeuvre d'un algorithme automatisant la génération des règles de réécriture de graphes pour la propagation de l'impact du changement. La section 5 illustre l'intégration de notre approche dans le cadre d'un atelier que nous développons depuis de nombreuses années et qui est dédié à la gestion de l'impact des applications distribués. La dernière section conclut le papier en faisant le bilan des résultats obtenus et en y esquissant ses perspectives.

2. Les applications centrées sur le BPM

Le cycle de vie des applications développées et déployées comme une instantiation exécutable de BPMs est une succession d'itérations formées chacune de quatre principales phases : *la modélisation, le déploiement, l'exécution et le monitoring* et *l'amélioration*. La phase de modélisation du BPM en termes de tâches ou activités est un préalable à l'implémentation d'un processus. Elle explicite, entre autres, l'ordre d'exécution des tâches, les acteurs humains et éventuellement les données nécessaires à l'accomplissement de ces tâches. Plusieurs modèles ou notations ont été introduits pour la représentation des processus métier. Les premières notations étaient destinées à produire des modélisations utilisables dans le cadre des projets de réingénierie des processus métier (BPR : Business Process Reingineering). Dans ce papier, nous considérons particulièrement les modèles utilisés dans le cadre de l'automatisation des processus métier. En d'autres termes, nous nous plaçons dans le cadre de l'utilisation des BPMs comme une abstraction de haut niveau destinée à être transformée en une application déployable sur une architecture informatique distribuée. Nous avons choisi le langage qui semble être reconnu comme un standard en la matière, en d'autres termes BPMN (Business Process Model Notation). FIGURE 1 montre un exemple de processus métier exprimé selon la notation BPMN et représentant une partie d'une chaîne de vente. La figure met en évidence deux principaux acteurs de cette chaîne et qui sont le *Client* et le *Processus de gestion de commande*. Au démarrage, un événement de début *(start event)* associé au client permet de considérer l'activité *Place Order* comme la première activité démarrant le processus. Cette activité génère une commande qui est transmise par courrier ou message à l'activité *Check Availability* qui vérifie la disponibilité du produit commandé. Cette dernière est reliée à un nœud de type *GateWay* ou passerelle qui selon le résultat de l'activité réalise un branchement vers l'activité *Check Payment* (dans le cas où le produit commandé est disponible) ou l'activité *Cancel Order* dans le cas contraire. L'activité *Check Payment* est elle même reliée en sortie à un nœud GateWay qui aiguille respectivement vers les activités *Confirm Order* ou *Cancel Order* selon que le paiement soit validé ou pas. Ces deux dernières activités sont reliées chacune à un événement de type *Message* mais également terminal (représenté en gras).

Le développement et le déploiement des applications centrées BPM sont deux activités séparées pouvant être effectuées de façon manuelle, semi-automatique ou automatique. En principe ces deux activités peuvent être comme des opérations classiques de génération et de déploiement de code où le BPM joue le rôle de conception dé-

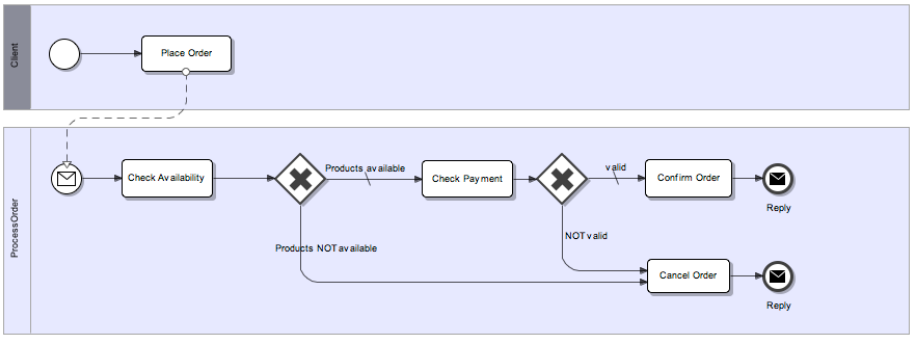


FIGURE *1. Exemple de processus métier en notation BPMN*

taillée et où le code est matérialisé par une application généralement déployée en forme d'application Web. Il existe, depuis quelques années, des outils automatisant le déploiement de ce type d'applications. C'est le cas notamment de Bonita [3], Intalio [4], BizAgi [5] et Barium Live! [6], etc. Ces outils permettent, à partir d'un BPM, la génération d'une application web de façon automatique et quasi transparente. Ces applications sont constituées de pages web dynamiques pouvant être des pages JSP, des ASP.NET, etc. Indépendamment de la question de l'automatisation de la génération de l'application implémentant un BPM, il existe des langages ayant pour but la mise en oeuvre sous la forme d'un programme *in the large*, d'orchestrations de services concourant à la réalisation effective de ce qui est modélisé dans le BPM. C'est le cas notamment du langage BPEL (Business Proces Execution Language) (Juric, 2006). Dans (Bouneffa, Ahmad, 2013a), nous décrivons une implantation de l'exemple de FIGURE 1 à l'aide d'une orchestration de services web BPEL contenant principalement des invocations de services web qui constituent les activités de base et des structures représentant des séquences d'invocations, ou des strutures conditionnelles et itéartives, etc.

L'exécution d'un BPM permet de recueillir certaines informations concernant les temps d'exécution, la consommation de ressources, etc. Ces données présentent un intérêt dans le cadre de l'étude des performances et en général de la qualité des processus mis en oeuvre. D'un autres côté, les experts métier mettent en place un certain nombre d'indicateurs de performance ou KPI (Key Performance Indicators). Cela permet de renseigner sur, par exemple, le temps moyen d'attente avant l'accomplissement d'une tâche donnée, etc. Certaines données concernant les KPI peuvent être obtenues par un

3. Bonita Open Solution url: http://www.bonitasoft.com/
4. IntaliolBPMS : http://www.intalio.com/
5. Bizagi BPM Suite : http://www.bizagi.com/
6. Barium Live! : http://www.bariumlive.com/

profiling de l'exécution de l'application (Ahmad *et al.*, 2009). D'autres informations ne peuvent être fournies que manuellement par un expert métier.

L'amélioration du BPM est un terme générique se référant à la notion d'évolution des processus du BPM. En réalité, l'amélioration attendue est le résultat de changements pouvant affecter chaque constituant d'une application centrée BPM et qui peut aussi bien être un élément du modèle de processus qu'un service web ou tout autre artefact logiciel rentrant dans la composition de l'application. Le but de tels changements peut être la correction d'erreurs ou d'anomalies constatées, un alignement avec des changements affectant le processus métier (changement de réglementation, réingéniérie du processus, etc.), etc. Dans la littérature, le changement est vu du point de vue du processus métier et concerne très rarement la composante logicielle implémentant ces processus.

3. Un méta modèle de BPM et de BPEL à base de graphes

Nous proposons un méta modèle représentant les concepts impliqués dans la définition des BPMs en utilisant la notion de graphes attribués et typés et les ontologies. L'utilisation des graphes typés et attribués est une conséquence de l'adoption dans tous nos travaux de la notion de graphes pour représenter tous les artefacts logiciels et leurs différents liens sémantiques et cela quelle que soit la nature de ces artefacts (programmes, schémas de bases de données, BPMs, etc.). Cela est également rendu nécessaire par le fait que nous préconisons l'utilisation des systèmes de réécriture de graphes pour l'implémentation des règles régissant l'identification et la propagation de l'impact du changement. L'utilisation des ontologies est arrivée comme une conséquence d'un besoin en matière d'explicitation de connaissances sémantiques supplémentaires et nécessaires, notamment, à la mise en œuvre d'un processus de génération automatique de règles de propagation de l'impact. Dans ce papier, nous nous sommes cantonnés à des connaissances concernant la conductivité de l'impact par les différentes relations reliant les artefacts d'une application basée BPM en utilisant le langage OWL [7] et la plate-forme *Protégé* [8]. Nous avons donc profiter du fait que OWL permette une hiérarchisation des relations par le lien ISA (FIGURE 2) pour définir 5 sous-types de relations selon le sens dans lequel ces relations conduisent l'impact à savoir : de la source ou domaine vers la destination ou co-domaine pour les relations *ForwardImpact* et inversement pour les relations *InverseImpact*. Nous avons également considéré le fait que l'impact soit certain ou conditionnel et également le fait qu'il n'y ait pas d'impact. Ces types de relations ne sont pas tous disjoints. Cela veut dire qu'une relation peut être de type *ForwardImpact* et *InverseImpact*. Une relation peut également être de type *ForwardImpact* et *CertainImpact* comme elle peut être *InverseImpact* et *ConditionalImpact*. Par exemple, une relation qu'on appellera *ImplementedBy* entre un BPM et le BPEL l'implémentant est aussi bien *ForwardImpact*

7. https://www.w3.org/2004/OWL
8. http://protege.stanford.edu

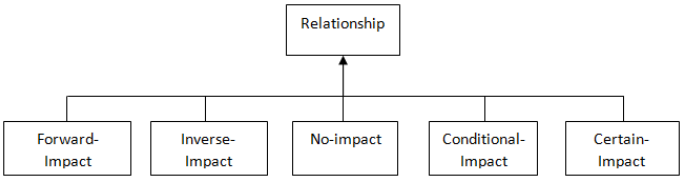


FIGURE 2. *Hiérarchie des classes de relations selon la propagation de l'impact*

que *InverseImpact*. En effet, le changement affectant un BPM peut affecter le BPEL l'implémentant. Il en est de même du changement affectant un BPEL par rapport au BPM correspondant. Cependant cette relation est *ConditionalImpact* puisque le changement d'un BPM ne conduit pas forcément à un changement affectant le BPEL correspondant et inversement.

Le méta modèle à base de graphes typés et attribués a été formalisé dans le cadre de la plate-forme AGG [9] dédiée à la transformation ou réécriture de graphes. Cette méta-modélisation est schématisée par FIGURE 3 qui présente les principaux concepts introduits par la notation BPMN. Le concept de *Processsus* représente les processus métier qui peuvent contenir des objets de flots. Ces derniers peuvent être des *tâches* ou activités, des *sous-processus* ou macro tâches raffinées par des processus, des *passerelles* (gateways), des *événements* ou des artefacts utilisés par les tâches tels que les documents, les données, etc. Un processus est associé à un acteur qu'on appellera *owner* représentant un ou plusieurs utilisateurs l'ayant défini et ayant le droit de le changer. Un processus est implémenté par une application pouvant héberger une ou plusieurs instances d'exécution de ce processus. Chaque instance fait intervenir des acteurs qui interagissent avec les différentes tâches exécutées durant le cycle de vie de l'instance. Le formalisme à base de graphes typés et attribués peut être vu comme un moyen permettant, entre autres, la vérification de la consistance des artefacts d'un BPM qui sont eux même formalisés par des graphes. Parmi les relations intéressantes notamment pour l'analyse de l'impact du changement, on considèrera les relations *input* et *output* associées à des objets de flots. En effet la relation *input* spécifie les objets de type *Tâche*, *Sous Processus* ou *Passerelle* qui constituent les entrées ou objets en amont d'un objet donné alors que la relation *output* spécifie les objets en sortie d'un objet donné. Ainsi, lors du changement d'un objet, on pourra, par le biais de la relation *output* , retrouver les objets en aval au niveau du flot de contrôle et par *input*, identifier les objets en amont de l'objet qui vient d'être modifié.

Le méta-modèle de FIGURE 3 représente la couche processus. Dans notre travail d'analyse de l'impact du changement nous avons également comme objectif la propagation de l'impact au niveau des artefacts d'un processus exprimé en BPMN vers

9. http://user.cs.tu-berlin.de/~gragra/agg/

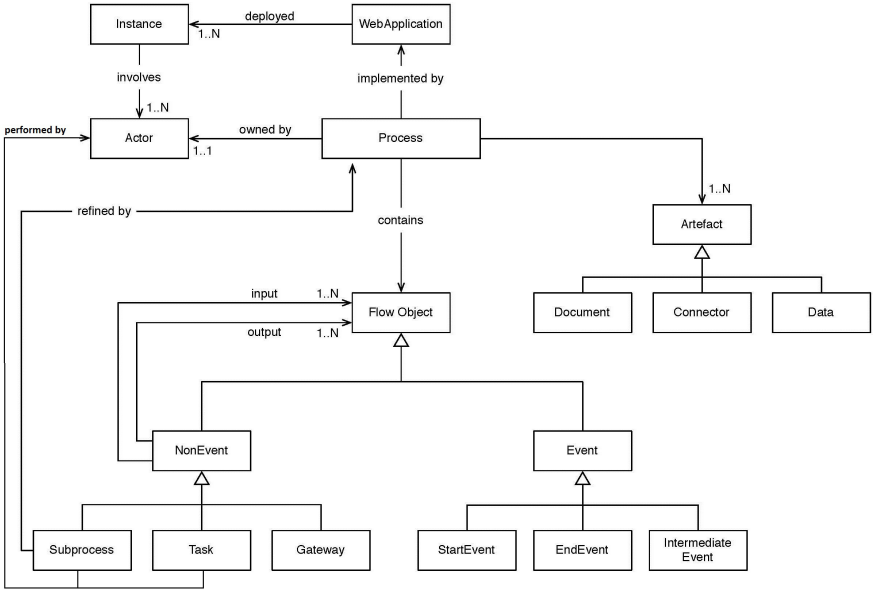


FIGURE *3. Métamodélisation des BPMs à base de graphes typés et attribués*

ceux de la couche service constitués d'orchestrations de services web. Le méta modèle de BPEL représente un processus BPEL comme étant composé d'activités qui peuvent être simples (*Basic Activity*) ou composites (*Structured Activity*). Une activité simple correspond à une invocation d'un service web (*Invoke*), l'attente de la réception d'un message (*Receive*), la sortie du processus (*Exit*) ou une activité vide (*Empty*) consistant à ne rien faire. Une activité composite consiste généralement en une orchestration d'activités simples ou composites et cela par le biais de structure telle que *Sequence* qui consiste à ordonnancer l'exécution de plusieurs activités en séquentiel, *Flow* qui correspond à une exécution parallèle de plusieurs activités, *Switch* qui est une exécution conditionnelle, etc. Dans ce papier, nous ne traitons pas du processus de transformation d'un BPMN en BPEL, mais nous partons du fait que des mappings existent et nous les matérialisons par deux relations appelées *mappedTo* et *ImplementedBy* et que nous exploitons dans le processus d'analyse et de propagation de l'impact qu'on appelle vertical et qui se propage d'une couche plus abstraite d'une application vers sa concrétisation ou implémentation et inversement. Ces relations s'expriment au niveau des graphes représentant les processus par un arc de type *ImplementedBy* dont la source est un artefact BPMN et la destination un artefact de BPEL et un lien inverse de type *mappedTo*. Un mapping peut exister entre un processus BPMN et un processus BPEL, entre une tâche BPMN et une activité BPEL, etc. En effet, plusieurs stratégies ont été proposées dans la littérature ayant conduit à différentes approches et algorithmes de transformation de BPMN en BPEL et *vice versa* (Ouvans *et al.*, 2006 ;

Mazanek, Hanus, 2011 ; Doux *et al.*, 2009). Ce qui est notable dans ces différents travaux est que ces transformations ne sont pas forcément évidentes et cela est du au fait que le BPMN est orienté graphe alors que le BPEL est orienté bloc et structuration de blocs.

4. Exécution des opérations de changement et analyse de l'impact du changement par un système de réécriture de graphes

La méta-modélisation de la section précédente nous permet de dresser une taxinomie des opérations de changement en considérant le critère de la granularité des constituants à modifier et la couche de l'application à laquelle ce constituant appartient. Nous avons ainsi considéré la couche BPM et la couche BPEL ou service et nous avons considéré chaque type de nœuds et d'arcs. Nous déterminons ainsi des opérations atomiques du changement telles que la création et la suppression d'une tâche d'un BPM, d'une activité d'un BPEL, etc. Nous définissons également des opérations composites telles que la fusion de deux activités, le remplacement d'une activité de base par une activité composite, etc. Une description plus détaillée de ces opérations a été effectuée dans (Bouneffa, Ahmad, 2013a). Bien entendu, nous sommes conscient qu'il serait plus judicieux de reprendre les taxinomies déjà existantes et particulièrement celles découlant de la définition de patterns de changement et du refactoring des processus métier. Cependant la limitation de la taille du papier nous contraint à réduire voir simplifier la taxinomie. L'objectif principal du papier étant plus la présentation du mécanisme de propagation de l'impact.

L'exécution des opérations de changement est formalisée par des règles de réécriture de graphes. Une règle de réécriture de graphe est en réalité une règle de production $LHS \longrightarrow RHS$ dans laquelle les deux termes *LHS (Left Hand Side)* et *RHS (Right Hand Side)* sont deux graphes. L'application d'une telle règle sur un graphe G dit graphe *Hôte*, consistera à remplacer les sous-graphes de G correspondant à *LHS* par *RHS*. En d'autres termes, cette règle consiste à trouver un morphisme m permettant d'établir une correspondance entre *LHS* et une partie g de G ($m(g)$=*LHS*) puis remplacer g par *RHS* dans G. *LHS* est appelée la pré condition de la règle et *RHS* la post condition. Il existe également des expressions qu'on appellera *NAC* ou pré conditions négatives qui stipulent que la règle ne peut être appliquée s'il existe dans le graphe hôte G, un sous graphe qui correspond à *NAC*. Ces règles sont souvent exprimées de façon visuelle. C'est notamment le cas du système *AGG* que nous utilisons. Ainsi, FIGURE 4 montre une règle correspondant à l'insertion d'une nouvelle tâche. Cette règle énonce comme pré condition à la création d'une tâche, l'existence d'un nœud de type *Process*. En d'autres termes, une tâche ne peut être créée que si elle est reliée à un processus existant. La pré condition négative énonce le fait que le processus apparaissant dans la pré condition ne doit pas contenir une tâche du même nom. Les numéros préfixant les deux nœuds de types processus, celui apparaissant dans la *LHS* et celui apparaissant dans la *NAC*, permettent de faire la liaison entre les deux processus et de stipuler qu'ils désignent le même processus.

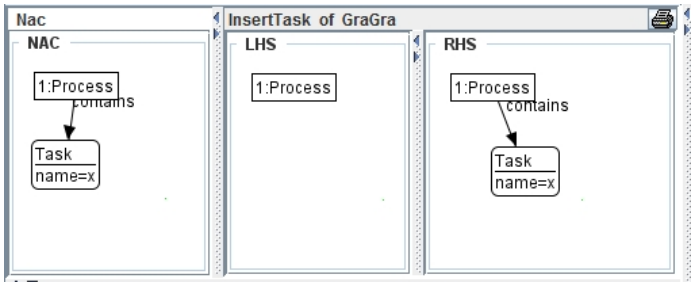


FIGURE *4. Règle de réécriture de graphe implémentant la création d'une tâche*

4.1. L'analyse de l'impact du changement

Dans le but d'analyser l'impact d'une opération de changement, nous considérons deux principales fonctionnalités : la génération et la propagation de l'impact. La génération de l'impact consiste à détecter les effets de bord d'un changement alors que la propagation de cet impact consiste à déterminer les artefacts qui sont directement ou indirectement liés aux artefacts qui viennent d'être changés et pour lesquels un impact a été généré. Pour ce faire, nous nous basons sur l'utilisation des règles de réécriture de graphes.

4.1.1. Génération de l'impact du changement

Comme nous l'avons déjà montré, chaque opération de changement est matérialisée par une règle de réécriture de graphes constituée de trois éléments : une pré condition (*LHS*), une post condition (*RHS*) et une pré condition négative (*NAC*). En général, la pré condition négative sert à éviter l'exécution d'une opération qui peut provoquer un effet de bord indésirable ou impact. En effet, une *NAC* empêche l'exécution d'une opération si cette exécution provoque la violation de différentes règles de cohérence matérialisée par les *NAC*. C'est le cas notamment dans FIGURE 4 où la *NAC* empêche la création d'une tâche portant le même nom qu'une tâche déjà existante. Dans notre approche, la génération de l'impact consiste à définir des opérations de réécriture ne contenant pas de *NAC*. En d'autres termes, on permet l'exécution de changements pouvant provoquer des incohérences. Cependant, nous ajoutons aux postconditions (*RHS*), un nœud représentant l'impact et des arcs le reliant aux nœuds qu'il affecte. En plus, ce nœud est attribué par une chaîne de caractères explicitant la cause de l'impact. FIGURE 5 représente l'opération de suppression d'une tâche reliée à deux autres tâches par un arc de type *outputs* (ces deux tâches se situent à la sortie de la tâche qui vient d'être supprimée). Dans ce cas, la pré condition négative aurait interdit cette suppression. Au lieu de cette interdiction, nous avons défini une *RHS* qui réalise la suppression et contient un nœud de type *Impact* qui est relié aux nœuds représentant les tâches affectées par cette suppression. Ce nœud contient un attribut appelé *explanation* dont la valeur est une chaîne de caractères représentant l'explication de l'ori-

gine de l'impact. Une opération de suppression peut conduire à l'écriture de plusieurs règles de générations d'impact. En d'autres termes, il peut y avoir plusieurs noeuds de type *Impact* correspondant chacun à la violation d'une règle de cohérence du BPM ou du BPEL. Cela peut être fastidieux et surtout répétitif pour le gestionnaire du changement à qui incombe l'écriture de ces règles. Nous avons donc proposé (voir section 4.2), un algorithme permettant d'automatiser la génération des règles de génération et de propagation de l'impact du changement.

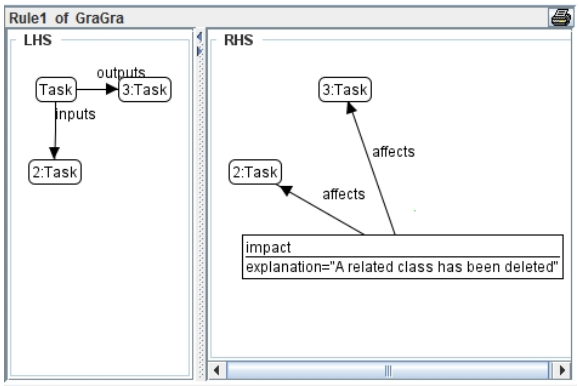


FIGURE *5. Génération d'un impact suite à la suppression d'une tâche*

4.1.2. Propagation de l'impact du changement

La propagation de l'impact d'un changement est un processus permettant d'identifier tous les nœuds indirectement affectés par l'impact de ce changement. Cette propagation est conduite par les différents liens ou relations reliant les nœuds en question. En effet, selon le type de relation et de changement, il est possible que l'impact soit propagé dans un sens ou un autre (de l'origine vers la destination de la relation ou de la destination vers l'origine) ou pas du tout. FIGURE 6 montre la propagation de l'impact affectant une tâche sur l'auteur exécutant cette tâche et cela par le biais de la relation *performs* qui relie un auteur aux tâches qu'il exécute. Nous avons défini un type d'arc appelé *inducedBy* matérialisant une relation de causalité entre impacts et permettant de visualiser la propagation ou effet de vague d'un impact.

Nous considérons deux principaux types de propagation d'impact du changement :

– La propagation horizontale consistant à propager l'impact à travers une relation reliant deux entités appartenant à la même phase du cycle de vie de l'application. En ce qui nous concerne cela revient à propager l'impact entre des constituants appartenant à la phase de modélisation du BPM ou à ceux appartenant à des artefacts appartenant à la couche service (BPEL).

– La propagation verticale consistant à propager l'impact d'un artefact appartenant à une phase du cycle de vie vers un autre artefact appartenant à une phase située en amont ou en aval. C'est le cas notamment de la propagation de l'impact entre une tâche du BPM et l'activité ou les activités BPEL qui participent à son implémentation (utilisation de la relation *ImplementedBy*) et inversement (utilisation de la relation *mappedTo*).

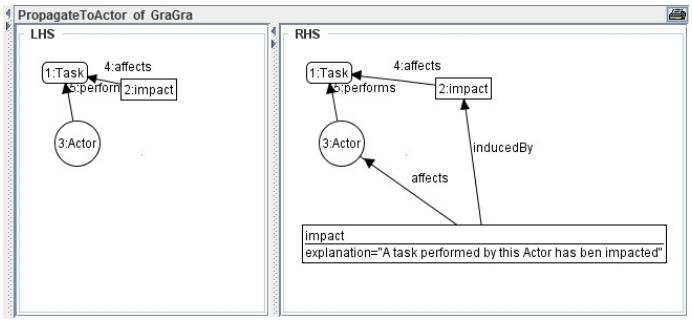


FIGURE *6. Une règle de propagation de l'impact du changement*

Nous considérons également une deuxième classification de la propagation de l'impact selon un autre critère qui est la stratégie d'exécution du processus. Ainsi, nous considérons les trois types de stratégie d'exécution suivante :

– La propagation totale qui consiste à simuler un changement, identifier l'impact de ce changement et le propager à tous les artefacts concernés à travers les différentes relations conductrices de l'impact.

– La propagation sélective qui consiste à propager l'impact à travers un nombre restreint de types de nœuds ou de relations.

– La propagation de type Change-And-Fix (Rajlich, Gosavi, 2004) qui consiste à simuler un changement et identifier l'impact aux seuls voisins directement liés au nœud qui vient d'être affecté par le changement.

La mise en œuvre de ces trois stratégies est simplifiée par l'utilisation d'un système de réécriture de graphes. En effet, ces systèmes proposent plusieurs manières d'exécuter les règles qui se distinguent par la façon de réaliser les morphismes ou mappings entre une *LHS* d'une règle et les parties du graphe *Hôte*. Dans le cas d'une propagation totale, on optera pour l'option d'exécution automatique qui laisse au système le soin de trouver lui même tous les morphismes possibles et donc toutes les règles exécutables. Dans le cas d'une exécution sélective, un regroupement des différentes règles par ensembles homogènes est préalable. Il suffira alors de déclencher le ou les ensembles de règles désirées. Dans le cas du Change-And-fix il faudra choisir l'option qui permet à l'utilisateur de choisir lui même les morphismes à chaque pas d'exécution. En effet, on peut demander au système de détecter les règles applicables à chaque pas d'exécution et choisir celle qu'on veut appliquer.

Listing 1 – Algorithme impactGeneration(artefact:a change:c)

```
1  R is a set of relationships having a as source or destination artefact
2  forAll Ri in R {
3      if Ri.forwardImpact then
4          if Ri.CertainImpact then
5              forAll d in Ri.destination {
6                  Ri.generateImpact('certainImpact', d)
7                  }
8          else
9              forAll d in Ri.destination {
10                 Ri.generateImpact('conditionalImpact', d)
11                 }
12         endif
13     endif
14
15     if Ri.inverseImpact then
16         if Ri.CertainImpact then
17             forAll s in Ri.source {
18                 Ri.generateImpact('certainImpact', s)
19                 }
20         else
21             forAll s in Ri.source {
22                 Ri.generateImpact('conditional', s)
23                 }
24         endif
25     endif
26 }
```

4.2. Automatisation de la génération des règles de propagation de l'impact

L'utilisation des règles de réécriture de graphes est un moyen de mise en œuvre flexible des processus d'analyse et propagation de l'impact du changement des applications centrées BPM. Ces règles peuvent servir comme un moyen de validation des processus de mise en oeuvre du changement. Les règles sont essentiellement basées sur une formalisation des différents constituants ou artefacts d'une application en forme de graphes et leur application est basée sur des morphismes essentiellement syntaxiques. Il est donc nécessaire de définir manuellement toutes les règles nécessaires à la propagation de l'impact en considérant chaque type de relations. Nous avons estimé qu'il serait nécessaire de simplifier ce processus en permettant son automatisation. Cela consiste à mettre en place un algorithme qui a pour entrée les artefacts à changer et le changement désiré et qui génère automatiquement des règles de propagation de l'impact à travers les différents types de relations. Il n'est malheureusement pas possible de mettre en oeuvre ce type d'algorithme sur des bases essentiellement syntaxiques. Cela montre donc la nécessité d'une explicitation de la sémantique par des langages et outils issus du web sémantique comme OWL. En utilisant l'explicitation d'une partie de la sémantique des relations par rapport à la conduction de l'impact, nous avons défini un algorithme permettant la génération de règles de propagation de l'impact (Listing 1). L'algorithme fait appel à la fonction *generateImpact(ImpactType:String, AffectedNode: Node)* dans laquelle *AffectedNode* est la source ou la destination d'une relation *Ri*. Cette fonction génère une règle dans laquelle la *LHS* est un sous graphe contenant *AffectedNode* avec l'arc correspondant à

Ri et toutes les extrémités de cet arc. La *RHS* contient *AffectedNode* non directement liée par l'arc correspondant à *Ri* et un nœud de type *ImpactType* est créé et est lié à *AffectedNode*.

5. Prototype d'implémentation

Le prototype que nous avons développé pour la l'implémentation de notre approche est constitué de deux parties. La première partie concerne l'utilisation du système de réécriture de graphe (*AGG*) et de l'outil *Protégé* pour l'élaboration de l'ontologie. Cela nous a permis de mettre en œuvre un prototype flexible permettant une formalisation exécutable des concepts que nous introduisons. Cet outil sert essentiellement dans un environnement de recherche et les utilisateurs sont des personnes capables de maîtriser les concepts issus de la réécriture de graphes et du web sémantique (notamment le langage OWL). Partant de la spécification opérationnelle de notre approche à l'aide des outils *AGG* et *Protégé*, nous avons intégré les processus d'analyse d'impact dans un atelier intégré appelé *Architect* (Ahmad *et al.*, 2013) que nous avions déjà développé et utilisé dans le cadre de l'analyse d'impact du changement des applications distribuées. Cet atelier se présente comme un ensemble de plugins Eclipse permettant des fonctionnalités incluant le parsing de code sources multi-langages (Java, C++, fichiers de configuration spring et J2EE, schémas de bases de données, etc.); le stockage des différents artefacts dans une base de données en forme de graphes au

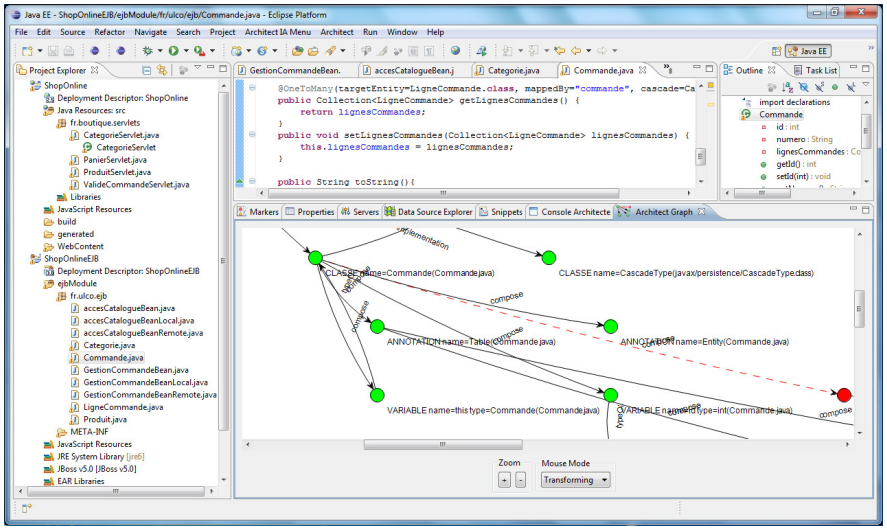


FIGURE 7. *Scénario de la propagation de l'impact dans l'atelier* Architect

format *GXL* [10]; la définition de règles d'analyse et de propagation de l'impact du changement en utilisant le moteur de règle *Drools* [11] et la visualisation et la manipulation à base de graphes des différentes artefacts ainsi que la visualisation de l'impact du changement en utilisant la librairie *Java Universal Network/Graph (JUNG)* [12].

Dans *Architect* (FIGURE 7), les règles de réécriture de graphes ont été transcrites en règles *Drools*. Nous avons donc transcrit chaque règle en explicitant sa pré condition et sa post condition en termes constitués d'objets Java représentant les noeuds et les arcs des sous graphes associés aux pré et post conditions.

6. Conclusion

Nous avons présenté une approche basée sur une méta-modélisation à base de graphes typés et attribués des constituants d'une application centrée BPM et sur les systèmes de règles de réécriture de graphe pour une mise en œuvre flexible de l'analyse et la propagation de l'impact du changement. Nous avons considéré la propagation horizontale de l'impact entre constituants appartenant à la même phase du cycle de vie d'une application centrée BPM et la propagation verticale entre des constituants appartenant à des phases différentes de ce cycle de vie. L'utilisation d'un langage d'ontologies nous a également permis d'expliciter la sémantique des relations inter-constituants en matière de conduction de l'impact du changement. Cela nous a permis de mettre en œuvre un algorithme générique permettant la génération de règles de propagation de l'impact. Les idées émergeant de notre approche ont été validées à l'aide d'un système de réécriture de graphes mais nous avons également mis en œuvre ces mêmes idées dans le cadre d'un atelier intégré que nous développons depuis de nombreuses années et qui est dédié à l'analyse et la propagation de l'impact dans le cadre des applications distribuées incluant, notamment, les applications web multi-tiers.

Nous continuons actuellement notre travail d'explicitation de la sémantique en tentant d'établir des liens de traçabilité entre les besoins des utilisateurs (experts métier) et les BPMs. Le but étant de permettre à un expert d'exprimer les changements au niveau des besoins exprimés en utilisant le vocabulaire de son domaine et d'analyser les impacts sur l'application. Nous avons commencé la mise en oeuvre de cette approche dans le cadre des applications d'optimisation de la chaîne logistique (Hendi *et al.*, 2016).

Bibliographie

Ahmad A., Basson H., Deruelle L., Bouneffa M. (2009, May). A knowledge-based framework for software evolution control. In *Actes du xxviième congrès inforsid*, p. 111-126. Toulouse, France, IRIT Press (www.irit.fr).

10. http://www.gupro.de/GXL/
11. http://www.jboss.org/drools/
12. http://jung.sourceforge.net/

Ahmad A., Bounefa M., Basson H. (2013). *Multi-modélisation de l'évolution du logiciel distribué et hétérogène (french edition)*. Allemagne, Éditions Universitaires Européenes.

Bouneffa M., Ahmad A. (2013a). The change impact analysis in bpm based software applications: A graph rewriting and ontology based approach. In *Enterprise information systems*, p. 280–295. Springer.

Bouneffa M., Ahmad A. (2013b, July). Change management of bpm-based software applications. In *15th international conference on enterprise information systems (iceis 2013)*, p. 37-45. Angers, France, Springer.

Doux G., Jouault F., Bézivin J. (2009). Transforming bpmn process models to bpel process definitions with atl. In *5th international workshop on graph-based tools.*

Folli A., Mens T. (2007). Refactoring of UML models using AGG. *ECEASST*, vol. 8.

Gottschalk K., Graham S., Kreger H., Snell J. (2002, April). Introduction to web services architecture. *IBM Syst. J.*, vol. 41, p. 170–177.

Hendi H. I., Bouneffa M., Ahmad A., Fonlupt C. (2016). Ontology based reasoning for solving passenger train optimization problem. In *Proceedings of aic-mitc 2016. to appear.* IEEE Computational Intelligence Society.

Juric M. B. (2006). *Business process execution language for web services bpel and bpel4ws 2nd edition.* Packt Publishing.

Kherbouche O. M., Ahmad A., Basson H. (2013). Using model checking to control the structural errors in BPMN models. In *IEEE 7th international conference on research challenges in information science, RCIS 2013, paris, france, may 29-31, 2013*, p. 1–12.

Kim D., Kim M., Kim H. (2007). Dynamic business process management based on process change patterns. In *Convergence information technology, 2007. international conference on*, p. 1154–1161.

Maweed Y., Bouneffa M., Basson H., Sack P. O. (2005). Vers une implémentation flexible des activités de maintenance et d'évolution du logiciel. In *Actes du xxiiième congrès inforsid, grenoble, france, 24-27 mai, 2005*, p. 201–216.

Mazanek S., Hanus M. (2011). Constructing a bidirectional transformation between bpmn and bpel with a functional logic programming language. *Journal of Visual Languages & Computing*, vol. 22, n° 1, p. 66 - 89. (Special Issue on Visual Languages and Logic)

Mens T. (2005). On the use of graph transformations for model refactoring. In *Generative and transformational techniques in software engineering, international summer school, GTTSE 2005, braga, portugal, july 4-8, 2005. revised papers*, p. 219–257.

Ouvans C., Dumas M., Hofstede A. ter, Aalst W. van der. (2006, Sept). From bpmn process models to bpel web services. In *Web services, 2006. icws '06. international conference on*, p. 285-292.

Rajlich V., Gosavi P. (2004). Incremental Change in Object-Oriented Programming. *IEEE Softw.*, vol. 21, n° 4, p. 62–69.

Reijers H., Vanderfeesten I., Aalst W. van der. (2016). The effectiveness of workflow management systems: A longitudinal study. *International Journal of Information Management*, vol. 36, n° 1, p. 126 - 141.

Session Exigences, Justification et Raisonnement

Vers une modélisation et une analyse des exigences spatio-temporelles

Mounir Touzani [1], Christophe Ponsard [2], Anne Laurent [1,3], Thérèse Libourel [1,3,4], Joël Quinqueton [1,5]

1. *LIRMM - Université de Montpellier, CNRS, Montpellier, France*
   touzani@lirmm.fr
2. *CETIC - Centre de recherche, Gosselies, Belgique*
   christophe.ponsard@cetic.be
3. *Université de Montpellier (UM), Montpellier, France*
   laurent@lirmm.fr
4. *Espace-Dev (UM, UAG, UR, IRD), Université de Montpellier, Montpellier, France*
   therese.libourel@umontpellier.fr
5. *Université Paul-Valéry (UPVM), Montpellier, France*
   jq@lirmm.fr

RÉSUMÉ. *L'Ingénierie des Exigences (IE) est une étape clef dans tout projet d'évolution d'un système d'information (SI). Les développements actuels, sur les systèmes mobiles par exemple, impliquent de facto une dimension spatio-temporelle, souvent réservée aux SI géographiques (SIG). Ceci nécessite des méthodes plus systématiques pour capturer et raisonner sur des exigences de nature spatio-temporelle. Cet article propose un cadre de référence permettant de systématiser l'identification, la structuration et le raisonnement sur ce type d'exigences. Ce cadre proposé intègre des contributions dans les domaines de l'IE et de la géomatique. Nous l'avons outillé et nous l'illustrons à travers une étude de cas de fusion de deux universités.*

ABSTRACT. *Requirements Engineering (RE) is a key step in any project aiming at evolving an information system (IS). Current developments, on mobile systems for example, involve a spatial and temporal dimension, often reserved to geographic IS (GIS). This requires more systematic methods for capturing and reasoning about the spatial and temporal nature of requirements. This paper proposes a framework for systematically identifying, structuring and reasoning about such requirements. This proposed framework includes contributions in the fields of RE and geomatics. We illustrate it through a case study: the merger of two universities.*

MOTS-CLÉS : *Ingénierie des Exigences, Exigences spatio-temporelles, Raffinement par les buts*

KEYWORDS: *Requirements Engineering, spatio-temporal requirements, goal-driven refinement*

Congrès INFORSID, 34ème édition, Grenoble, 31 mai - 3 juin 2016

1. Introduction

L'Ingénierie des Exigences (IE) est le processus qui a pour objet d'établir et de maintenir un accord avec les parties prenantes sur les exigences du système à construire (ISO29148, 2011). Il s'agit de dégager des responsabilités qui seront confiées à différents agents : des êtres humains, des dispositifs matériels ou des systèmes d'information. La collaboration de ces agents permet de réaliser des objectifs du système dans son ensemble. De tels objectifs peuvent être identifiés, structurés, analysés et documentés à l'aide de méthodes d'IE orientées buts (Lamsweerde, 2009).

L'IE est une étape cruciale pour le succès d'un projet : de nombreuses études montrent qu'omettre celle-ci est une cause majeure d'échecs de projets (Hughes *et al.*, 2015). Des méthodes ont été développées afin d'assurer des propriétés clés de complétude, précision, non-ambiguïté et testabilité. Ces méthodes peuvent être de nature très générique (classification d'exigences, listes de contrôles, techniques de raffinements de buts) ou très spécifiques à des domaines (exemple de critères de sécurité). Les premières sont utiles mais n'apportent que des garanties limitées, tandis que les secondes permettent une grande précision mais ne s'appliquent que sur des domaines pointus. Il est donc intéressant de considérer des classes de propriétés intermédiaires partagées par de nombreux systèmes et permettant un bon compromis entre leur applicabilité et leur apport. Plus précisément, on s'intéresse ici aux propriétés liées aux caractéristiques physiques spatiales et temporelles du système analysé.

Les propriétés temporelles ont été largement étudiées dans le cadre de l'analyse de systèmes à base de logiciels, notamment au niveau de systèmes réactifs et temps réels pour lesquels, des logiques spécifiques ont été proposées(Manna, Pnueli, 1992). A l'inverse, les propriétés spatiales ont été abordées de manière assez restreinte en IE (Touzani *et al.*, 2015) et ont surtout été étudiées au niveau de l'analyse de système d'information géographique (SIG), notamment sur la base de formalismes objets (Kosters *et al.*, 1996) et conceptuels/graphiques (Bédard, Larrivée, 2008 ; Pinet, 2012).

Ceci devient limitatif pour la spécification d'un nombre croissant de systèmes reposant de plus en plus intensivement sur un logiciel toujours plus fortement ancré et connecté avec le monde réel. On peut citer les systèmes mobiles, cyber-physiques, l'internet des Objets et des champs d'applications comme les villes intelligentes, les usines du futur ou la logistique. Ces systèmes nécessitent de disposer d'une perception précise du monde réel et donc des exigences spatiales et temporelles sur celui-ci.

Notre proposition est centrée autour de la prise en compte de la dimension spatio-temporelle (ST). Dans le cadre de cet article, nous restons focalisé sur des systèmes d'information géographique, en particulier au niveau de l'étude de cas présentée. Notre priorité est la définition de notations simples mais avec un bon pouvoir d'expression mais sans lui associer de sémantique formelle à ce stade. Le point d'ancrage est une approche d'IE orientée buts, répondant déjà aux questions du "POURQUOI/COMMENT" (les buts et leur raffinement en exigences), du "QUOI" (les opérations) et du "QUI" (les agents). Nous abordons ici plus précisément et de manière

liée, les questions du "QUAND" et du "OÙ". Nous utilisons pour ceci deux axes de recherche : d'une part, nous explorons la dualité dimensions spatiales/temporelles, afin de transposer à la dimension spatiale des techniques d'IE déjà définies. D'autre part, nous prenons en considération des notations largement utilisées dans les SIG, et ce, afin de les intégrer dans les primitives d'IE et faciliter ainsi la capture d'exigences spatio-temporelles. Nous avons utilisé le référentiel KAOS d'IE orienté buts (Lamsweerde, 2009) et avons réalisé un prototype à l'aide de l'outil Objectiver (Respect-IT, 2005). Cependant, nos résultats peuvent être appliqués à d'autres référentiels et des domaines plus large que la pure information géographique.

Ce travail a pour fil conducteur une étude de cas concrète et riche en termes d'exigences spatiales et temporelles : il s'agit de la fusion de deux établissements universitaires bien connus des auteurs. Les effets de cette fusion créent une dynamique dans l'espace et une évolution dans le temps, mettant en évidence le mouvement des personnes en tant qu'entité de l'espace ainsi que tout l'aspect organisationnel entraînant des changements ST majeurs au niveau des différentes composantes et directions déjà existantes des deux universités.

Cet article est organisé comme suit : dans la section 2 nous dressons un état de l'art autour de la dimension spatio-temporelle et des spécificités de l'information géographique (IG), des formalismes de représentation des objets ST et quelques concepts de l'ingénierie des exigences, et plus spécifiquement de la méthode orientée buts KAOS servant de cadre de base pour ce travail. La section 3 présente l'étude de cas de fusion de deux universités, qui sera utilisée pour illustrer notre propos tout au long de l'article. La section 4 présente le cadre de référence unifié en termes de notations ainsi qu'une proposition d'extensions aux différentes dimensions de la modélisation des exigences, afin de faciliter la capture des dimensions ST. Ensuite, la section 5 propose des extensions méthodologiques permettant de raisonner qualitativement sur celles-ci. Enfin, la section 6 conclut par une analyse critique des apports de notre contribution, de ses limitations et propose quelques perspectives.

2. État de l'art

Dans cette section, nous passons en revue les approches spatio-temporelles en pointant les principaux types de relations spatiales et temporelles ainsi que les formalismes qui ont été développés pour les représenter. Nous décrivons également les mécanismes d'IE pour identifier, structurer et raisonner sur des exigences en termes d'espace et de temps.

2.1. L'approche spatio-temporelle

2.1.1. Spécificités de l'information géographique

Selon (Becker, al, 23-27 juillet 1990), l'information géographique (IG) décrit un objet, un phénomène ou encore une action du monde réel. Elle fournit, pour chaque objet concerné, des informations sur le nom, le type, les caractéristiques thématiques,

la forme, la localisation géographique ou même des informations relatives à des objets en relation de proximité.

L'IG est devenue la matière première qui a permis le développement des Systèmes d'Information Géographique (SIG) (Laurini, Thompson, 1992). Les SIG sont cependant restés confinés à ce domaine et souvent réservés à des utilisateurs avertis.

Les logiciels SIG sont, quant à eux, spécifiques à des travaux d'analyse spatiale et de cartographie tandis que le cadre plus large des SI envisagés demande de pouvoir prendre en compte certaines informations qualitatives relatives à des données géographiques et induisant des descriptions particulières. Par exemple : *cette personne travaille à côté du point de rassemblement de la cafétéria.*

2.1.2. Relations spatio-temporelles

L'analyse spatio-temporelle introduit concomitamment les notions d'espace et de temps. L'espace fait référence aux informations géographiques de localisation et permet de définir des relations spatiales entre les objets. Ces relations sont aussi importantes que les entités elles-mêmes (Papadias, Kavouras, 1994 ; Clementini, 2009 ; Egenhofer, Franzosa, 1991). Beaucoup de directions ont été prises pour les définir en trois classes : topologiques (Randell *et al.*, 1992) (p.ex. adjacence : la pharmacie est collée au laboratoire d'analyses), métriques (distance) (Pullar, Egenhofer, 1988) (p.ex. la ville est située à 5 km de la plage) ou par projection (orientation) (Zimmermann, Freksa, 1996 ; Ligozat, 1998) (p.ex. Paris est au nord de Toulouse). La figure 1 représente les relations spatiales qui peuvent exister entre deux objets.

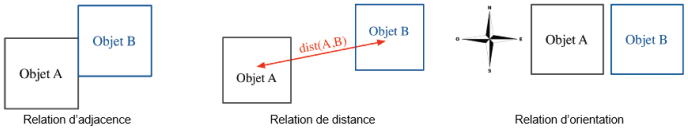


Figure 1. Relations spatiales entre deux objets

L'introduction du temps vise la dimension temporelle (Allen, Yabushita, 1984), (Frank, 1994) qui peut correspondre à des événements se produisant soit à un instant ou à une période, soit à des changements sur plusieurs instants voire périodiques.

La perception des relations entre objets dans l'espace d'une part, et objets dans le temps d'autre part, montre une forte analogie. (Le Parc-Lacayrelle, Al, 2007) a décrit dans son article trois types de relations : l'adjacence (p.ex. autour du 15 novembre 2015), l'inclusion (p.ex. au milieu de l'année) et la distance (p.ex. deux semaines avant la fin de l'année). La figure 2 représente les relations temporelles qui peuvent exister entre deux objets.

Les notions ST peuvent bien sûr être toutes considérées simultanément. La notion du "mouvement" reste à notre sens la plus évidente, représentant une succession de localisations spatiales qui évoluent dans le temps. Un exemple d'événement ST est "Le camion entre dans la zone d'approche d'un dépôt". Des configurations spatiales peuvent aussi évoluer au cours du temps. Un exemple parlant est la récente fusion des

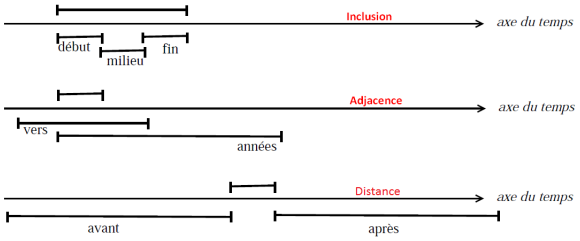


Figure 2. Relations temporelles entre deux objets

régions en France. Des typologies complètes de concepts ST pouvant être combinées sont détaillées notamment dans (Claramunt, Jiang, 2001 ; Mathian, Sanders, 2014).

2.1.3. Formalismes de représentation des objets spatio-temporels

En termes de modélisation, l'objectif des chercheurs a toujours été de rendre l'IG moins complexe (Zoghlami, 2013), d'où le besoin de formaliser les propriétés spatiales et temporelles, et ce, afin de rendre facile la description de la géométrie (dimension, position, taille, forme et orientation) ainsi que la temporalité pour une meilleure communication.

Par exemple, une route peut être considérée comme un objet linéaire au sein d'un graphe si le but est de gérer le trafic ou comme un volume si le but est de gérer la répartition de réseaux d'eau et de gaz. Deux modes de représentation interne existent selon la perception de l'IG : le mode raster (ensemble de pixels, généralement utilisés pour représenter une variable continue telle que la température) et le mode vecteur (ensemble de points, lignes et polygones, généralement utilisés pour décrire des objets avec une géométrie).

	0D	1D	2D	3D
Spatial (abstraction 2D)				N/A
Spatial (abstraction 3D)				
Temporel			N/A	N/A

Figure 3. Exemples de pictogrammes spatiaux et temporels

La représentation des objets spatiaux nécessite une modélisation adaptée aux phénomènes spatio-temporels. En ce sens, des formalismes de représentation spatio--temporelle, via l'utilisation de pictogrammes spatiaux et temporels dans les modèles, ont été proposés et affinés par Bédard depuis Modul-R puis PVL (Plugin for Visual Language) (Bédard, Larrivée, 2008) et son évolution plus récente PictograF[1]. De nombreux formalismes de type entité-relation ou orienté objet, ont été aussi proposés par d'autres auteurs afin de faciliter la modélisation de l'information spatiale et les aspects temporels associés. Une étude exhaustive est disponible dans le mémoire de (Pinet, 2012). La figure 3 présente des pictogrammes spatiaux et temporels.

1. http://pictograf.scg.ulaval.ca

2.2. L'ingénierie des exigences

Une **exigence** peut être définie comme étant une condition ou capacité dont l'utilisateur a besoin pour résoudre un problème ou parvenir à un objectif. Elle doit être satisfaite par un système ou un composant d'un système pour satisfaire un contrat, une norme, une spécification, ou autres documents formellement imposés (IEEE1990, 1990).

Pour structurer les besoins relatifs aux systèmes à développer, un processus d'IE peut être décomposé en quatre étapes de développement: élucidation, analyse, spécification et validation. Ces étapes sont coordonnées par un processus de gestion des exigences. Nous nous limiterons ici aux étapes d'élucidation et d'analyse.

Parmi les méthodes d'IE existantes, les méthodes orientées buts se démarquent par les garanties de complétude et de précision qu'elles peuvent apporter (Rolland, Salinesi, 2005). Un **but** est un objectif que le système considéré devrait atteindre. Les formulations de buts se réfèrent à des propriétés destinées à être assurées (Lamsweerde, 2009). Les buts peuvent être exprimés à différents niveaux d'abstraction, depuis des buts stratégiques de haut niveau, comme "réaliser la fusion des universités avec succès" jusqu'à des buts opérationnels tels que "planifier l'affectation des amphithéâtres pour les cours du semestre" (voir figure 5). Les buts de haut niveau peuvent être progressivement raffinés en buts plus concrets et finalement opérationnels au moyen de relations liant un but parent à plusieurs buts fils, avec des conditions de satisfaction différentes soit "ET" (tous les fils nécessaires) soit "OU" (un des fils suffisant : c.-à-d. des alternatives possibles). A partir d'un but donné, la question du « POURQUOI » permet d'identifier le but père, tandis que la question du « COMMENT » permet d'identifier un ensemble de buts fils permettant d'atteindre ce but. La décomposition s'arrête quand on atteint un but contrôlable par un **agent**, c.-à-d. qui répond à la question « QUI » déterminant la responsabilité. Ils correspondent soit à des **exigences** sur le logiciel soit à des **attentes** sur le comportement d'agents de l'environnement. Un exemple d'exigence concret est le contrôle d'accès à un bâtiment de l'université: "Le système autorise l'ouverture d'un point d'accès à tout utilisateur disposant du droit d'accès". Des propriétés du domaine peuvent également entrer en ligne de compte pour justifier un raffinement. De telles propriétés sont intrinsèquement valables. Un exemple est la propriété spatio-temporelle suivante : "un objet physique ne peut se trouver qu'à un endroit à un instant donné".

La figure 4 représente l'articulation de ces concepts au sein du méta-modèle KAOS, [2]. Il est composé des quatre sous-modèles suivants: Le **modèle des Objets** décrit le domaine (entités, relations, événements) utilisé pour exprimer les buts. Leur représentation se base sur les diagrammes de classe UML.

Le **modèle des Buts** structure les buts fonctionnels et non-fonctionnels que le système doit atteindre par la coopération d'agents. Il permet aussi d'identifier et raisonner

2. KAOS : Keep All Objectives Satisfied

sur les conflits entre les buts ainsi que des obstacles susceptibles de bloquer leur réalisation. Il est représenté graphiquement par un arbre de buts.

Le **modèle des Agents** identifie les agents du système, les informations échangées (interfaces) et les exigences sous leurs responsabilités. Ces responsabilités sont reprises dans les arbres de buts. Des flux entre agents peuvent aussi être représentés.

Le **modèle des Opérations** décrit comment les agents coopèrent fonctionnellement afin d'assurer la réalisation des exigences qui leur sont confiées ainsi que les buts du système. Nous utilisons ici le diagramme de flux fonctionnel.

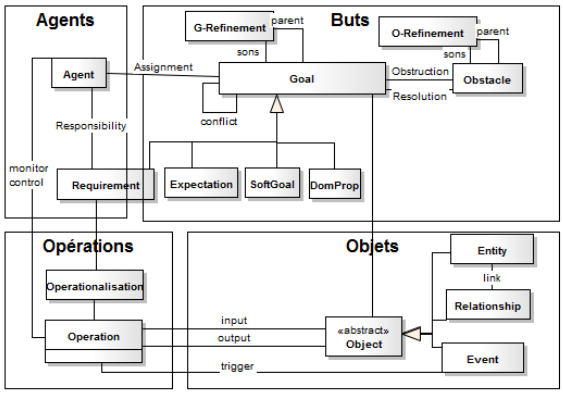


Figure 4. Le méta-modèle KAOS

Seule la dimension temporelle est explicitement traitée par les méthodes d'IE orientées buts, KAOS en particulier. Elle est présente sous les aspects suivants :

– les propriétés temporelles d'un but peuvent être exprimées en utilisant des mots-clefs spécifiques comme "toujours", "finalement", "avant une échéance", "tant que", etc. Une quantification temporelle peut être également utilisée pour représenter la durée pendant laquelle une propriété temporelle ou son échéance doit être respectée. Ces *patterns* ont été répertoriés par (Dwyer *et al.*, 1999). Sur cette base, (Mahaux, 2004) a proposé un système de spécification allégée, basée sur un ensemble structuré de patrons de formulation. Cela encadre et guide la spécification de contraintes temporelles de façon cohérente, non-ambiguë, expressive et qui reste aisée à comprendre.

– la logique temporelle peut être utilisée pour donner une sémantique mathématique précise, permettant de mettre en œuvre des outils de vérification automatique ou de preuve de modèles. Dans ce travail, nous ne prenons pas en considération ce niveau formel très précis mais peu aisé à comprendre et qui peut être "occulté" par les patrons décrits au point précédent. A titre d'exemple, l'exigence de contrôle d'accès peut s'exprimer ainsi : $\forall u : User, c : AccessControl \cdot \Box(authorised(u, c) \rightarrow \bullet open(c))$. Les différents prédicats sont des entités, relations ou attribut du modèles objets. Ils peuvent être combinés à des opérateurs temporels : $\Box$ signifie "tout le temps" et $\bullet$ indique que l'ouverture se produit à l'état suivant.

– des patrons de raffinements basés sur le temps peuvent être utilisés pour structurer les buts. Le patron le plus connu est le "jalon temporel" qui décompose une propriété devant être "finalement" atteinte en plusieurs étapes intermédiaires. Ces patrons sont prouvés une fois pour toutes et permettent typiquement de découvrir des sous-buts manquants. Une bibliothèque très élaborée de raffinements a été proposée par (Darimont, Lamsweerde, 1996).

3. Description de l'étude de cas : fusion de deux universités

Notre étude de cas est largement inspirée de la fusion récente des Universités de Montpellier 1 et 2. Il s'agit d'un processus initié en 2012 (Cholley-Gomez, 2015) et qui a abouti à la naissance de l'université de Montpellier en janvier 2015 [3]. Ce processus est encore en cours en 2016.

Pour une raison d'abstraction et de simplification, nous considérons deux établissements universitaires indépendants et situés dans une même région mais pas à proximité immédiate. Les objectifs stratégiques auquel la fusion répond est de renforcer l'offre de formation, d'augmenter le potentiel de recherche par de multiples synergies entre les laboratoires et d'accroître le rayonnement international, sur cette base des buts plus précis de regroupement de services, unification de l'infrastructure et d'intégration des programmes de cours sont définis et illustrés à la figure 5. Certains de ces buts seront raffinés par la suite.

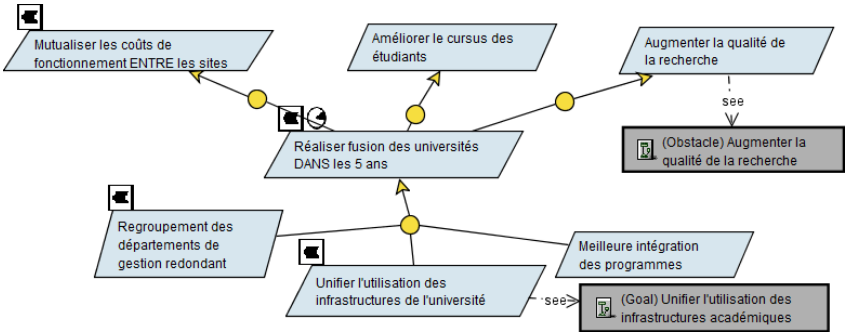


Figure 5. Buts stratégiques de la fusion illustrés dans l'outil Objectiver étendu

La distance qui sépare les établissements, de l'ordre d'une dizaine de kilomètres, est suffisamment faible pour permettre une mobilité mais suffisamment élevée pour nécessiter de repenser les affectations géographiques et la mobilité des personnes dans la cadre de la fusion. Les grandes questions qui émergent sont :

– La nécessité du rapprochement de certains services (administratifs, financiers et de gestion de ressources humaines,...) ayant des missions similaires et qui devront mettre en commun leurs activités, voire réunir leurs compétences en un seul lieu.

3. A noter que d'un point de vue spatio-temporel, l'université de Montpellier a existé de manière unifiée entre 1289 et 1793, puis entre 1896 et 1970. Il s'agit donc d'une renaissance.

– L'amélioration de l'organisation et la planification de la circulation des étudiants sur le nouveau campus commun : deux cours qui se suivent ne peuvent pas se dérouler dans deux salles distantes de plus de 500 mètres par exemple, ou encore un local de recherche en biologie doit être situé à proximité de salles avec des paillasses pour les travaux pratiques, etc.

4. Enrichissement des modèles d'IE avec des notations spatiales et temporelles

Dans cette section, nous présentons les extensions ST aux modèles des objets et des buts de la méthode KAOS. Ces extensions ont été implémentées et expérimentées à l'aide de l'outil "Objectiver" offre des possibilités d'extension à la fois de son méta-modèle et des visualisations par l'utilisation de connecteurs (ou plugins). Le plugin développé est disponible en ligne [4].

4.1. Extension du modèle objet

Pour introduire les notions spatiales et temporelles dans le modèle objet, nous nous sommes basés sur le système de notations "PictograF". Ces notations développées depuis une vingtaine d'années ont atteint un bon niveau de maturité et de standardisation, notamment au niveau de leur intégration au moyen de stéréotypes UML. Leur valeur ajoutée au niveau de l'IE est de permettre de capturer facilement et systématiquement des propriétés du domaine qui seraient spécifiées de manière textuelle et généralement ponctuelle à des raffinements au sein du modèle. Ces propriétés seront utiles pour mener la démarche de raffinement de buts et vérifier leur cohérence et complétude.

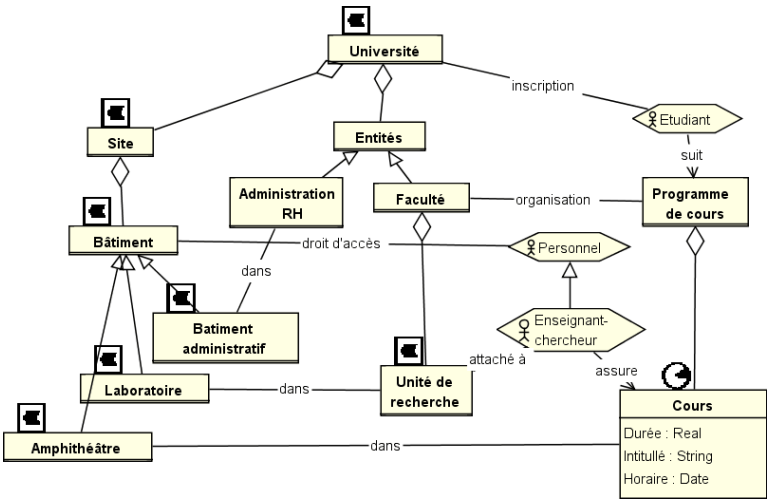


Figure 6. Modèle objet du domaine

4. Le plugin est téléchargeable depuis : http://www.objectiver.com/packages/plugins/STPlugin.jar

La figure 6 illustre l'application de ces notations à notre exemple. La partie gauche du diagramme représente une structure de décomposition d'entités de nature spatiale (site, bâtiment, amphithéâtre, etc.). A noter que la relation d'agrégation prend également un sens spatial quand elle est associée à des entités de cette nature. La partie centrale représente des structures de l'organisation (facultés, unités de recherches, etc.) situé dans ces espaces. Enfin la partie de droite présente les agents humains "étudiant" et "enseignant-chercheur" ainsi que les relations qu'ils entretiennent via les activités de cours. Ces dernières ont une dimension temporelle (durée) mais ont un lien direct avec la dimension spatiale via l'amphithéâtre dans lequel elles sont assurées.

4.2. Extension du modèle des buts

Les notations "PictograF" peuvent se généraliser aux buts : un but pourra être marqué comme ayant une dimension spatiale et/ou temporelle spécifique si :

– des éléments spatiaux et temporels qu'il référence ont ces dimensions. Il s'agit d'une règle de cohérence entre les annotations du modèle objet et celui des buts. Ainsi, un but référençant un bâtiment aura la dimension spatiale correspondante.

– des exigences propres à ce but sont de nature spatiale ou temporelle. On peut rappeler ici l'exigence "Deux cours qui se suivent ne peuvent pas se dérouler dans deux salles distantes de plus de 500 mètres" (bien que la notion de salle apporte déjà la dimension spatiale dans ce cas). Au niveau temporel, on pourrait l'exprimer par : "un enseignant-chercheur ne passera pas plus de 3h en déplacement entre différents sites".

Au niveau spatial, nous nous sommes limité à un sous-ensemble de notations PictograF. Par simplicité, l'idée est de rester le plus souvent possible sur une abstraction 2D. Par exemple, à ce stade, nous n'avons pas considéré le cas des éléments linéaires dans un monde 3D. Au niveau temporel, par contre, nous avons ressenti le besoin d'étendre les notations PictograF sur base de la mise en correspondance avec les patrons de formulation déjà définis :

– Les buts de progrès "**Achieve**" expriment généralement l'occurrence d'un événement spécifique qui peut être exprimé par un pictogramme d'événement ponctuel mais de nombreux systèmes peuvent également exprimer une répétition pour laquelle nous avons introduit un pictogramme de fréquence ⊛. On pourra ainsi exprimer une exigence telle que "Les dispositifs de contrôles d'accès doivent faire l'objet d'une maintenance TOUS les ans".

– Les buts d'assurance "**Maintain**" quant à eux expriment une propriété qui doit être vraie tout le temps ou de manière plus précise sur une durée bien déterminée (p.ex. "Aucun accès aux étudiant n'est autorisé entre 22h et 7h du matin"). Le second cas est couvert par le pictogramme de durée. Pour le premier cas qui est important pour exprimer des propriétés de sûreté de fonctionnement, nous avons introduit un cercle "plein" qui est son extension naturelle ●. A noter que ce symbole ne doit pas être confondu avec le celui utilisé au sein des formules de logique temporelles qui signifie lui "l'instant d'après". Le contexte d'utilisation est cependant différent.

Des patrons de formulation peuvent être introduits pour aider à formuler des exigences spatiales. Certains principes de dualité peuvent être appliqués pour les mettre en évidence. Le tableau 1 illustre les principes que nous avons identifiés.

Tableau 1. Dualité entre les primitives temporelles et spatiales

	Domaine temporel	Domaine spatial
Dimension	1	0, 1, 2 ou 3
Quantificateur existentiel	Une fois (dans le passé, le futur)	A un endroit (dans une direction/dimension donnée)
Quantificateur universel	Tout le temps (dans le passé, le futur)	Partout (dans une direction/dimension donnée)
Mesure absolue	Temps "universel"	Coordonnées "GPS"
Mesure relative	Temps par rapport à un événement de référence	Distance, surface, volume d'un objet référencé
Fréquence	Périodicité temporelle (ex. toutes les secondes)	Notion de régularité spatiale (ex. tous les X mètres, à tous les étages, etc.)
Raisonnement qualitatif	Positions relatives d'événements, d'intervalles	positions relatives d'objets (topologie)
Raisonnement quantitatif	Comptage d'événements métiers dans un intervalle donné	Capacité d'un espace en termes de métier

5. Raisonnement sur les exigences spatiales et temporelles

L'enrichissement des modèles d'IE avec les notations ST décrites dans la section précédente, permet d'en améliorer la **précision**. Il permet aussi d'envisager de nouvelles formes de raisonnement apportant des bénéfices supplémentaires en termes de **complétude** pour guider dans la découverte systématique d'exigences via des techniques de raffinement de buts spécifiques. Par ailleurs, ceci permet aussi d'aider à spécifier un système qui sera **robuste** face à des risques liés aux propriétés ST (exemple : une échéance peut causer un risque de retard, une contrainte de capacité peut causer un risque d'indisponibilité). Enfin on considérera aussi dans cette section des stratégies de **délégation de responsabilité d'agents** prenant en compte des contraintes ST.

5.1. Raffinement de buts guidé par les exigences spatio-temporelles

On peut considérer un raffinement dirigé par la dimension spatiale et/ou temporelle, avec des interactions possibles entre ces deux dimensions. Les principales stratégies suivantes ont pu être identifiées pour tirer partie de nos extensions ST :

– **L'application de patrons de raffinement temporels et spatiaux**. Des schémas de décomposition de buts ont déjà été proposés (Darimont, Lamsweerde, 1996 ; Lamsweerde, 2009). Le plus connu est le patron "jalon" ("milestone" en anglais), qui décompose un but à accomplir en étapes logiques intermédiaires. Ces étapes peuvent être temporelles, par exemple : décomposer une tâche d'une durée d'une heure en trois étapes successives de 20 minutes. Au niveau spatial, ces "jalons" peuvent se traduire par une progression spatiale unidimensionnelle ou en 2 voire 3 dimensions. On peut par exemple, définir des exigences sur un transport de marchandises, sur un processus d'évacuation progressive d'un bâtiment ou encore définir une stratégie de rénovation d'un site pour assurer une certaine continuité des services. La figure 7 illustre un jalon

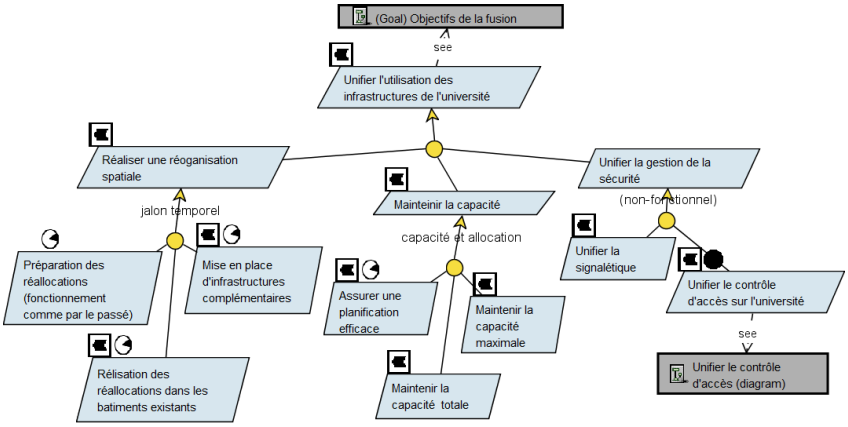


Figure 7. Raffinement du but d'unification de l'utilisation des infrastructures

temporel au niveau du raffinement du but "réaliser une réorganisation spatiale". Celui-ci est composé de trois étapes successives dans le temps : une première étape "immédiate" de planification, où le fonctionnement est comme par le passé, une deuxième "à moyen terme", où une réorganisation se fait dans les infrastructures existantes utilisables en l'état, et une troisième étape "à plus long terme" qui demande de rénover, voire construire de nouvelles infrastructures.

– **La propagation d'une exigence spatiale ou temporelle sur une structure du domaine**. La figure 8 illustre cette propagation plus concrètement sur le sous-but d'unification de la sécurité.

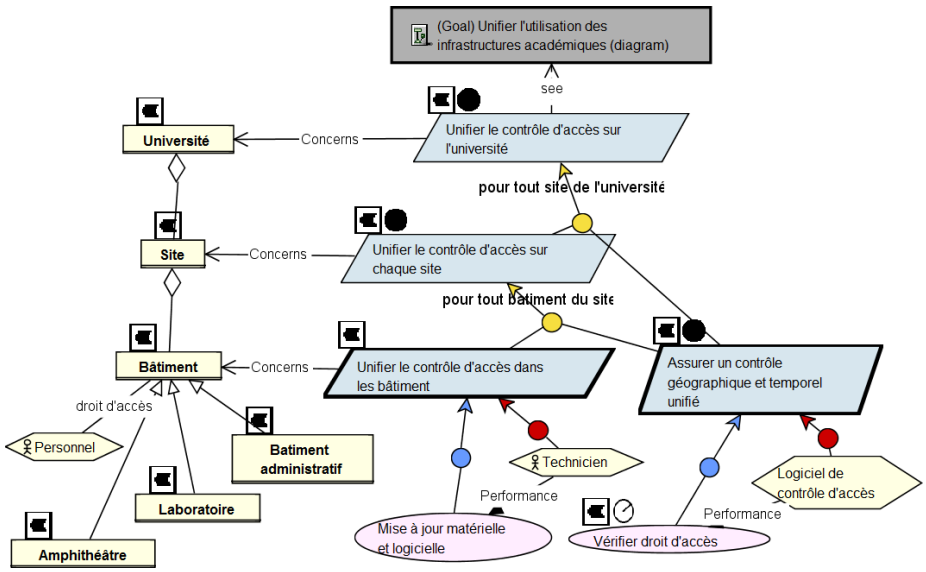


Figure 8. Propagation d'une exigence temporelle sur une structure spatiale

La figure montre comment différents sous-buts sont liés à des structures de nature spatiale. Des raisonnements quantitatifs sont également possibles. Par exemple, l'exigence de capacité d'accueil totale de l'université doit être préservée par la fusion. La capacité peut se calculer (récursivement) comme une *somme* des capacités de toutes les sous-entités de cette entité (en faisant l'hypothèse d'un attribut *capacité*). On peut aussi raisonner sur la capacité maximale et plus généralement imaginer d'autres opérateurs arithmétiques. Des algorithmes de propagation sont disponibles pour réaliser de tels calculs et peuvent être mis en œuvre (Darimont, Ponsard, 2015).

– **La transformation d'exigences spatiales en exigences temporelles.** Par exemple pour des déplacements, les distances spatiales sont généralement plus pertinentes à exprimer de manière temporelle parce qu'elles sont liées à d'autres contraintes (comme un rendez-vous, une tolérance maximale sur un temps acceptable). La conversion entre distance et temps peut faire apparaître des alternatives, par exemple liées au mode de déplacement considéré, voire des contraintes de planning (éviter les heures de pointe). Un exemple est présent dans la figure 9.

5.2. Analyse d'obstacles selon des critères spatio-temporels

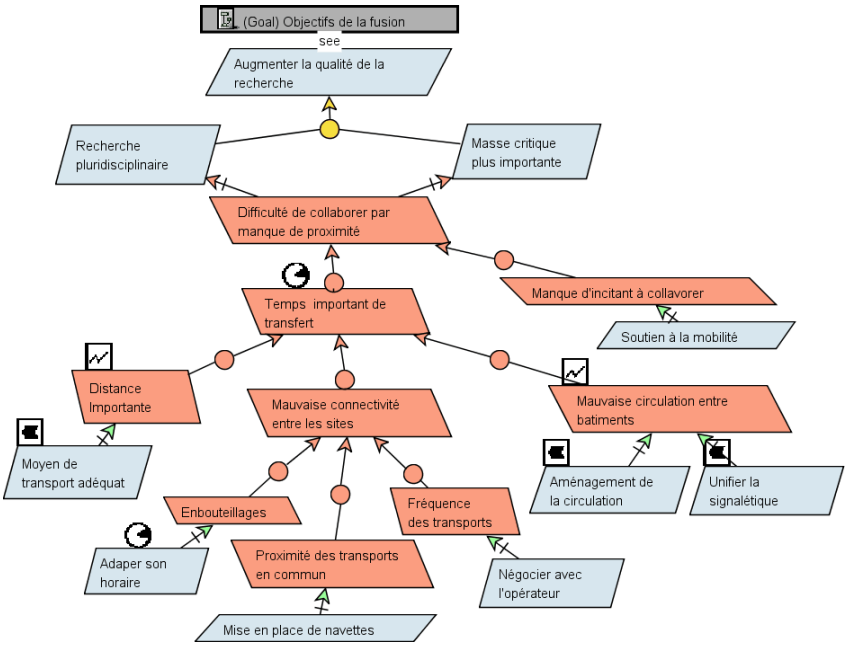


Figure 9. Analyse d'obstacles spatiaux et temporels

Les buts sont souvent formulés de manière trop idéalisée. Afin de les raffiner en exigences robustes. Il faut les mettre à l'épreuve de conditions adverses (risques) qui peuvent empêcher leur réalisation. Des techniques d'IE ont été définies pour générer des obstacles à des buts en commençant par des obstacles monolithiques puis en

permettant de les décomposer en causes élémentaires, pouvant alors être contrôlées (Lamsweerde, Letier, 2000). Parmi les techniques de raffinement d'obstacles, les techniques heuristiques se révèlent les plus utiles en pratique. Celles-ci s'appuient sur des altérations des contraintes exprimées par l'exigence, telle que l'altération d'une information reçue pour différentes causes et qui peuvent être explorées (par exemple en lien avec sa transmission, son stockage, son traitement, etc.). La présence de contraintes ST permet de générer des obstacles spécifiques. Il en découle une modélisation des exigences ST plus réaliste.

La figure 9 illustre plusieurs heuristiques utilisées pour générer des obstacles relatifs à des contraintes ST : un délai peut être causé par un temps de déplacement à une vitesse inférieure à celle planifiée, un temps d'attente (bouchon, attente d'une connexion), la nécessité de faire un détour, etc. Au niveau spatial, le terme d'obstacle peut reprendre son sens premier : l'absence de connexion physique entre deux bâtiments ou des caractéristiques les rendant impraticables pour des personnes à mobilité réduite (pente trop forte, escalier, etc.) (Ponsard, Snoeck, 2006).

6. Conclusion et perspectives

Sur la base d'un attirail de notations et de méthodes qui existent déjà autour de la dimension ST et répertoriées dans notre état de l'art, notre contribution propose d'établir des fondations pour un cadre unifié de modélisation d'exigences spatio-temporelles. Nous mettons en œuvre des notations graphiques parlantes ainsi qu'une formulation basée sur des mots clefs spécifiques aux exigences spatiales et temporelles. Le tout est guidé par une série de règles méthodologiques permettant de produire des exigences plus précises, complètes et robustes. Notre travail se situe dans un cadre de référence d'ingénierie des exigences orienté but, qui permet déjà de spécifier le "QUOI" et de justifier le "POURQUOI". Nous proposons de l'étendre pour mieux traiter le "QUAND" et le "OÙ". Nous gardons l'utilisateur au cœur d'un processus de développement des exigences en favorisant une bonne communication entre les parties prenantes via une vision explicite et graphique.

Nos travaux ont été appliqués à une étude de cas concrète et sont supportés par un outil pointu du domaine. Ce travail ne prétend cependant pas être exhaustif : à ce stade, il s'est principalement centré sur la définition d'un cadre unifié et sa validation. Afin d'être utilisable systématiquement, nos travaux se poursuivent dans les directions suivantes :

– Au niveau des notations, des patrons ST spécifiques peuvent également être identifiés pour le modèle des responsabilités et des opérations, afin d'apporter une guidance supplémentaire.

– Le cadre proposé doit être valable dans d'autres domaines d'application que les "SIG", notamment des systèmes mobiles et cyber-physiques. Ceci permettra de déterminer le niveau de généralité et la nécessité éventuelle de définir des règles ou patrons plus spécifiques à un domaine d'application. Cette démarche est soutenue par des études de cas supplémentaires.

– Une sémantique formelle peut-être envisagée à la fois pour les logiques temporelles (p.ex. logique temporelle linéaire) et spatiales (p.ex. algèbre d'Allen). Ceci permettrait de réaliser des vérifications automatiques ou des transformations vers des notations plus spécifiques à un domaine (p.ex. pour configurer un SIG). Le défi est cependant d'arriver à "cacher" ces logiques complexes derrière notre système de notations graphiques et patrons de formulation.

– Enfin, nous considérons l'adaptation des notations à d'autres outils, en particulier libres, tels que StarUML et ArgoUML qui disposent déjà d'extensions spécifiques, respectivement à l'IE et à l'IG.

Remerciements

Ce travail a été financé en partie par le projet PIT de la Région wallonne (conv. nr. 7481). Nous remercions Respect-IT pour la mise à disposition du SDK de son outil.

Bibliographie

Allen A., Yabushita S. (1984). On galaxy interactions during violent relaxation of clusters. *The Astrophysical Journal*, vol. 278, p. 468–468.

Becker R., al. (23-27 juillet 1990). Network visualization. *4th International Symposium on Spatial Data Handling, Zurich, Switzerland*.

Bédard Y., Larrivée S. (2008). Modeling with pictogrammic languages. *Shekar S, Xiong, H. (ed(s)), Encyclopedia of Geographic Information Sciences*, p. 716–725.

Cholley-Gomez M. (2015, 24 Janvier). *Nouvelle Université de Montpellier : une fusion réussie.* http://www.lenouveaumontpellier.fr/nouvelle-universite-montpellier-fusion-reussie.

Claramunt C., Jiang B. (2001). An integrated representation of spatial and temporal relationships between evolving regions. *Journal of Geographical Systems*, vol. 3, n° 4, p. 411–428.

Clementini E. (2009). *A Conceptual Framework for Modelling Spatial Relations*. Thèse de doctorat en informatique, INSA Lyon.

Darimont R., Lamsweerde A. van. (1996). Formal refinement patterns for goal-driven requirements elaboration. In *Proceedings of the 4th acm sigsoft symposium on foundations of software engineering*, p. 179–190. New York, NY, USA, ACM.

Darimont R., Ponsard C. (2015). Supporting quantitative assessment of requirements in goal orientation. In *23rd IEEE International Requirements Engineering Conference.*

Dwyer M. B., Avrunin G. S., Corbett J. C. (1999). Patterns in property specifications for finite-state verification. In *Proc. of the 21st int. conf. on soft. eng.* ACM.

Egenhofer M. J., Franzosa R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Systems*, vol. 5, n° 2, p. 161–174.

Frank A. U. (1994). Qualitative temporal reasoning in gis - ordered time scales. *In Waugh, T. C. et (eds.), R. C. H., éditeurs : Proc. of the 6th Int. Symposium on Spatial Data Handling.*.

Hughes D. L., Dwivedi Y. K., Simintiras A. C., Rana N. P. (2015). *Success and failure of is/it projects: A state of the art analysis and future directions* (1re éd.). Springer Int. Publishing.

IEEE1990. (1990). *Standard glossary of software engineering terminology.*

ISO29148. (2011, Dec). Systems and software engineering – life cycle processes –requirements engineering. *ISO/IEC/IEEE 29148:2011(E)*, p. 1-94.

Kosters G., Pagel B.-U., Six H.-W. (1996). Geoooa: object-oriented analysis for geographic information systems. In *Proc. of the 2nd int. conf. on requirements engineering*, p. 245-253.

Lamsweerde A. van. (2009). *Requirements engineering - from system goals to UML models to software specifications*. Wiley.

Lamsweerde A. van, Letier E. (2000, octobre). Handling obstacles in goal-oriented requirements engineering. *IEEE Trans. Softw. Eng.*, vol. 26, n° 10, p. 978–1005.

Laurini R., Thompson D. (1992). *Fundamentals of geographic information systems.* Academic Press Limited. 0-12-438380-7.

Le Parc-Lacayrelle A., Al. (2007). *Entreposage de documents et données semistructurées.* Hermes, Lavoisier.

Ligozat G. (1998). Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, vol. 9, n° 1, p. 23 - 44.

Mahaux M. (2004). *Vers une Spécification Allégée pour l'Analyse des Besoins Orientée-Objectifs.* Mémoire de fin d'étude, EPL, Université catholique de Louvain.

Manna Z., Pnueli A. (1992). *The temporal logic of reactive and concurrent systems*. New York, NY, USA, Springer-Verlag New York, Inc.

Mathian H., Sanders L. (2014). *Objets géographiques et processus de changement*. London, ISTE.

Papadias D., Kavouras M. (1994). Acquiring, representing and processing spatial relations. In *Presented at sixth international symposium on spatial data handling.* Taylor Francis.

Pinet F. (2012, juillet). Entity-relationship and object-oriented formalisms for modeling spatial environmental data. *Environ. Model. Softw.*, vol. 33, p. 80–91.

Ponsard C., Snoeck V. (2006, July). Objective accessibility assessment of public infrastructures. In *Computers helping people with special needs, 10th int. conf. linz, austria*, p. 314–321.

Pullar D. V., Egenhofer M. J. (1988). Towards the defaction and use of topological relations among spatial objects. *In Proc. of the 3rd Int. Symposium on Spatial Data Handling*.

Randell D., Cui Z., Cohn A. (1992). A spatial logic based on regions and connection. In *Proc. 3rd int. conf. on knowledge representation and reasoning.* San Mateo, Morgan Kaufmann.

Respect-IT. (2005). *Objectiver Requirements Engineering Tool.* http://www.respect-it.com.

Rolland C., Salinesi C. (2005). Engineering and managing software requirements. In A. Aurum, C. Wohlin (Eds.),, p. 189–217. Berlin, Heidelberg, Springer Berlin Heidelberg.

Touzani M., Anne L., Libourel T., Quinqueton J. (2015, avril). Towards Geographic Requirements Engineering. In *KMIKS'2015.* Hammamet, Tunisia.

Zimmermann K., Freksa C. (1996). Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence*, vol. 6, p. 49–58.

Zoghlami A. (2013). *Modélisation et conception de systèmes d'information géographique gérant l'imprécision*.

La Validation dans le Processus de Développement

Imen Sayar — Jeanine Souquières

LORIA – CNRS UMR 7503 – Université de Lorraine
Campus Scientifique, BP 239
F-54506 Vandœuvre lès Nancy cedex
{Firstname.Lastname}@loria.fr

RÉSUMÉ. *L'amélioration de la qualité d'un logiciel commence par l'expression de ses exigences en langage naturel. Notre objectif est de combler l'écart entre le cahier des charges, celui du client, et la spécification formelle, celle de l'informaticien. Dans ce papier, nous abordons la validation tout au long du processus de développement, en tenant compte des exigences et la spécification en Event-B du logiciel. La vérification peut aussi détecter des incohérences dans les exigences. La place des outils, notamment avec la plateforme Rodin, est importante au long de ce processus, améliorant sa qualité et sa documentation. Notre approche est illustrée par l'étude de cas d'un système de contrôle du train d'atterrissage d'un avion.*

ABSTRACT. *The amelioration of the quality of a system begins by the requirements elicitation. Our goal is to bridge the gap between requirements, those of the client, and the specification, this of the computer scientist. In this paper, we talk about the validation all along the development of a system, taking into account its requirements and its Event-B specification. The verification may also detect incoherences in the requirements. The Rodin platform is important all along to improve the quality and the documentation of the system, both of its specification and its development process. We illustrate our approach on the case study of an aircraft landing system.*

MOTS-CLÉS : *exigences, spécification, raffinement, validation, vérification, outils*

KEYWORDS: *requirements, specification, refinement, validation, verification, tools*

1. Introduction

Un travail de recherche sur l'ingénierie des exigences présenté dans (van Lamsweerde, 2008) insiste sur deux activités à résoudre : l'analyse du domaine et la modélisation des exigences. Le groupe Standisch a conduit des études par l'interview d'entreprises dans le domaine du logiciel. Il a récemment publié une dernière version du Rapport CHAOS dont le premier date de 1994 [1] : l'une des principales causes des difficultés dans le développement de systèmes réside dans la prise en compte des exigences. Ces exigences sont souvent très pauvres (Abrial, 2009), voire inexistantes.

Le processus de développement de systèmes à l'aide du raffinement utilisé dans Event-B est comparable au processus de la cascade : le modèle initial précise son invariant que le système doit garantir et chaque raffinement est à nouveau prouvé par son invariant. Cette approche assume les propriétés suivantes : 1) toutes les exigences sont explicitées pour décrire le modèle initial ; 2) le modèle initial est une description formelle répondant aux exigences de sécurité et fonctionnelles et 3) toutes les décisions prises lors d'un raffinement doivent être mémorisées relativement à une exigence. En réalité, peu de développements décrivent entièrement ces propriétés. Les exigences évoluent avec le développement, toutes les exigences ne sont pas forcément exprimées dans le modèle initial, de nombreux raffinements introduisent de nouvelles informations dans le modèle (Abrial, 2006).

Les activités de validation et de vérification sont utilisées à chaque étape du développement. Parmi les techniques de validation de modèles, leur exécution est plus attrayante, particulièrement dans le cadre de la modélisation à base de modèles Event-B. La plus grande difficulté vient du non-déterminisme de la plupart des modèles raffinés. En fait, il est recommandé de réduire l'abstraction et le non-déterminisme pas-à-pas. Tandis que des outils actuels, tels que ProB (Leuschel *et al.*, 2003), peuvent animer des modèles avec un non-déterminisme modéré, ils ne peuvent pas toujours traiter l'ensemble des problèmes soulevés. Des stratégies d'exploration exhaustives tombent rapidement dans l'explosion combinatoire.

Dans notre approche, nous mémorisons le cahier des charges, nous prenons en compte ses différentes mises à jour et nous explicitons sa place dans le processus de développement (Abrial, 2009). Pour cela, un ensemble de liens, ou relations, entre exigences et spécifications a été identifié et réalisé avec la plateforme Rodin. Ces liens sont mis à jour tout au long du processus du développement, depuis le cahier des charges jusqu'à la spécification terminée. Différentes actions entre l'évolution des deux "mondes" sont présentées, voir figure 1. Le choix d'une ou plusieurs exigences nous amène à la spécification en cours de développement. Depuis la spécification, les actions modifiant les exigences permettent :

- d'ajouter des termes formels dans une exigence,
- d'ajouter, supprimer et mettre à jour une exigence,
- de valider et vérifier la spécification vis-à-vis des exigences à l'aide des outils disponibles.

La spécification est en cours de développement et décrite pas à pas. L'ensemble des outils disponibles, tels que les outils de validation et de vérification, sont utilisés tout au long de ce développement, et non seulement lorsque la spécification est terminée.

1. Rapport CHAOS du Standish Group (http ://www.standishgroup.com)

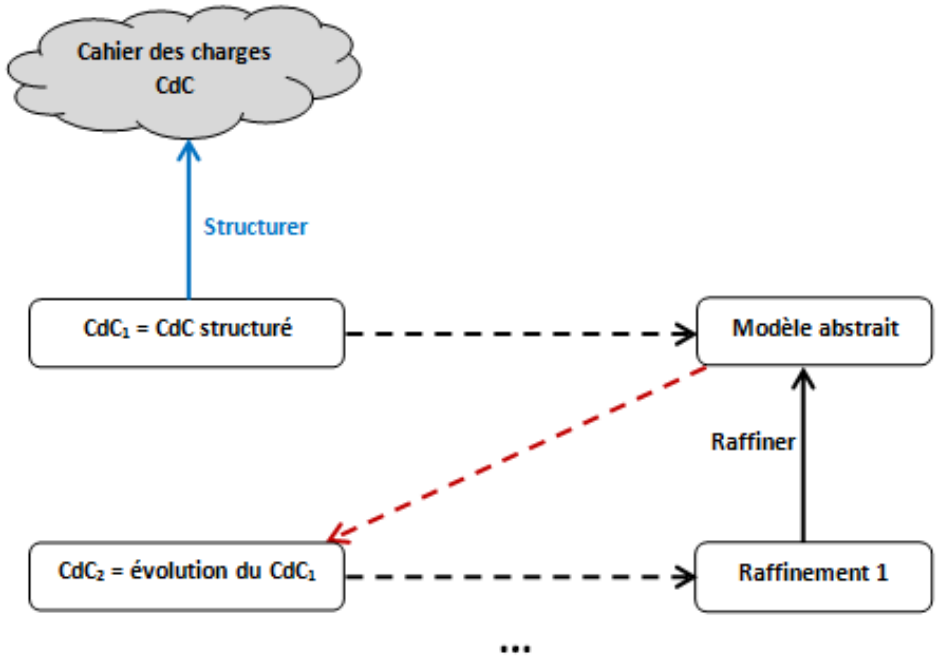


Figure 1. *Notre approche*

Dans la suite de cet article, la partie 2 présente brièvement la méthode Event-B, la plateforme Rodin et l'outil ProR pour prendre en compte les besoins informels en lien avec la spécification en Event-B. La partie 3 décrit la prise en compte de la validation dès la première étape du processus de développement, celle de l'expression de ses exigences en langage naturel. La partie 4 aborde la détection d'incohérences dans les exigences en liaison avec la vérification. Notre approche est illustrée par l'étude de cas d'un système de contrôle du train d'atterrissage d'un avion (Boniol *et al.*, 2014). L'état de l'art sur le sujet est abordé dans la partie 5. Une conclusion est proposée dans la partie 6 avec nos travaux futurs.

2. Cadre formel et outils support

2.1. *La méthode Event-B*

Event-B est une méthode formelle pour spécifier et modéliser des systèmes complexes à partir de la notion de machines abstraites et du raffinement (Abrial, 2010). Une spécification est décomposée en deux parties : (1) le contexte pour décrire la partie statique du modèle à l'aide des ensembles énumérés, des constantes et des axiomes ; (2) la machine pour décrire la partie dynamique à l'aide des variables et des événements. Event-B permet de modéliser un système et son environnement à travers la notion d'observation des événements.

– Description du système. Il est modélisé par un état ou une fonction associant des noms à des valeurs, et contraint par un invariant qui circonscrit l'ensemble des valeurs licites que peut prendre cet état. L'invariant est une formule logique du premier ordre portant sur les valeurs des variables et des constantes. Un événement est une substitution gardée sur l'état. La garde est un prédicat du premier ordre sur l'état. La structure générale d'une machine en Event-B est la suivante :

MACHINE <*Machine_identifier*> **REFINES** <*Abstract_machine_identifier*> **SEES** < *Context_identifier_list* > **VARIABLES** < *variable_identifier_list* > **INVARIANTS** <label>: <*predicate*> ...	**EVENTS** *Event* < *event_identifier* > **REFINES** < *abstract_event_identifier* > **WHEN** <label>: <*predicate*> **THEN** <label>: <*action*> **END** **END**

– Sémantique. Elle est liée à la notion de correction. Le modèle doit être réalisable, l'ensemble des états licites n'est pas vide et les événements relient des états licites. L'invariant est préservé lorsqu'un événement est déclenché depuis un état licite. Le raffinement maintient l'invariant abstrait : il existe une fonction d'abstraction reliant l'état du modèle concret au modèle abstrait, et chaque événement concret maintient l'invariant abstrait. Les propriétés de chaque modèle ainsi que celles des raffinements sont données par un ensemble d'obligations de preuve, appelées POs.

– Développement. Le développement de spécifications Event-B s'effectue par une approche générale progressive. Il s'agit du raffinement ou relation qui lie deux modèles par l'enrichissement de l'un par un autre. La correction des raffinements est définie par les POs qui garantissent que les invariants de la machine précédente sont préservés par le raffinement.

2.2. *La plateforme Rodin*

Rodin [2] est une plateforme autour d'Event-B développée avec Eclipse. Elle est étendue à l'aide de "plug-ins" et fournit un soutien pour le raffinement et la preuve mathématique. Elle permet d'éditer, animer, prouver et contre-prouver les modèles. Parmi les plug-ins, deux outils complémentaires à la preuve sont indispensables pour la validation des modèles Event-B. Il s'agit d'un animateur permettant de détecter un ensemble de problèmes comme l'inter-blocage ou des comportements non autorisés. Le deuxième outil est un contre-prouveur qui aide à la preuve interactive avec ProB pour trouver des contre-exemples, via un accès direct aux obligations de preuve.

ProR est un plug-in de Rodin pour exprimer une structure hiérarchique des exigences (Jastram, 2010). L'approche proposée commence par l'élicitation des exigences initiales et les hypothèses du domaine. Elle ne propose pas de notation particulière mais un classement des artefacts. Elle s'exprime de la manière suivante :

2. http ://wiki.event-b.org

ID	Description
FUN-G	The landing system's goal is maneuvring gears and their associated doors
FUN-G-1	Maneuvring gears consists on extending or retracting them and reversing their mouvement

Elle est basée sur le modèle de référence WRSPM (Gunter *et al.*, 2000). Elle crée manuellement les liens entre exigences et éléments du modèle Event-B en cours de construction, ces liens pouvant être annotés. Pour gérer la traçabilité entre exigences et modèles formels, ProR permet de :

- définir manuellement des liens depuis la spécification Event-B vers les exigences texte sous ProR. Initialement, les exigences sont informelles,
- insérer des éléments formels dans les exigences de ProR, ces éléments étant issus de la spécification Event-B.

Cet outil enregistre des informations importantes pour les activités de validation et de vérification.

3. Validation

Nous abordons la validation tout au long du développement. Un premier travail commence par la compréhension des exigences et à les re-structurer, si nécessaire. Au fur et à mesure du développement de la spécification sous Rodin et en utilisant ProR, les exigences structurées sont mises à jour pour :

- ajouter des liens entre exigences et éléments formels de la spécification en cours,
- introduire des termes formels dans les exigences structurées, termes issus de la spécification. Ces termes entre crochets [] sont distingués du reste de l'exigence.

3.1. *Structuration des exigences*

La structuration du cahier des charges (CdC) a pour objectif l'obtention d'un document lisible et accessible par toutes ses parties prenantes. Dans notre étude de cas, celui d'un système de contrôle du train d'atterrissage d'un avion, le CdC est compréhensible en l'état. Par conséquent, notre activité consiste à ré-écrire ce document sous forme de phrases étiquetées, en utilisant la proposition de (Abrial, 2010). La validation est prise en compte dès la structuration du CdC et tout au long du processus de développement du système. Elle concerne les futurs modèles formels en Event-B par rapport aux exigences du client. Ces exigences s'expriment sous forme des :

- données du futur système (notées Fact dans la suite de ce document),
- fonctionnalités attendues (Functionality),
- comportements (Behavior) ou séquences de fonctions prévues ou non autorisées,
- conditions (Obligation) avec lesquelles le système fonctionne sous forme de pré-conditions et post-conditions (Hoare, 1969).

Cette phase du développement concerne les deux étapes suivantes :

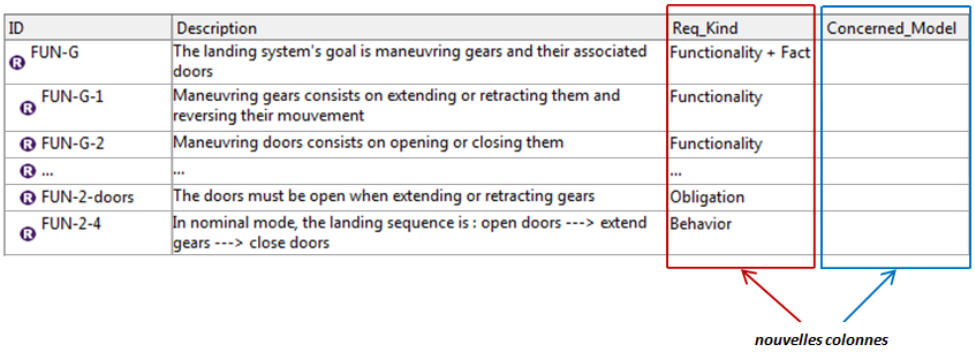


Figure 2. *Prise en compte de la validation avec ProR*

Req_Kind	**Extracted element**	**ID**
Fact	- gears - associated doors	FUN-G FUN-G
Functionality	- maneuvring gears and doors - maneuvring gears : extend gears, retract gears, reverse gears mouvement - maneuvring doors : open doors, close doors	FUN-G FUN-G-1 FUN-G-2
Behavior	- landing sequence : open doors –> extend gears –> close doors	FUN-2-4
Obligation	- *pre-condition* doors = open - *post-condition* extend gears, retract gears	FUN-2-doors

Tableau 1. *Eléments pour la validation*

1) La première consiste à ajouter des paramètres au document structuré avec ProR. La structure du CdC dispose de deux nouvelles colonnes contenant des informations relatives à la validation :

- La colonne *Req_Kind*. Un élément de cette colonne indique les catégories de l'exigence correspondante : *Fact*, *Functionality*, *Behavior* ou *Obligation*.
- La colonne *Concerned_Model*. Un élément de cette colonne fait le lien entre l'exigence et le nom du modèle Event-B correspondant, via des machines et des contextes.

Exemple. La figure 2 est décrite à l'aide de ProR. Elle contient ces deux nouvelles colonnes du CdC structuré. Le travail de spécification n'a pas encore commencé ; donc, nous n'avons pas encore introduit de modèles lors de cette étape.

2) La deuxième consiste à extraire les éléments destinés à la validation du futur modèle formel, présents dans les exigences du client. Ces éléments de type *Fact*, *Functionality*, et *Obligation* sont extraits en se posant certaines questions :

- Quelles sont les données présentes dans le futur système ?

- Quelles sont ses fonctionnalités attendues ?
- Sous quelles conditions le système devra fonctionner ?

Les comportements extraits du CdC sont décrits en termes de scénarios de validation. Ces scénarios décrivent les différents états du modèle Event-B en animant des évènements. Ils apparaissent dans le CdC structuré sous forme de phrases décrivant un enchaînement d'actions.

Exemple. Les éléments de validation extraits de la figure 2 sont présentés dans le tableau 1, relativement à chaque catégorie des exigences.

3.2. *Évolution des exigences et de la spécification*

Cette partie est illustrée par l'étude de cas à l'aide des différentes sortes d'exigences proposées dans la partie 3.1. Nous introduisons la machine `Gears_M0` et son contexte `Gears_Ctx0`. La colonne Concerned_Model introduite dans la figure 2 est mise à jour par les deux nouveaux noms de modèles `Gears_Ctx0` et `Gears_M0`. Ces noms correspondent au `context` et à la `machine` d'Event-B liés aux deux exigences FUN-G et FUN-G-1. L'état du développement est présenté dans la figure 3. La spécification Event-B en cours est proposée dans l'annexe.

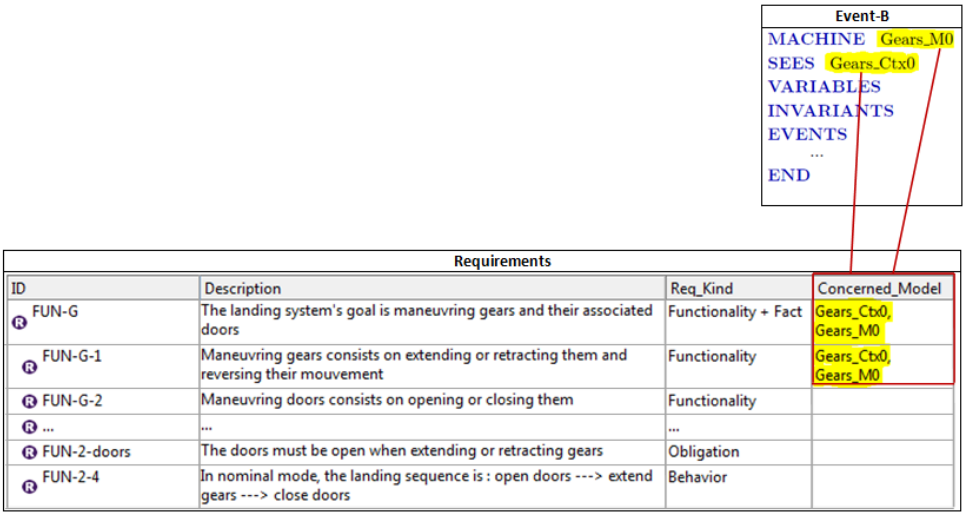


Figure 3. *Introduction d'une machine et son contexte*

3.2.1. *Introduction d'une donnée*

L'introduction de la variable `gears_pos` dans la machine `Gears_M0` se répercute dans la description des exigences : le terme informel `gears` dans les exigences FUN-G et FUN-G1 est remplacé par le terme formel `[gears_pos]` issu de la spécification `Gears_M0`. Les liens entre la spécification Event-B et les besoins sous ProR sont mis à jour. La figure 4 montre l'introduction de la variable `gears_pos`.

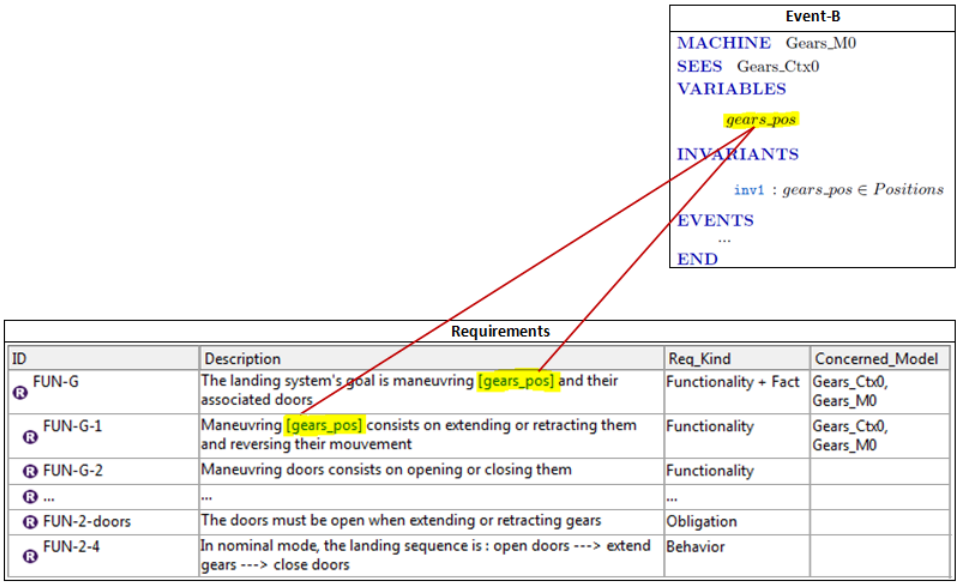


Figure 4. *Introduction d'une donnée*

3.2.2. *Introduction d'une fonctionnalité*

La figure 5 présente l'introduction de l'événement `open_doors` dans la spécification existante et dans le texte des exigences FUN-G-2 et FUN-2-4. La mise à jour du développement concerne :
- la spécification avec un nouvel événement,
- les nouveaux éléments formels dans les exigences,
- les liens entre eux, via ProR.

3.2.3. *Introduction d'une condition*

La figure 6 présente l'introduction de la garde `doors = open` pour les deux événements `extend_gears` et `retract_gears` dans la spécification existante. La mise mise à jour de l'exigence FUN-2-doors concerne :
- une garde dans deux événements existants dans la spécification,
- une nouvelle variable, `doors_pos`,
- les nouveaux éléments formels dans les exigences,
- les liens entre eux, via ProR.

3.3. *Validation de la spécification relativement au CdC*

Afin de valider chacun des modèles Event-B en cours de développement, relativement aux éléments de validation issus lors de la phase de structuration du CdC, nous procédons aux deux étapes suivantes.

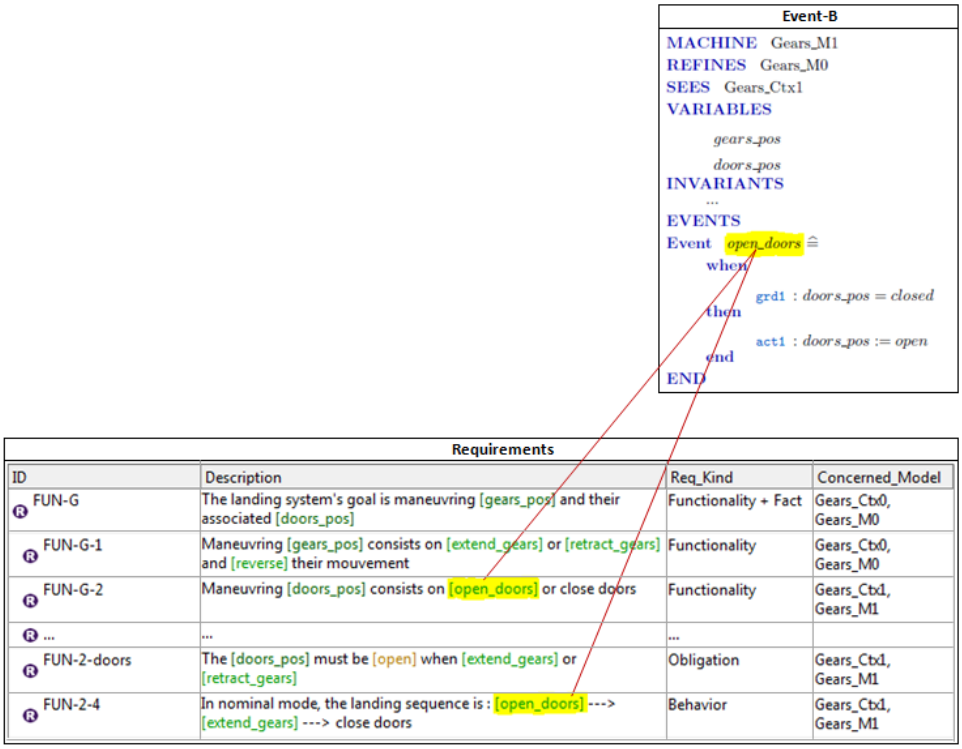


Figure 5. *Introduction de la fonction open_doors*

- Étape 1. Recherche d'éléments de types Fact, Functionality et Obligation dans le modèle décrit précédemment.

  Exemple. Dans la machine `Gears_M1`, les deux variables `gears_pos` et `doors_pos` correspondent aux éléments extraits de type Fact du tableau 1. Concernant la validation des éléments de type Functionality,

  - l'élément *maneuvring gears and doors* n'a pas été modélisé par notre choix de stratégie de développement,
  - les autres éléments de type Functionality ont été pris en compte.

  Ce résultat est présenté dans le tableau 2. L'élément de type Obligation a été modélisé par une garde nommée `grd_FUN2-2-doors` (voir la figure 6).

- Étape 2. Validation par animation du comportement du modèle exprimé par les éléments de type Behavior. L'animation est réalisée avec ProB en utilisant des scénarios.

  Exemple. Pour animer le modèle `Gears_M1` par rapport au scénario décrit dans le tableau 1, nous avons simulé la séquence d'événements Event-B suivante :

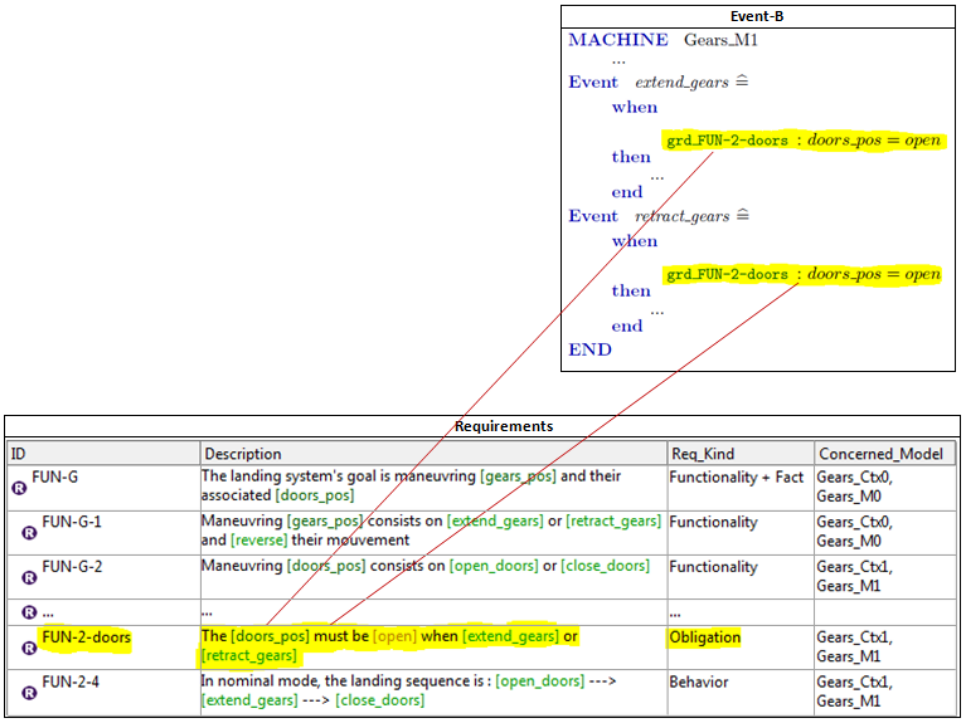


Figure 6. *Introduction d'une condition dans un événement*

Functionality	**Event-B element**
maneuvring gears and doors	non modélisé (choix de développement)
extend gears	event `extend_gears`
retract gears	event `retract_gears`
reverse gears mouvement	event `reverse`
open doors	event `open_doors`
close doors	event `close_doors`

Tableau 2. *Prise en compte des éléments de type Functionality*

```
INITIALISATION -> open_doors -> retract_gears -> close_doors
```

3.4. *Scénarios de validation pour l'animation*

Le CdC peut contenir plusieurs scénarios décrivant les comportements du futur système. Cependant, ces scénarios ne sont pas suffisants pour couvrir tous les comportements possibles. Afin de définir de nouveaux scénarios non prévus dans le CdC,

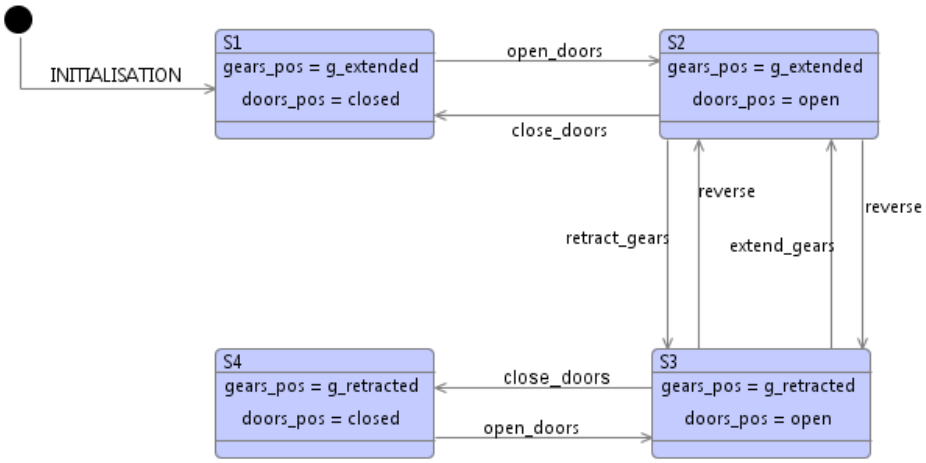


Figure 7. *Une sous-machine à états de* `Gears_M1`

nous utilisons les machines à états, State Machine SM, et une recherche de scénarios non permis par le futur système.

3.4.1. *Utilisation des machines à états*

Afin de construire la SM correspondante à une machine Event-B, nous utilisons Event-B Statemachines [3], un plugin de la plateforme Rodin. A chaque modèle, une SM est associée :

- chaque état représente une valeur donnée à chacune des variables Event-B,
- chaque transition entre deux états représente un événement.

Exemple 1. La figure 7 présente la SM correspondante à la machine Event-B `Gears_M1` de la figure 8. Cette SM décrit la séquence de rétraction et d'extension des trains. Ses états sont décrits par les valeurs de ses variables `gears_pos` et `doors_pos`. Le passage de l'état **S1** à l'état **S2** s'effectue par l'événement `open_doors` ; la variable `doors_pos` passe de `closed` à `open`.

Exemple 2. A partir de la figure 7, de nouveaux scénarios sont proposés dans lesquels des enchaînements d'états ne sont pas autorisés :

- *initialisation -> retract gears* : ce scénario passe de **S1** à **S3**. Il est interdit car il n'est pas possible de rétracter les trains sans ouvrir les portes.
- *initialisation -> close doors* : il n'y a pas de passage immédiat entre un état de trains étendus (état **S1**) à un autre un état de trains pliés (état **S4**) sans passer par la séquence de rétraction.

3. http ://wiki.event-b.org/index.php/Event-B_Statemachines

- *initialisation -> open doors -> retract gears -> retract gears -> close doors* : il n'est pas possible de rétracter les trains plus d'une fois dans la même séquence.

Ces scénarios ne sont pas permis par la machine `Gears_M1` lors de son animation avec ProB.

3.4.2. *Scénarios non autorisés*

A partir de la compréhension des exigences, nous introduisons une action décrivant un comportement non souhaité dans un scénario décrivant un comportement existant.

Exemple. A partir du scénario de validation décrit dans le tableau 1,

open doors –> extend gears –> close doors,

l'action *open doors* est introduite à nouveau avant que la porte soit fermée. L'objectif de ce scénario est de vérifier si l'ouverture des portes des trains d'atterrissage se réalise une deuxième fois dans une séquence d'extension des trains. Ce scénario a la forme suivante :

open doors –> extend gears –> ***open doors*** *–> close doors*

La machine `Gears_M1` a interdit l'animation de ce scénario, à l'aide de ProB. La garde de l'événement `open_doors` interdit sa simulation lorsque les portes sont déjà ouvertes.

4. Vérification

L'activité de vérification permet de découvrir des incohérences dans les exigences. Celles-ci peuvent correspondre à des contradictions, des répétitions ou des oublis. Dans l'étude de cas, nous avons ajouté l'exigence suivante :

ID	Description	Req_Kind
FUN-i	The gears can move only if doors are closed	Obligation

Cette exigence appelée FUN-i est modélisée en Event-B par l'invariant suivant :

```
gears_moving = TRUE => doors_pos = closed
```

Dans cet invariant, `gears_moving` est une variable booléenne qui indique si les trains sont en déplacement ou non (voir figure 8).

Avant l'ajout de cet invariant, le modèle en Event-B était mathématiquement correct. Après l'introduction de cet invariant, les prouveurs de Rodin ont donné lieu à un ensemble d'obligations de preuve non déchargées. Une de ces obligations de preuve a précisé que l'événement `extend_gears` ne respecte pas ce nouvel invariant :

- Une contradiction entre la garde et cet invariant :
  - garde : `doors_pos = open`
  - invariant : `gears_moving = TRUE => doors_pos = closed`

- Cette garde et cet invariant modélisent deux exigences du CdC. Une contradiction entre ces exigences est détectée à l'aide des prouveurs de Rodin.

5. Etat de l'art

Les documents des exigences utilisés dans l'industrie sont souvent pauvres et difficiles à comprendre (Abrial, 2006). Dans (Su *et al.*, 2011), les auteurs recommandent de ré-écrire ce document sous la forme de deux textes différents. L'un est destiné à la compréhension du problème et l'autre contient les définitions et les besoins en termes de phrases courtes dans un langage naturel.

Notre approche commence par la re-structuration du document des exigences proposée par Abrial pour exprimer les liens entre les exigences et les spécifications ; nous ajoutons pas-à-pas des liens avec les modèles Event-B en cours de développement. La première étape de structuration est celle proposée par Abrial. Les autres étapes utilisent l'outil ProR (Jastram, 2010). Nous proposons une évolution de la structure des exigences avec la préparation de l'activité de la validation. De plus, nous ajoutons de nouveaux liens entre la spécification Event-B et les exigences.

Dans l'approche présentée dans (Heisel *et al.*, 1999), les exigences et la spécification sont clairement identifées. Un guide méthodologique détaillé pour ces deux activités est proposé et n'introduit pas de nouveaux langages ou formalismes. Le processus d'élicitation des besoins est indépendant du langage de spécification utilisé. Un lien de traçabilité entre les exigences et la spécification est maintenu via les agendas. Notre approche est une évolution liée à l'utilisation de la plateforme Rodin et l'outil ProR.

L'approche KAOS (van Lamsweerde, 2009) est orientée buts ; elle permet de les identifier et de les raffiner progressivement jusqu'à l'obtention des contraintes. La méthode associée propose de dériver les besoins en termes d'objets et d'actions. Un modèle multi-vues intègre tous les concepts du langage utilisé, pour articuler les exigences et les spécifications (van Lamsweerde, 2008). A partir de plusieurs applications industrielles décrites à l'aide de KAOS, la présentation de (Ponsard *et al.*, 2015) considère les interactions entre les artifacts des exigences ; des outils supports sont nécessaires. Dans notre approche, les exigences doivent être comprises avant de les structurer. Dans l'approche proposée dans (Ponsard *et al.*, 2015), les auteurs utilisent plusieurs sortes de diagrammes pour aider à comprendre les exigences.

L'évolution des besoins est prise en compte (Hallerstede *et al.*, 2014). Cela signifie que les exigences doivent être fréquemment changée et incorporée incrémentalement dans la spécification formelle. Dans notre approche, nous mémorisons les liens entre exigences et spécifications tout au long du développement avec les actions qui les font évoluer.

(Driss, 2014) propose un modèle pivot basé sur une ontologie, pour faire le lien entre exigences et spécifications formelles. Dans notre approche, nous utilisons les liens offerts par ProR et la plateforme Rodin pour relier les exigences avec le modèle formel en cours de développement.

Les outils d'aide à la validation de modèles Event-B (Mashkoor *et al.*, 2016a; Mashkoor *et al.*, 2016b) ont mis en évidence la possibilité de définir des sémantiques

mathématiques qui garantissent la correction de modèles exécutables. Ils ont esquissé une extension de la notion de raffinement en tant qu'étape dans le processus de développement. Suite à ce travail, notre approche propose une meilleure intégration entre le formel et le semi-formel dans le processus de développement.

Dans l'approche décrite dans (Alkhammash *et al.*, 2015), des structures semi-formelles sont utilisées pour combler l'écart entre les exigences et les modèles Event-B. Cette approche conserve la trace entre les exigences et les modèles Event-B. Dans une première étape, les exigences sont classées et affectées à des composants Event-B. Dans la deuxième étape, des détails sont introduits graduellement dans les modèles formels. La troisième étape est d'utiliser l'outil UML-B et l'outil de décompostion atomique pour générer des modèles Event-B. Une différence entre notre présentation et celle-ci concerne le fait que notre approche n'est pas seulement pour des modèles Event-B, mais elle est basée sur la combinaison des exigences évolutives alternativement avec les étapes de raffinement des modèles Event-B.

Dans (Hallerstede *et al.*, 2011), un algorithme en ProB a été décrit pour l'animation du raffinement de plusieurs niveaux simultanés. Cette animation permet de détecter une variété d'erreurs introduites avec le raffinement. Ces résultats sont empiriques ; la preuve et l'animation se complètent relativement à la validation. Nous utilisons cette proposition pour raisonner tout au long du développement en utilisant le raffinement.

6. Conclusion et perspectives

Dans ce papier, nous avons pris en compte la validation tout au long du processus de développement, depuis la structuration des exigences jusqu'à la spécification en Event-B d'un logiciel. Mémoriser les exigences implique sa mise à jour et sa place dans le processus de développement (Abrial, 2009). Nous avons utilisé la plateforme Rodin et illustré notre approche par l'étude de cas d'un système de contrôle du train d'atterrissage d'un avion [4]. Nous avons aussi développé une étude de cas d'une machine d'hémodialyse (Mashkoor, 2015).

La validation est prise en compte dès la première étape du processus de développement, celle de l'expression de ses exigences en langage naturel. Nous ré-écrivons ces exigences sous forme de phrases étiquetées. Nous ajoutons de nouveaux paramètres au document structuré avec ProR et décrivons des scénarios de validation.

Dans les étapes suivantes, nous ajoutons des termes formels dans les exigences à partir de la spécification Event-B. La mise à jour des exigences est prise en compte. La préparation à la validation est guidée par les types des éléments formels. L'animation et l'aide à la validation sont guidées par la définition de nouveaux scénarios dans lesquels des enchaînements d'actions ne sont pas autorisés ou non précisés dans les exigences. La vérification permet de détecter des incohérences dans les exigences.

Aujourd'hui, la compréhension et l'utilisation directe du cahier des charges, lorsqu'il existe, n'a pas encore atteint la maturité suffisante pour proposer une démarche de développement. L'évolution des outils disponibles, comme par exemple la plateforme Rodin permettant d'éditer, animer, prouver et contre-prouver les modèles, apporte un point de vue important pour combler l'écart entre ces deux mondes, celui

4. Ce développement est accessible à l'adresse http ://dedale.loria.fr

des exigences et celui des spécifications formelles. Si nous regardons notre approche présentée dans la figure 1, le raffinement utilisé en Event-B pourrait être généralisé à l'ensemble des couples (besoins, spécifications) et des liens entre eux. Une partie du développement pourrait être outillée, voire automatisée.

La notion de liens introduite dans ce papier est importante dans le processus de développement d'un logiciel et dans sa validation. A ce jour, ces liens sont extraits des différents documents disponibles, qu'ils soient formels ou informels. Ils sont utilisés manuellement dans le processus. Cette notion doit être clarifiée. Concernant la validation, un travail sur des outils manuels et semi-automatiques est nécessaire en lien avec les différentes catégories des exigences.

7. Bibliographie

Abrial J.-R., « Formal Methods in Industry : Achievements, Problems, Future », *in* L. J. Osterweil, H. D. Rombach, M. L. Soffa (eds), *28th International Conference on Software Engineering, Shanghai, China*, ACM, p. 761-768, 2006.

Abrial J.-R., « Faultless Systems : Yes We Can ! », *IEEE Computer*, vol. 42, n° 9, p. 30-36, 2009.

Abrial J.-R., *Modeling in Event-B : System and Software Engineering*, Cambridge University Press, 2010.

Alkhammash E., Butler M., Fathabadi A. S., Cirstea C., « Building Traceable Event-B Models from Requirements », *Science of Computer Programming (Special Issue on Automated Verification of Critical Systems, AVoCS 2013)*, vol. 111, Part 2, p. 318-338, 2015.

Boniol F., Wiels V., « Landing Gear System case Study », *ABZ Conference, Communications in Computer and Information Science, Springer*, vol. 433, p. 1-18, 2014.

Driss S., From natural language specifications to formal specifications via an ontology as a pivot model, Theses, Université Paris Sud - Paris XI, June, 2014.

Gunter C. A., Gunter E. L., Jackson M., Zave P., « A Reference Model for Requirements and Specifications- Extended Abstract », *Proceedings of the 4th International Conference on Requirements Engineering*, IEEE Software, p. 17 : 37-43, 2000.

Hallerstede S., Jastram M., Ladenberger L., « A Method and Tool for Tracing Requirements into Specifications », *Sciences Computer Program*, vol. 82, p. 2-21, 2014.

Hallerstede S., Leuschel M., Plagge D., « Validation of Formal Models by Refinement Animation », *Sci. Comput. Program.*, vol. 78, n° 3, p. 272-292, 2011.

Heisel M., Souquieres J., « A Method for Requirements Elicitation and Formal Specification », *Proc. 18th Int. Conference on Conceptual Modeling, ER'99*, n° 1728 in *LNCS Springer-Verlag*, p. 309-324, 1999.

Hoare C. A. R., « An Axiomatic Basis for Computer Programming », *Commun. ACM*, vol. 12, n° 10, p. 576-580 & 583, 1969.

Jastram M., « ProR, an Open Source Platform for Requirements Engineering based RIF », *SEISCONF*, 2010.

Leuschel M., Butler M. J., « ProB : A Model Checker for B », *in* K. Araki, S. Gnesi, D. Mandrioli (eds), *International Symposium of Formal Methods Europe, Pisa, Italy, Proceedings*, vol. 2805 of *LNCS*, Springer, p. 855-874, 2003.

Mashkoor A., The Hemodialysis Machine Case Study, Technical report, Tecnical report SCCH-TR-1542, Austria : Software Competence center Hagenberg GmbH, 2015.

Mashkoor A., Jacquot J.-P., « Validation of Formal Specifications through Transformation and Animation », *Requirements Engineering, Springer Verlag, 16 pages*, 2016a.

Mashkoor A., Yang F., Jacquot J.-P., « Refinement-based Validation of Event-B Specifications », *Software and Systems Modeling, Springer Verlag, 33 pages*, 2016b.

Ponsard C., Darimont R., Michot A., « Combining Models, Diagrams and Tables for Efficient Requirements Engineering : Lessons Learned from the Industry », *Actes du XXXIIIème Congrès INFORSID, Biarritz, France, May 26-29*, p. 235-250, 2015.

Su W., Abrial J.-R., Huang R., Zhu H., « From Requirements to Development : Methodology and Example », *13th International Conference on Formal Engineering Methods, Durham, UK*, p. 437-455, 2011.

van Lamsweerde A., « Requirements Engineering : from Craft to Discipline », *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Atlanta, Georgia, USA*, p. 238-249, 2008.

van Lamsweerde A., *Requirements Engineering - From System Goals to UML Models to Software Specifications*, Wiley, 2009.

A. Annexe

MACHINE *Gears_M1*
REFINES *Gears_M0*
SEES *Gears_Ctx1*
VARIABLES
gears_pos
doors_pos
gears_moving
INVARIANTS
inv1 : $doors\_pos \in Doors\_Positions$
FUN−i_inv: $gears\_moving = TRUE \Rightarrow doors\_pos = closed$

EVENTS
INITIALISATION
BEGIN
act1 : $gears\_pos := g\_extended$
act2 : $doors\_pos := closed$
END
Event extend_gears
REFINES
extend_gears
WHEN
grd1 : $gears\_pos = g\_retracted$
grd_FUN−2−doors: $doors\_pos = open$
THEN
FUN−G−1: $gears\_pos := g\_extended$
END
Event retract_gears
REFINES
retract_gears
WHEN
grd1 : $gears\_pos = g\_extended$
grd_FUN−2−doors: $doors\_pos = open$
THEN
FUN−G−1: $gears\_pos := g\_retracted$
END
Event reverse
REFINES
reverse
THEN
...
END
Event open_doors
WHEN
grd1 : $doors\_pos = closed$
THEN
FUN−G−2: $doors\_pos := open$
END
Event close_doors
...

END

Figure 8. *Machine* `Gears_M1`

Approches de Design Rationale

Cadre de Référence

Salim Fathy[1], Elena Kornyshova[2]

1. CEDRIC, Conservatoire National des Arts et Métiers
2, rue Conté, 75003, Paris, France
salim.fathy.90@gmail.com

2. CEDRIC, Conservatoire National des Arts et Métiers
2, rue Conté, 75003, Paris, France
elena.kornyshova@cnam.fr

RESUME. Le rôle des systèmes d'information s'élargissant de plus en plus, ils doivent être conçus d'une manière organisée, suivant une logique bien réfléchie avec une argumentation des décisions prises claire et justifiable. Le Design Rationale (DR) est une discipline qui répond à cette exigence car elle comprend la conception et la modélisation des raisons pour lesquelles des décisions de conception des systèmes d'information ont été prises. Plusieurs travaux sont menés autour de Design Rationale. Cependant, il n'existe pas d'état de l'art permettant d'avoir un aperçu des approches de DR, de les comparer et de guider la sélection des approches dans des cas concrets. Le but de ce papier est de présenter un cadre de référence qui atteint ces objectifs et d'appliquer ce cadre sur quelques approches de DR les plus connues. Le cadre de référence proposé comporte quatre vues et treize critères organisés dans ces vues.

ABSTRACT. The role of Information Systems is growing constantly. They should be elaborated in an organized way following a well-defined logic using a clear and justifiable argumentation of the decision made. Design Rationale (DR) is a discipline responding to this requirement as it includes the design and modeling of the reasons justifying the decisions made during the design of information systems. Many works are done in the Design Rationale domain. However, a structured state-of-the-art allowing to have an overview of the DR approaches, to compare them and to guide the selection of the DR approaches in concrete situations does not exist. The goal of this paper is to present a comparison framework meeting these goals and to apply this framework to several most well-known DR approaches. A comparison framework contains four views and thirteen criteria organized into these views.

MOTS-CLES : Design Rationale, Cadre de référence, Etat de l'art

KEYWORDS: Design Rationale, Comparison framework, State-of-the-Art

1. Introduction

A l'ère de l'informatique, les systèmes d'information deviennent de plus en plus demandés par les entreprises, les administrations et d'autres institutions. Le besoin ne cesse d'augmenter et les clients deviennent de plus en plus exigeants vis à vis de ces systèmes. Pour répondre à ces demandes, les systèmes doivent être conçus d'une manière organisée, suivant une logique bien réfléchie avec une argumentation des décisions prises qui soit claire et justifiable. Pour aboutir à ces résultats, les concepteurs doivent exploiter les expériences précédentes pour décider, faire leurs choix et apporter des améliorations à ce qui a été déjà réalisé.

Pour ce faire, il faut tracer dans chaque projet la logique suivie lors de la phase de conception. Il faut conserver les questions posées, les discussions échangées entre concepteurs "Designers" et les propositions de chaque membre avec les arguments qui défendent chaque solution (proposition). Ce traçage est nommé le Design Rationale (DR). Le Design Rationale est une discipline qui comprend la conception et la modélisation des raisons pour lesquelles des décisions de conception des systèmes ont été prises. Le but de DR est d'avoir un historique réutilisable de ces décisions et raisonnements ainsi que les arguments/interprétations qui vont avec, de la façon la plus réaliste et la moins coûteuse. En effet, les concepteurs devraient conserver toute réflexion ou action qui mène à un résultat concluant lors de la conception des systèmes. Les discussions entres concepteurs, les prises de notes lors des réunions, les modélisations fonctionnelles et techniques, les choix effectués, les décisions prises sont tous des éléments importants qui alimentent le Design Rationale.

Concernant la discipline de DR, on trouve dans la littérature bon nombre d'approches et/ou méthodes qui proposent plusieurs façons d'implémentation du Design Rationale. Chaque approche a ses caractéristiques, ses particularités et ses outils. Ces approches ont aussi plusieurs points en commun, notamment l'objectif global qui est d'assurer la traçabilité des décisions prises. Malgré la présence de certains travaux de recherche présentant l'état de l'art (Regli et al., 2000) (Tang, 2007) (Tang et al., 2006), nous n'avons pas trouvé d'état de l'art structuré du domaine de DR dans la littérature. En effet, le travail de (Regli et al., 2000) décrit la comparaison des outils de DR par rapport aux méthodes de capture, de représentation et d'extraction, à l'approche technique (orientée processus ou bien orientée paramètres) et au domaine de conception. Les auteurs se concentrent donc sur les outils et omettent les méthodes de DR non outillées. De plus, la liste des critères de comparaison est limitée dans leur proposition. A. Tang (Tang, 2007) décrit huit groupes d'approches de DR, mais n'effectue pas de comparaison entre ces approches. (Tang et al., 2006) décrit les résultats d'une étude de la perception de la valeur de DR et des façons dont les professionnels utilisent et documentent leurs décisions de conception.

L'objectif de ce papier est de proposer un cadre de référence permettant de comparer les approches Design Rationale existant dans la littérature et d'appliquer ce cadre aux approches les plus utilisées et connues chez les concepteurs et la communauté de chercheurs de DR. Ceci pour répondre aux questions suivantes :

– Quels sont les critères de comparaison des différentes approches de DR ?

– Comment une approche peut-elle être décrite à l'aide du cadre de référence ?

Les résultats de ce papier pourraient dans le futur : être complétés pour prendre en compte un plus grand nombre d'approches de DR ; aider à proposer un guide de sélection des approches de DR et générer des idées pour créer de nouvelles approches de DR.

La section suivante décrit la notion du Design Rationale, présente les principales approches de DR et explique la sélection d'une sous-partie de ces approches pour être comparées lors de notre étude. La section 3 décrit le cadre de comparaison entre les approches Design Rationale. Dans la section 4, nous analysons les approches DR choisies par rapport au cadre de comparaison. Nous terminons dans la section 5 par une présentation des conclusions et des questions ouvertes ainsi que des perspectives qui vont élargir notre recherche.

2. Le Design Rationale, définitions et sélection des approches à comparer

L'origine du Design Rationale est assignée à Kunz et Horst Rittel à travers leurs travaux et au développement de l'approche IBIS (Issue-Based Information System) en 1970. Cette approche a été exploitée par la suite par plusieurs chercheurs. La première version présentée à la communauté Design Rationale après l'IBIS est l'approche PHI (Procedural Hierarchy of Issues) décrite par Ray McCall lors de sa thèse. Ensuite, le langage de représentation des décisions DRL (Decision Representation Language) orienté vers l'ingénierie des logiciels a été proposé par Potts et Bruns. Puis le QOC (Questions, Options and Criteria) comme méthode d'analyse et de choix entre plusieurs alternatives (Lewkowicz et Zacklad, 1998).

Ces approches sont la base principale du Design Rationale et c'est à partir d'elles que d'autres chercheurs ont pu développer et présenter leurs propres approches. Ces dernières sont très nombreuses dans la littérature de DR. Chaque chercheur ou groupe de chercheurs, et ce depuis la naissance de DR, a apporté sa vision et sa perception de DR ce qui a permis d'enrichir les connaissances dans ce domaine.

2.1 Définitions du Design Rationale

Dans la littérature, on trouve plusieurs définitions du Design Rationale. Nous présentons ci-dessous quelques versions de ces définitions qui permettent de mieux comprendre ce concept.

– Le Design Rationale est la conception et la modélisation des raisons pour lesquelles des décisions de conception des systèmes ont été prises. L'objectif est d'avoir un historique réutilisable de ces décisions et raisonnements ainsi que les arguments/interprétations qui vont avec, de façon la plus réaliste et la moins coûteuse (Lewkowicz et Zacklad, 1998).

– Le Design Rationale est le raisonnement et l'argumentation qui conduit à la décision finale de la façon dont la réflexion de conception est réalisée. Cette

réflexion de conception est l'effet attendu ou le comportement que le concepteur vise pour remplir la fonction souhaitée (Sim et Duffy, 1994).

– Le Design Rationale est composée des informations qui expliquent pourquoi un artefact est structuré d'une certaine façon et a un certain comportement (Conklin et Burgess-Yakemovic, 1995).

Pour résumer, le DR peut être défini comme une activité qui consiste à modéliser les questions, les propositions, les arguments et les décisions prises lors de la conception des systèmes. Ce processus de décisions est souvent réutilisé lors de la modélisation de nouveaux systèmes; d'où son utilité.

2.2 Approches du Design Rationale

La littérature propose plusieurs approches, méthodes et langages de DR. Chacun possède ses propres caractéristiques et propriétés, mais souvent ces approches partagent de nombreux points en commun.

Parmi les approches existantes, les plus connues sont :

– IBIS (Issue-Based Information System) (Ebadi et al., 2009) avec son outil gIBIS (Graphical IBIS) (Conklin et Begeman, 1988) et les variantes d'IBIS : PHI (Procedural Hierarchy of Issues) (Fischer et al., 1989), REMAP (Representation and MAintenance of Process knowledge) (Ramesh et Luqi, 1995) ;

– DRL (Decision Representation Language) (Lee, 1990) ;

– QOC (Questions Options and Criteria) (Lewkowicz et Zacklad, 1998) et ses variantes : ABRICO (Accord, But, pRoposition, Interprétation en Conception) (Lewkowicz et Zacklad, 1998), QUIMERA (Métamodèle de qualité pour l'amélioration du Desin Rationale) (García Frey et al., 2011), TEAM (Traceability, Exploration and Analysis Model) avec son outil DREAM (Design Rationale Environment for Argumentation and Modelling) (Lacaze et Palanque, 2007), une extension de QOC par les critères et les facteurs ergonomiques (Farenc et Palanque, 1999).

Il existe d'autres approches DR qui sont moins connues : FBS (Fonction, Behaviour, Structures) (Kannengiesser et Gero, 2010), REMIS (Rationale-driven Evolution and Management Information System) (Ocampo et Münch, 2007), Kuaba (Approche de mise en relation avec ingénierie d'exigences), DIPA (Données, Interprétations, Propositions, Accord) (Roy, 2012). Le Design Rationale peut également être exprimé en utilisant les modèles de variabilité (feature models) (Schlee et Vanderdonckt, 2004), les méthodes d'évaluation (Farenc et al., 1995) ou encore les méthodes issues du domaine décisionnel telles que les arbres de décision (Kast, 2002) ou la méthode AHP – Analytic Hierarchy Process (Saaty, 1980).

2.3 Sélection des approches DR à comparer

Plusieurs sources citent les approches IBIS, DRL et QOC (Lewkowicz et Zacklad, 1998) (Farenc et Palanque, 1999) (Lacaze et Planque, 2007) (Tang, 2007). Pour comparer les approches DR dans un premier temps et tester l'applicabilité de notre

cadre de référence, nous avons sélectionné ces trois approches et quelques approches dérivées : PHI et REMAP pour IBIS et ABRICO et QUIMERA pour QOC. En effet, les approches d'origine (comme IBIS et QOC) proposent souvent des fonctionnalités limitées, il est donc judicieux de compléter l'analyse par des variantes plus développées des approches de base. Ensuite, nous avons ajouté à notre étude l'approche FBS car cette dernière est orientée fonction, c'est-à-dire elle part des fonctionnalités prévues par le système pour élaborer les décisions de conception.

3. Présentation du cadre de référence

Afin de comparer les approches de DR nous utilisons un cadre de référence inspiré des travaux de (Jarke et al., 1992). Ce cadre de référence comportant quatre vues (ou quatre mondes) : Sujet, System, Utilisation et Développement a été utilisé dans plusieurs domaines : l'ingénierie des systèmes d'information (Jarke et al., 1992), l'ingénierie des exigences (Jarke et al., 1993), l'ingénierie des processus de développement des systèmes d'information (Rolland, 1998), l'ingénierie des méthodes (Rolland, 1997) (Kornyshova, 2011). Le cadre de référence à quatre mondes a démontré son efficacité pour la compréhension de différentes disciplines (Rolland, 1998).

Dans le cadre de référence initial conçu pour le domaine de l'ingénierie des SI (Jarke et al., 1992), les quatre vues sont décrites comme suit :

– ***La vue Sujet*** est consacré aux connaissances clés du domaine pour lequel le SI donné est conçu. Il représente les objets du monde réel qui feront l'objet de la modélisation du système.

– ***La vue Système*** comprend les spécifications du système à différents niveaux de détail. Il comprend les modèles des entités, des événements, des processus etc., ainsi que les liens entre les spécifications de conception et leur implémentation.

– ***La vue Usage*** décrit l'environnement organisationnel du SI, c'est-à-dire, les objectifs et les activités des agents, ainsi que comment le système est utilisé pour atteindre les objectifs, y compris ceux des parties prenantes telles que les propriétaires et les utilisateurs.

– ***La vue Développement*** est orienté sur les entités et les activités qui correspondent au processus même de l'ingénierie.

Dans le contexte des approches de DR, nous définissons les vues (mondes) de la façon suivante. La vue Sujet est celui des décisions prises lors de la conception des SI. La vue Système détaille les différentes facettes de représentation utilisées dans les approches de DR : formalisation, granularité, etc. La vue Usage nous permet d'étudier les utilisations de DR selon les différentes approches. Et, enfin, la vue Développement détaille les différents processus permettant de construire et de tracer les décisions prises.

Nous considérons que cette séparation en quatre vues nous permettra de mieux comprendre les différentes approches de DR et de faciliter leur comparaison. Notre

version des critères de comparaison qui composent les quatre vues est détaillée dans les sous-sections suivantes.

3.1 Vue Sujet

La vue Sujet correspond aux décisions de conception (appelées en anglais « design decisions ») étudiées dans les approches de Design Rationale. Il importe donc d'étudier à ce niveau les différents composants des décisions de conception (les attributs de décision). Nous appelons le critère correspondant 'Notation'.

Critère 1 : Notation. Venant du domaine d'aide à la décision la définition d'une décision comporte les éléments suivants : problématique (sélection, tri, classement) (Roy, 2005) ; alternatives, critères, évaluations des alternatives par rapport aux critères (Roy, 1996) (Vincke, 1995). Cependant, dans le domaine de DR qui fait partie du celui de conception des SI, ces notions sont exprimées souvent différemment. Par exemple, on retrouve les termes 'Positions' (IBIS (Ebadi et al., 2009), PHI (Fischer et al., 1989)) et 'Options' (QOC (Lewkowicz et Zacklad, 1998), QUIMERA (García Frey et al., 2011)) pour désigner les alternatives, ou encore le terme 'Arguments' pour un équivalent des critères (IBIS (Ebadi et al., 2009), PHI (Fischer et al., 1989)). Nous définissons par conséquent un ensemble d'attributs de décisions de conception pour chaque approche de DR pour comparer le contenu principal de ces approches.

3.2 Vue Système

Si la vue Sujet est focalisée sur les décisions de conception, la vue Système détaille la façon dont ses décisions sont représentées dans les différentes approches de DR. Cette vue inclut quatre critères : degré de formalisme, granularité, type statique ou dynamique et ouverture.

Critère 2 : Degré de formalisme. Ce critère montre le niveau de formalisation des décisions dans une approche DR (Kannengiesser et Gero, 2010). En effet, en fonction de l'approche, la décision peut être formalisée de façon plus ou moins complète. Les valeurs possibles de ce critère sont : formelle pour les approches décrites et documentées complètement, semi formelle pour les approches qui sont peu documentées et non formelle pour les approches non documentées.

Critère 3 : Granularité (Rolland, 1998)**.** Ce critère définit le niveau de détail de l'argumentation des décisions préconisé par une approche DR. Le niveau peut être très fin (très détaillé), moyennement détaillé ou peu détaillé. Le critère de granularité est important dans la comparaison des approches DR car l'utilisateur pourra sélectionner celle qui correspondra mieux aux besoins du projet.

Critère 4 : Type (Statique/Dynamique) (Rolland, 1998). Il s'agit d'un critère pour affirmer la présence ou l'absence d'échanges entre les acteurs et les parties prenantes de DR. Dans notre cas il s'agit de mettre en évidence le niveau d'échanges existant dans chaque approche. Ce critère peut prendre comme valeur : statique s'il n'y a pas

d'échanges entre les acteurs (designers, client, AMOA,..) ou bien dynamique lorsque les échanges et le partage d'informations entre ces acteurs existent.

Critère 5 : Ouverture. L'ouverture d'une approche est la prise en compte de plusieurs points de vue utilisateurs dans la construction de DR (Atwood et Horner, 2007). Pour répondre à ce critère, il suffit d'affirmer ou de nier la présence des acteurs et parties prenantes qui peuvent participer à la construction du Design Rationale à travers une approche définie. Ce critère permet aux designers de distinguer les approches qui prennent en compte les différents acteurs lors de la construction du Design Rationale, de ceux qui n'en tiennent pas compte. Ce critère est complémentaire au critère précédent.

3.3 Vue Usage

Cette vue permet de décrire les situations types pour lesquelles les approches DR ont été développées. La vue Usage comporte trois critères : domaine d'utilisation, orientation et évolutivité. L'analyse des critères correspondants permettra de compléter les vues Sujet et Système par les éléments du contexte d'application des approches DR.

Critère 6 : Domaine d'utilisation (Rolland, 1998). Le domaine d'utilisation définit les situations dans lesquelles l'application d'une approche DR est efficace. Un domaine d'utilisation peut être, par exemple : les systèmes complexes à concevoir, les systèmes à fortes exigences, les systèmes avec une fréquence de changements élevée, etc.

Critère 7 : Orientation. Ce critère indique si l'approche est orientée processus ou fonction. Cette distinction a été mentionnée dans l'article (Regli et al., 2000). En effet, une approche orientée processus est une approche qui définit des étapes successives et ordonnancées. Par conséquent, ce sont ces étapes (le processus) qui définissent le but final. Alors que dans une approche orientée fonction, c'est la fonction (ou but final) qui oriente les étapes à réaliser pour construire le Design Rationale. Autrement dit, quel que soit l'ordonnancement des étapes à suivre, l'objectif est d'atteindre la fonction finale. Ce critère permet de bien définir l'orientation des approches afin d'aider les designers à choisir l'approche qui correspond à leur méthode de travail et à l'orientation du système à concevoir.

Critère 8 : Evolutivité. L'évolutivité d'une approche concerne la prise en compte des changements et des évolutions des besoins par une approche DR. Les évolutions peuvent être déclenchées par un changement dans le cahier de charge ou par l'apparition d'une nouvelle exigence (Santos et Pereira de Medeiros, 2011). De cette façon, une approche peut être adaptée ou bien inadaptée à l'utilisation dans un contexte changeant. Pour répondre à ce critère, nous détaillons comment les changements sont pris en compte dans une approche DR, par exemple : par changement des fonctions, en modifiant les hypothèses, en ajoutant des problèmes.

3.4 Vue Développement

La vue Développement ajoute un nouveau point de vue sur la comparaison des approches DR, à savoir les différentes étapes du processus sous-jacent à l'application des approches DR : capture d'informations, élaboration de la décision, enregistrement des informations, ainsi que la publication des résultats et le temps nécessaire à l'élaboration de la décision. Ces cinq critères détaillent les aspects pratiques d'application des approches DR en complément de l'analyse des éléments statiques et orientés contexte présents dans les trois vues précédentes.

Critère 9 : Capture d'informations. C'est le mécanisme avec lequel on capture l'information à utiliser et à exploiter dans une approche DR (Regli et al., 2000). Les informations de DR sont souvent capturées lors des réunions et des discussions entre les acteurs. Ceci se fait en répondant aux questions et en justifiant les propositions de chaque membre. La capture d'information peut être soit orale soit par écrit.

Critère 10 : Elaboration de la décision. C'est un enchainement d'étapes suivies pour élaborer une décision de conception. Dans notre cas, les valeurs de ce critère sont des tâches à effectuer pour construire le DR. Exemple: Définir les problèmes, Exprimer les opinions, Présenter les arguments, Définir les questions, Présenter les alternatives, Proposer les objectifs, etc.

Critère 11 : Enregistrement d'informations. C'est le mécanisme d'enregistrement et d'historisation d'informations dans une approche DR (Regli et al., 2000). Une fois les informations initiales sont exploitées et traitées et les décisions sont prises, elles doivent être enregistrées. Les différentes méthodes d'enregistrement peuvent être : dans des rapports écrits, dans les bases de données informatiques ou encore dans les schémas de développement. Ce critère permet de savoir où l'on stocke les informations et les résultats atteints pour des utilisations futures.

Critère 12 : Publication/Outillage. Le critère de publication/affichage des résultats définit le système dans lequel sont affichés les résultats atteints au cours de l'exécution d'un processus (Rolland, 1998). Dans notre étude comparative les résultats sont les décisions prises lors de l'application d'une approche Design Rationale au cours de la conception d'un système. Les décisions sont publiées souvent soit dans des rapports écrits, soit dans des outils associés aux approches.

Critère 13 : Temps de prise de décision (Kannengiesser et Gero, 2010). Le temps de prise de décision permet de déterminer le temps nécessaire que l'on met avec une approche afin de prendre une décision. Les valeurs de ce critère décrivent de manière générale le temps mis dans une approche DR : long, moyen ou court. Ce critère peut être utile lors de la conception des systèmes avec des fortes exigences en termes de délai.

4. Application du cadre de référence aux approches de DR

4.1 Analyse de la Vue Sujet

Critère 1 : Notation. Ce critère permet de décrire les concepts utilisés dans une approche de DR. Nous avons identifié les attributs suivants pour chaque approche :

– IBIS : Issue (problème) ; Positions ; Arguments.

– PHI : Issue (problème) ; Positions ; Arguments ; Positions ; Arguments.

– REMAP : Issue (problème), Position, Argument ; Exigence, décision contrainte et hypothèse.

– DRL : Question ; Alternative ; Objectif.

– QOC : Question ; Options ; Critères.

– ABRICO : But ; Interprétations ; Propositions ; Accord.

– QUIMERA : Question ; Options ; Critères : Formules de calcul.

– FBS : Fonction ; Comportements ; Structure.

La structure des approches n'est pas figée, elle peut changer selon la situation de conception, mais l'ensemble des concepts clés doit toujours être présent dans l'implémentation de chaque approche.

4.2 Analyse de la Vue Système

Critère 2 : Degré de formalisme. Les approches sont réparties par degré de formalisme en deux catégories :

– Les approches formelles : QOC, ABRICO, QUIMERA, DRL, REMAP.

– Les approches semi-formelles : FBS, IBIS, PHI.

Les approches DR sélectionnées sont toujours accompagnées de documentation qui contient au moins l'historique des décisions prises.

Critère 3 : Granularité. Le niveau de détails dans les arguments diffère d'une approche à une autre. Le résultat de cette comparaison est le suivant :

– IBIS : Argumentation très fin. Le modèle peut s'affiner autant que souhaité.

– PHI : Argumentation très fin. Même les questions peuvent être détaillées.

– REMAP : Peu détaillé. Ce sont les exigences qui orientent les arguments.

– DRL : Argumentation très détaillée.

– QOC : Argumentation avec un niveau fin. On peut atteindre le niveau de détail désiré.

– ABRICO : Le niveau d'argumentation est variable. La granularité dépend des acteurs.

– QUIMERA : Peu d'arguments. Basé sur le calcul.

– FBS : Argumentation très détaillée.

Il est à noter que le niveau d'argumentation peut différer dans une même approche. L'interactivité des concepteurs et des différentes parties prenantes jouent un rôle important dans le niveau d'argumentation.

Critère 4 : Type (Statique/Dynamique). Pour le critère de dynamisme des approches, la plupart des approches sont dynamiques et permettent un échange entre les acteurs DR. Les approches statiques sont: QOC, QUIMERA et une variante du modèle d'ABRICO (donc les approches de la famille QOC).

Critère 5 : Ouverture. Les approches ouvertes (tenant compte des points de vue différents en plus de celui des concepteurs) sont : IBIS, PHI, REMAP, DRL et ABRICO. Les approches QOC, FBS et QUIMERA ne sont pas ouvertes aux acteurs externes. Elles permettent juste de choisir entre plusieurs possibilités (QOC et QUIMERA) ou bien de compléter un DR déjà existant (FBS).

4.3 Analyse de la Vue Usage

Critère 6 : Domaine d'utilisation. Les domaines d'utilisation diffèrent selon les approches DR. Par rapport à ce qui est défini dans chaque approche, nous résumons ces situations comme suit :

– IBIS : Pour les systèmes complexes, qui nécessitent plusieurs échanges et débats avant chaque décision.

– PHI : Pour les systèmes complexes, avec des constructions d'artefacts.

– REMAP : Pour des grands projets.

– DRL : Pour des projets en équipe avec beaucoup de détails fonctionnels.

– QOC : Pour les situations de conceptions où les concepteurs sont amenés à faire des choix entre plusieurs options.

– ABRICO : Pour les systèmes complexes et pour les systèmes avec plusieurs parties prenantes.

– QUIMERA : Pour les situations où l'on désire mesurer et évaluer les critères de choix dans les décisions ; également, lorsque la qualité est exigée.

– FBS : Dans des systèmes orientés fonction ; nécessité d'adaptation.

Critère 7 : Orientation. Les approches orientées processus sont : QOC, ABRICO, QUIMERA, IBIS, PHI, DRL et REMAP. L'approche FBS est la seule approche orientée fonction parmi les approches étudiées.

Critère 8 : Evolutivité. Un des critères les plus importants dans le choix de l'approche à implémenter pour construire un DR est la capacité d'évolution et d'implémentation des changements dans le cahier des charges des systèmes. Pour suivre ces changements, les approches prennent en comptes ces modifications de différentes façons :

– IBIS : En ajoutant des Issues (problèmes), Positions (réponses) et les Arguments.

– PHI : Même chose que l'approche IBIS.

– REMAP : En modifiant les contraintes et les hypothèses.

– DRL : En ajoutant des Questions, Alternatives, Objectifs.

– QOC : En ajoutant des nouvelles questions, options et critères

– ABRICO : Par ajout d'acteurs dans le diagramme dynamiques.

– QUIMERA : L'impact est sur les formules de calculs.

– FBS : Par changement des fonctions.

Toutes les approches permettent d'implémenter les éventuelles évolutions que peuvent subir les systèmes au cours de leur vie. Ceci est une bonne chose, par contre aucune approche n'a prévu d'implémenter un module de prédiction qui peut signaler les possibles chemins d'évolution d'un système dès la phase de conception.

4.4 Analyse de la Vue Développement

Critère 9 : Capture d'informations. Pour appliquer une approche DR, la première étape consiste à capturer les informations. Il existe plusieurs façons de faire :

– IBIS : Soit à travers l'édition dans un outil, soit en répondant aux questions (Issues) oralement et/ou par écrit.

– PHI : A travers l'outil Janus.

– REMAP : En éditant dans l'outil.

– DRL : En répondant aux questions par des alternatives. Les réponses sont éditées soit dans le schéma DRL ou bien dans l'outil SYBIL.

– QOC : En répondant aux questions, en groupe ou individuellement.

– ABRICO : A travers l'échange et la communication verbale et/ou écrite.

– QUIMERA : En se posant des questions, en énumérant les options possibles et en valorisant les critères.

– FBS : A partir de l'historique des DR existants.

La capture d'information est la source de DR. Pour cette raison, elle doit être organisée, claire et permettre d'identifier les bons éléments qui serviront à développer la solution et pousser l'analyse.

Critère 10 : Elaboration de la décision. L'enchainement d'étapes pour construire un DR diffère aussi d'une approche à l'autre. Les étapes essentielles pour chaque approche se résument comme suit :

– IBIS : 1- Définir les problèmes (issues) 2-Proposer les solutions (positions), 3-Présenter les arguments.

– PHI : Même processus que IBIS avec plusieurs itérations.

– REMAP : 1- Etablir le processus d'IBIS, 2- Alimenter par les exigences, 3-Prendre les décisions, 4- Déterminer les contraintes et hypothèses liées aux décisions prises.

– DRL : 1- Définir les Questions, 2- Présenter les alternatives, 3- proposer les objectifs.

– QOC : 1-Définir les questions, 2- Proposer plusieurs options, 3- Lister les critères de choix.

– ABRICO : 1- Définir le But, 2- Attendre l'interprétation du But, 3- Recevoir et émettre des propositions.

– QUIMERA : 1- Définir les questions, 2- Proposer plusieurs options, 3- Etablir les formule de calcul de la qualité.

– FBS : 1- Identifier la Fonction à construire, 2- Définir son comportement future, 3- Définir son point final et sa structure.

L'enchainement de ces étapes n'est pas obligatoire, mais il permet de mieux s'organiser et de prendre une décision de conception selon les bonnes pratiques définies par les créateurs de ces approches.

Critère 11 : Enregistrement d'informations. Quand les informations sont capturées, traitées, analysées puis les décisions sont prises, il faut les enregistrer pour garder la trace de ces décisions. Les approches de DR procèdent de différentes manières :

– IBIS : Dans la base de données de l'un des outils ou dans des rapports écrits.

– PHI : Dans la base de données de l'outil Janus.

– REMAP : Dans la base de données de l'outil.

– DRL : Dans des schémas écrits ou dans la base de données de l'outil.

– QOC : Prise de décision en choisissant les solutions proposées dans un schéma formel QOS ; par rédaction.

– ABRICO : Les accords (décisions) sont enregistrés dans le schéma choisi pour le modèle ; par rédaction.

– QUIMERA : Par un schéma formel avec des formules de choix des critères.

– FBS : En modifiant par écrit le modèle exploité ; par rédaction.

Les deux formats qui sont souvent utilisés pour enregistrer les informations de DR sont soit par écrit soit par insertion dans la base de données. Les approches informatisées sont celles qui possèdent des bases de données de stockage.

Critère 12 : Publication/Outillage. Quand les décisions sont prises, il faut les afficher et publier sous différentes formes :

– Par des rapports écrits : QOC, ABRICO, FBS, QUIMERA, IBIS, PHI, REMAP.

– Par des outils informatique : IBIS, PHI, REMAP, DRL.

– Par le code informatique : DRL.

Dans la plupart des travaux les chercheurs mettent l'accent sur la clarté des résultats à afficher indépendamment de leurs formes. Ceci fait partie des bonnes pratiques de développement du Design Rationale.

Critère 13 : Temps de prise de décision. Le temps de prise de décision à travers les approches est un facteur important dans la sélection d'une approche par les designers afin d'implémenter un DR. Les approches qui ont un temps de réponse

court sont : QOC, FBS et QUIMERA. Les autres approches sont tous longues dans le temps de prise de décision. En effet, le facteur qui ralenti le temps d'exécution de l'approche et par suite le temps de prise de décision est la fréquence d'échange et de discussion entre les acteurs de DR. Puisque le QOC et QUIMERA sont statiques, alors il est logique d'avoir le temps de réponse court. Lors de l'implémentation de FBS, on ne démarre pas du zéro ; il s'agit d'exploiter l'existant pour élaborer des nouvelles décisions. Cela constitue la principale raison pour laquelle le temps de réponse est court.

Les résultats de l'application du cadre de référence aux approches de DR sélectionnées sont résumés dans l'Annexe.

5. Conclusions, questions ouvertes et perspectives de recherche

Aujourd'hui, le Design Rationale est devenue une priorité de la communauté des concepteurs, surtout avec l'évolution des systèmes et leurs niveaux de complexité qui ne cessent d'augmenter. Pour avoir une meilleure maitrise de la conception et du développement des systèmes, le traçage des décisions prises et des réflexions effectuées lors des projets demeure très utile à moyen et à long terme.

D'après l'analyse comparative des principales approches DR, nous avons pu tirer les conclusions suivantes :

– Les approches historiques, les plus utilisées et exploitées dans la recherche, sont : IBIS, PHI et DRL ; à un moindre niveau QOC, ABRICO et REMAP.

– Toutes les approches permettent l'échange entre différents acteurs à part QOC, QUIMERA et une variante d'ABRICO.

– Toutes les approches partagent les étapes de capture d'informations, d'échange d'informations, d'enregistrement de décisions prises et de discussions échangées pour des futures exploitations.

– L'absence de schéma intégral et commun entre toutes les approches de DR.

– Le choix d'une approche doit se faire selon le cadre de référence et les contraintes de chaque système à concevoir. Dans ce point, on pourrait noter l'absence d'un référentiel qui guiderait et orienterait les développeurs dans le choix des approches DR.

– Les outils informatique de DR facilitent l'usage des approches mais ne sont pas suffisants sans un support méthodologique.

– Les outils informatiques de DR présents sur le marché sont tous liés à une approche spécifique ce qui confirme un manque de solution intégrale.

– Les approches de DR doivent être capables de prendre en compte les évolutions et les changements que connaissent les systèmes. Or, ce n'est pas le cas de toutes les approches DR existantes dans la littérature.

– Il manque une étape de fouille dans les bases de données de DR. En effet, il n'y a pas d'algorithmes ou de méthodes qui permettent d'interroger rapidement le DR.

– Absence de module de prédiction des éventuelles évolutions pour les systèmes à concevoir.

En nous basant sur ces conclusions, nous avons défini les perspectives de recherche suivantes. Dans un premier temps, nous allons approfondir la comparaison des approches de DR afin d'inclure un plus grand nombre d'approches et de constituer un état de l'art complet de ce domaine, ainsi que de compléter le cadre de référence par un guide de sélection des approches. Dans un deuxième temps, nous allons étudier la possibilité de proposer un modèle décisionnel de DR qui prendrait en compte un plus large ensemble de concepts utilisés dans ce domaine ; développer les aspects de fouille de données appliquée à DR ; améliorer l'évaluation des décisions prises dans le cadre de DR et analyser les impacts potentiels entre le domaine de DR et le domaine de BI (Business Intelligence).

Bibliographie

Atwood M.E. et Horner J., Redesigning the Rationale for Design Rationale, *Human-Computer Interaction*, pp 11-19, 2007.

Conklin E. J. et Begeman M.L., GIBIS: A Hypertext Tool for Exploratory Policy Discussion, in *ACM Transactions on Office Information Systems* 6(4), pp. 303–331, 1988.

Conklin E. J. et Burgess Yakemovic K. C., A Process-Oriented Approach to Design Rationale, in *Design Rationale Concepts, Techniques, and Use*, T. Moran and J. Carroll, eds., Lawrence Erlbaum Associates, Mahwah, NJ, pp. 293-428, 1995.

Ebadi T., Purvis M. et Purvis M., A collaborative web-based Issue Based Information System (IBIS) framework, *Information Science Discussion Papers Series*, 2009.

Farenc Ch. et Palanque Ph., Exploitation des notations de Design Rationale pour une conception justifiée des applications interactives, In *11ème conférence francophone Interaction Homme-Machine*, Montpellier, France, 1999.

Farenc Ch., Palanque Ph. et Vanderdonckt J., User Interface Evaluation: is it Ever Usable? *In Proceedings of 6th International Conference on Human-Computer Interaction HCI International'95*, Amsterdam, Netherlands, pp. 329-334, 1995.

Fischer G., Mccall R. et Morch A., JANUS: Integrating Hypertext with a Knowledge-based Design Environment, In *Proceeding of the second annual ACM conference on Hypertext*, 1989.

García Frey A., Céret E., Dupuy-Chessa S. et Calvary G., QUIMERA: A Quality Metamodel to Improve Design Rationale, In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, 2011.

Kast R., *La théorie de la decision*, Nouv. éd. Paris : Découverte, 2002.

Kannengiesser U. et Gero G. S., A Framework for Constructive Design Rationale, In *Design Computing and Cognition* 10, 2010.

Lacaze X. et Palanque Ph., DREAM & TEAM: A Tool and a Notation Supporting Exploration of Options and Traceability of Choices for Safety Critical Interactive Systems, In *Proceedings of INTERACT 2007*, Part II, pp. 525 – 540, 2007.

Lee J., SIBYL: A Tool for Managing Group Decision Rationale, In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, 1990.

Lewkowicz M. et Zacklad M., La capitalisation des connaissances tacites de conception à partir des traces des processus de prise de décision collective, *In Proceedings of Ingénierie des Connaissances 1998*, pp. 177-188, 1998.

Ocampo A. et Münch J., The REMIS Approach for Rationale-Driven Process Model Evolution, *Software Process Dynamics and Agility*, 2007.

Ramesh B. et Luqi, An Intelligent Assistant for Requirements Validation, *Journal of Systems Integration*, Volume 5, Issue 2, 1995.

Regli W. C., Hu X., Atwood M. et Sun W., A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval, *Engineering with Computers*, pp. 209-235, 2000.

Rolland C., A Primer for Method Engineering, in Proceedings of the Conference INFORSID'1997, Toulouse, France, 1997.

Rolland R., A Comprehensive View of Process Engineering, In *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*, 1998.

Roy B., *Multicriteria Methodology for Decision Aiding*, Dordrecht, Kluwer Academic Publishers, 1996.

Roy, B., Paradigms and challenges, Book chapter, *In Multiple Criteria Decision Analysis - State of the Art Survey*, Springer. editor(s) J. Figueira, S. Greco, M. Ehrgott, pp. 3-24, 2005.

Roy R., Industrial Knowledge Management: A Micro-level Approach, *Springer Science & Business Media*, 2012.

Saaty T.L., *The Analytic Hierarchy Process*, NY, McGraw Hill, 1980.

Santos E. G. et Pereira de Medeiros A., Design Rationale Representation in Requirements Engineering using the KAOS meta-model, *In WER*, 2011.

Schlee M. et Vanderdonckt J., Generative Programming of Graphical User Interfaces, *In proceedings of the ACM Conference on Visual Interfaces AVI'2004*, Gallipoli, Italy, 2004.

Sim S. et Duffy, A., A New Perspective to Design Intent and design Rationale, in *Artificial Intelligence in Design Workshop Notes for Representing and Using Design Rationale*, 15-18 August, pp. 4-12, 1994.

Tang A., Ali Babar M., Gorton I. et Han J., A survey of architecture design rationale, *Journal of Systems and Software*, Volume 79, Issue 12, pp. 1792-1804, December, 2006.

Tang A., *A Rationale-based Model for Architecture Design Reasoning*, PhD thesis, Swinburne University of Technology, 2007.

Vincke Ph., A short note on a methodology for choosing a decision-aid method, *Advances in Multicriteria Analysis*, Kluwer Academic Publishers, Netherlands, pp. 3-7, 1995.

Annexe. Tableau récapitulatif de l'application du cadre de référence aux approches de DR.

Approche	Axe Sujet		Axe Système			Axe Usage		Axe Développement				
	Notation	Degré de formalisme	Granularité	Type (Statique/ Dynamique)	Ouverture	Domaine d'utilisation	Orientation	Evolutivité	Capture d'informations	Elaboration de la décision	Enregistrement d'informations	Publication / Outillage	Temps de prise de décision
IBIS	Issue(problème) ==>Positions ==> Arguments.	Semi Formelle. Composé de questions, réponses et arguments qui peuvent être représenté selon le contexte.	Très fin. Le modèle peut s'affiner autant que voulu.	Dynamique.	OUI	Pour les systèmes complexes, qui nécessitent plusieurs échanges et débats avant chaque décision.	Orienté Processus.	En ajoutant des Issue(problème), Positions (réponses) et les Arguments.	Soit à travers l'édition dans l'un des outils. Soit en répondant aux questions (ISSUES) oralement et/ou par écrit.	1- Définir les Issue 2- Proposer les positions, 3- Présenter les arguments.	Dans une base de données de l'un des outils ou dans des rapports écrits.	Sur l'un des outils.	Long. Demande l'implication de l'ensemble des Designers
PHI	Issue(problème) ==>Positions ==> Arguments ==> Positions==>Arguments ;...	Semi Formelle. Comporte que des question, sous questions et des arguments.	Très fin. Même les questions peuvent être détaillés.	Dynamique.	OUI	Pour les systèmes complexes, avec des constructions d'artefacts.	Orienté Processus.	Meme chose que IBIS.	A travers l'outils JANUS.	Le même processus que IBIS avec plusieurs itérations.	Dans la base de données de l'outils.	Sur l'un des outils.	Long. Demande l'implication de l'ensemble des Designers
REMAP	ISSUE, Position Argument (IBIS), ==> Exigence, décision contrainte et hypothèse.	Formelle, il est schématisé avec des liaisons entre les composants.	Peu détaillé. C'est les exigences qui orientent les arguments.	Dynamique.	OUI	Pour des grands projets ou les exigences sont nécessaires à atteindre.	Orienté Processus.	En modifiant les contraintes et hypothèses.	En éditant dans l'outil Ce dernier est primordiale si on veut utiliser le modèle REMAP	1- Etablir processus d'IBIS, 2- Alimenter par les exigences, 3- Prendre les décisions, 4- Déterminer les contraintes et hypothèses liées aux décisions prises.	Dans la base de données de l'outils.	Sur l'un des outils.	Long. Demande des itérations jusqu'à l'obtention d'un consensus sur l'exigence.
DRL	QUESTION ==> ALTERNATIVE==> OBJECTIF.	Formelle, il est schématisé avec des relation d'héritage pour les question et Objectifs.	Très détaillé.	Dynamique.	OUI	Pour des projets en équipe avec beaucoup de détail fonctionnelle.	Orienté Processus.	En ajoutant des QUESTIONS, ALTERNATIVES, OBJECTIVES.	En répondant aux questions par des alternatives. Les réponses sont édités soit dans le schéma DRL ou l'outil SYBIL.	1- Définir les Questions, 2- Présenter les alternatives, 3- proposer les objectifs	Dans les schémas écrits ou la base de l'outils.	Par un code informatique. (Algorithme)	Long. Demande un niveau de détails très fin.
QOS	Forme : Question==> Options ==>Critères.	Formelle (Schéma QOS)	Niveau fin. On peut atteindre le détail désiré.	Statique	NON	Les situations de conceptions ou les concepteurs sont amené à **faire des choix** entre plusieurs **possibilités**.	Orienté Processus.	Oui, en ajoutant des nouvelles questions, options et critères	En répondant aux questions, soit en groupe ou individuellement	1-Définir les questions, 2-Proposer plusieurs options, 3- Lister les critères de choix	Prise de décision en choisissant les solutions proposés dans un schémas formelle QOS ==> Rédigé.	Sur des rapports écrits.	Temps de réponse court. Décision prise en répondant à des questions directe.
ABRICO	But ==>Interprétations, ==>proposition et à la fin l'accord.	Formelle (documenter sous la forme de différents schémas)	Variable. Niveau de granularité dépend des acteurs.	Statique et dynamique	OUI	Pour les systèmes complexes. Pour les systèmes avec plusieurs parties prenantes.	Orienté Processus.	Oui, par ajour d'acteurs dans le diagramme dynamiques	A travers l'échange et la communication verbale et/ou écrite.	1- Définir le But, 2- Attendre l'interprétation du But, 3- Recevoir et émettre des propositions.	Les accords(décisions) sont enregistré dans le schéma choisie pour le modèle. ==> **Rédigé.**	Sur des schémas ou discussions écrites.	Temps de réponse Long. Des temps d'attente sont prévus.
QUIMERA	Forme : Question==> Options ==>Critères : Formules de calcule	Formelle. Schéma avec 2 Parties : (QOC) avec modèle de qualité (diagramme de classe)	Peu d'arguments. Basé sur le calcul.	Statique.	NON	Situation ou on désire mesurer et évaluer les critères de choix dans les décisions. Aussi, lorsque la qualité est exigée.	Orienté Processus.	Oui, impact sur les formules de calculs.	En se posant les questions, en apportant les options possibles et les valeurs de critères.	1-Définir les questions, 2-Proposer plusieurs options, 3- Etablir les formule de calcule de la qualité.	Par Schéma formelle avec des formules de choix des critères.	Sur des rapports écrits avec les relations de calculs.	Temps de réponse court. Les décisions sont prises en évaluant les critères de choix.
FBS	Fonction ==> Comportements ==> Structure	Semi Formelle (le schéma ne suffit pas pour présenter le Framework.	Très détail.	Dynamique	NON	Dans des systèmes avec plusieurs points de ressemblance. Nécessité d'adaptation.	Orienté Fonction.	Par changement des fonctions.	A partir de l'historique des DR existants.	1- Identifier la Fonction à construire, 2- Définir son comportement future, 2- Définir sont point final et sa structure.	En modifiant par écrit le modèle exploité. ==> Rédigé.	Sur des rapports écrits.	Temps de réponse court. On démarre pas du zero.

Session Web et Réseaux Sociaux

SpecificSearch : Un outil de recommandation automatique pour la veille d'information sur le web

Christophe Brouard[1], Christian Pomot[2]

[1]*Université Grenoble Alpes, LIG UMR 5217/équipe AMA, Grenoble, France*
Christophe.Brouard@imag.fr

[2]*Société Com&Net, Télespace Vercors, 38250 Villard-de-Lans, France*
cpomot@com-et-net.com

RÉSUMÉ. *Les systèmes de recommandation automatique par le contenu ont pour principale fonction de proposer à un utilisateur des informations susceptibles de l'intéresser sur la base des retours de pertinence qu'il a pu donner antérieurement sur d'autres informations. Différents algorithmes d'apprentissage automatique ont été intégrés à des systèmes de recherche d'information pour proposer des solutions permettant de réaliser cette tâche. Ces solutions n'ont cependant pas débouché sur des systèmes de recommandation automatique pour le web accessibles à tous. Il existe bien des agrégateurs de flux RSS permettant de recueillir de l'information sur le web mais les systèmes intégrant un apprentissage en sont encore à leurs balbutiements. Nous présentons ici les fonctionnalités et l'architecture d'une application web nommée SpecificSearch accessible en ligne qui se présente comme un agrégateur de flux RSS intégrant un apprentissage (http://www.specific-search.com). Une première évaluation permet de montrer la réalisabilité et l'utilité d'un tel système.*

ABSTRACT. *The main goal of a content-based recommender system is to propose to a user new documents which are likely to have some interest for him considering feedbacks he gave for other documents. Different machine learning algorithms have been integrated to information retrieval systems in order to cope with this task. However, these systems have not become as popular on the web as the well-known search engines. Otherwise, the RSS feed aggregators allow to gather information on the web but these systems do not integrate machine learning in order to improve the quality of the recommendations with the users' feedbacks. Here, we present the functionnalities and the architecture of a web application available online called SpecificSearch (http://www.specific-search.com) which is an RSS aggregator integrating a machine learning algorithm. A first evaluation shows how a such tool can be implemented and how it can be useful.*

MOTS-CLÉS : *recommandation automatique, veille d'information sur le web, filtrage adaptatif, flux RSS.*

KEYWORDS: *recommender system, web content monitoring, adaptive filtering, RSS feed aggregator*

Congrès INFORSID, 34ème édition, Grenoble, 31 mai – 3 juin 2016

1. Introduction

Les moteurs de recherche (Google, Bing, Yahoo, pour citer les plus utilisés[1]) constituent actuellement les outils incontournables pour l'accès à l'information sur le web. Ces outils de recherche permettent à l'utilisateur de saisir quelques mots clefs pour exprimer son besoin d'information et fournissent en retour une liste de pages web. Même si les moteurs de recherche intègrent une forme de personnalisation basée notamment sur les clics des utilisateurs sur les résultats proposés, cette adaptation est implicite et assez diffuse. Il n'est pas possible, pour l'utilisateur d'indiquer explicitement qu'un résultat correspond ou pas à un besoin d'information particulier. L'interaction entre l'utilisateur et ces outils est d'abord conçue comme un simple échange requête-réponse. Rien n'empêche l'utilisateur de reformuler sa requête en tenant compte de la première liste de résultats retournée par le moteur de recherche, mais ce travail de recherche et de composition de mots clefs lui revient. Si cette situation peut être admissible dans le cas d'une recherche ponctuelle, elle l'est beaucoup moins lorsque que le besoin d'information est récurrent comme dans le cas de la veille d'information où l'utilisateur souhaite être informé en permanence des nouveautés relatives à un sujet qui l'intéresse. En effet, dans ce cas, n'ayant pas la possibilité de capitaliser tout le travail réalisé lors des précédentes interactions, l'utilisateur répète les mêmes actions à chaque nouvelle interaction et le besoin de nouveaux outils adaptés et permettant une veille d'information efficace se fait alors ressentir. Les technologies et algorithmes permettant le développement de tels outils existent. Cependant, le développement d'outils dédiés à la veille d'information, en mesure d'apprendre les préférences de l'utilisateur et accessibles à tous comme le sont les moteurs de recherche en est encore à ses balbutiements et les fonctionnalités, l'architecture et le mode d'interaction avec l'utilisateur de tels outils restent à inventer.

L'objet de cet article est la présentation de SpecificSearch, un outil de veille d'information sur le web qui permet la capitalisation du travail de recherche et en particulier celui d'expression du besoin d'information. Il repose sur une interface homme-machine facilitant l'interaction et sur un apprentissage automatique permettant de garder une trace utile de celle-ci. La suite de l'article a la structure suivante : dans un premier temps, nous mettons en relief le fossé qui existe entre les besoins liés à une tâche de veille d'information et les outils classiques de recherche que l'on peut trouver sur le web. Dans un deuxième temps, nous décrivons des systèmes existants (agrégateurs de flux RSS et systèmes de filtrage adaptatif) qui apportent des solutions partielles par rapport aux besoins énoncés. Nous présentons ensuite SpecificSearch un outil empruntant aux différents systèmes décrits précédemment pour proposer une réponse la plus complète possible. Nous terminons en décrivant une première évaluation, la mise en place de l'outil et ses perspectives d'évolution.

[1] selon le site https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0

2. *La nécessité d'outils novateurs pour la veille d'information le web*

2.1. *Introduction à la notion de veille d'information*

La veille d'information englobe différents types de veille (la veille technologique, économique, réglementaire, etc...). Sans entrer dans le détail de cette typologie, Serge Cacaly (2008) définit la veille d'information ou veille informationnelle comme un « processus continu et dynamique faisant l'objet d'une mise à disposition personnalisée et périodique de données ou d'informations, traitées selon une finalité propre au destinataire, faisant appel à une expertise en rapport avec le sujet ou la nature de l'information collectée ». Cette définition fait notamment ressortir deux aspects importants de l'activité sur lesquels nous reviendrons par la suite. D'abord, le processus est continu, il ne correspond pas à un besoin ponctuel mais s'étend sur une certaine durée. Ensuite, il est lié à l'expertise de la personne réalisant la veille et suppose l'existence d'un savoir important permettant de distinguer dans un important volume d'informations celles qui sont pertinentes par rapport à un objectif ou une problématique donnée.

D'un point de vue plus analytique, on a coutume de distinguer plusieurs étapes dans une activité de veille. Selon, Elisabeth Noël (2008), on peut distinguer quatre étapes : le ciblage, la recherche, l'analyse et la diffusion. Le ciblage correspond à la définition précise du sujet d'intérêt et des objectifs de la veille ainsi qu'à la sélection des sources d'information. La recherche correspond au recueil d'informations pertinentes par rapport au sujet défini. L'analyse correspond à l'exploitation des informations recueillies afin d'en extraire du sens (cette étape fera intervenir des connaissances autres que celles présentes dans les informations recueillies). La diffusion correspond essentiellement à la préparation de l'information sous une forme intelligible et à sa transmission aux personnes susceptibles d'être intéressées.

2.2. *Ce qui manque aux traditionnels moteurs de recherche*

Les moteurs de recherche sont conçus pour des besoins d'information ponctuels. Ils permettent bien de trouver des sources d'informations et des informations pertinentes mais bien qu'il soit aussi possible de sauvegarder les sources d'informations (URLs de pages web) ou les informations (contenus des pages web) au moyen des favoris du navigateur, ces fonctions de sauvegardes restent rudimentaires. On notera que la sauvegarde de l'URL comme favori est une sauvegarde de la source d'information et non de l'information elle-même car si le contenu de la page change, l'information est perdue. L'information ne peut être réellement sauvegardée qu'en enregistrant tout simplement la page au moyen du navigateur ou d'un outil d'aspiration de site. Il y a aussi un problème de granularité car la sauvegarde de la totalité de la page web n'est pas adaptée si l'information pertinente se situe au milieu d'une page contenant beaucoup d'informations. De plus, les résultats non pertinents lus par l'utilisateur ne sont pas non plus marqués

comme ayant déjà été lus et l'utilisateur relira peut-être la page web avant de se souvenir qu'il l'avait déjà lue et conclu à son inadéquation par rapport à sa recherche. De même, tout le travail de formulation de la requête consistant à trouver les bons mots clés par reformulations successives en fonctions des résultats retournés par le moteur de recherche n'est pas non plus capitalisé. Enfin, les moteurs de recherche n'offrent pas d'outil d'annotation pour commenter les résultats et transmettre l'information et son commentaire à d'autres utilisateurs.

Par ailleurs, les moteurs de recherche sont des outils dits « PULL ». Ils nécessitent une action de l'utilisateur pour aller chercher l'information. Cela signifie que l'utilisateur devra régulièrement se connecter aux différents moteurs ou outils de recherche sur les sites pertinents par rapport à son sujet de veille pour aller chercher l'information. Cette vérification de l'utilisateur peut se révéler très fastidieuse et inutile si aucune information intéressante n'est apparue sur le site qu'il surveille. Les outils dits « PUSH » sont mieux adaptés à une activité de veille, dans ce cas, l'utilisateur utilise un seul outil qui récupère automatiquement les informations à partir des différentes sources et les lui présente.

3. *Les outils et technologies existantes*

3.1. *Flux RSS et agrégateurs de flux*

Les flux (ou fils) RSS sont de simples fichiers XML respectant une structure XML prédéfinie[2]. Au même titre que des pages HTML, ce fichier est servi par le serveur web associé au site et récupéré au moyen d'une requête HTTP. Ce fichier peut être récupéré et affiché par n'importe quel navigateur (cf Fig.1). Il est composé d'une suite d'items (par exemple les différentes nouvelles liées à l'actualité) chaque item étant lui-même composé d'un titre, d'une description et d'un éventuel hyperlien renvoyant sur une page web entrant dans le détail de la nouvelle. Il est souvent mis à jour pour contenir uniquement les informations les plus récentes (de nouvelles informations sont ajoutées, d'anciennes sont supprimées).

De nombreux sites proposent des résumés de l'actualité de leur contenu sous forme de flux RSS. Si le flux RSS pour une page web que l'on souhaite surveiller n'existe pas on peut le créer à partir d'outils existants[3]. Il suffit d'indiquer la façon dont il faut analyser le fichier HTML correspondant à la page à surveiller pour que l'outil génère un fichier RSS et une URL pour le récupérer. Cela fonctionne à condition que le fichier HTML dispose d'une structure et que cette structure ne change pas en permanence. D'autres outils dédiés de même nature mais pour des types de sites particuliers ayant tous la même structure existent aussi. Il est par exemple possible de transformer les messages d'un mur facebook ou les tweets d'un

[2] la spécification du format est ici : http://www.rssboard.org/rss-specification
[3] comme http://feed43.com/, http://www.page2rss.com/, ...

compte twitter en items d'un flux RSS[4]. A l'origine Facebook et Twitter donnaient accès à un flux RSS associé à chaque compte. Mais il existe une tendance pour certains sites, pour maximiser leur trafic, à forcer l'utilisateur à consulter les informations qu'il délivre directement sur le site en ne proposant plus la création automatique de flux RSS qui pourraient être consultés à partir d'outils externes. Cette tendance est corroborée par certaines expérimentations montrant que dans certains cas, il est préférable pour un site de se passer de flux RSS pour maximiser son trafic (Dan, 2012). Il restera néanmoins toujours possible de passer par des outils externes. Potentiellement, beaucoup de contenu web peut donc être transformé en flux RSS. L'un des intérêts des flux RSS est de pouvoir découper la page web en unités de contenu et donc d'avoir un niveau de granularité d'information plus faible et même réglable si on passe par un outil générant le flux à partir de la page.

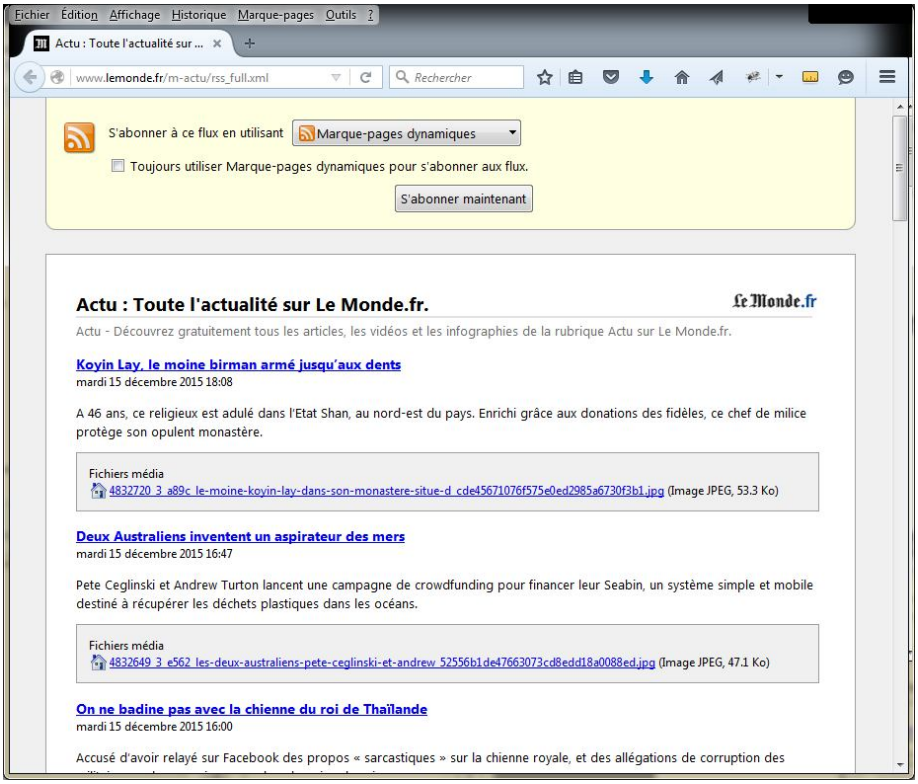


Figure 1. *Affichage du flux d'URL : http://www.lemonde.fr/m-actu/rss_full.xml dans un navigateur*

[4] http://twitrss.me/ et http://www.wallflux.com/fr/

Bien que ces fichiers RSS puissent s'afficher dans des navigateurs, des outils dont l'interface conviviale permet l'abonnement, l'affichage et le rafraîchissement automatique de plusieurs flux existent. Ces outils s'appellent des agrégateurs de flux. Il en existe de nombreux. Il peut s'agir d'agrégateurs en ligne (une application web sur laquelle l'utilisateur se crée un compte) comme netvibes[5] ou feedly[6] ou bien d'applications s'exécutant en local sur le poste de l'utilisateur comme par exemple rssowl[7] ou thunderbird (l'outil de messagerie).

En plus de l'intérêt des flux RSS qu'ils manipulent, les agrégateurs de flux offrent, comparativement aux moteurs de recherche, différents avantages pour la veille d'information. D'abord, il s'agit d'outils « PUSH ». L'utilisateur n'a plus besoin d'aller chercher l'information sur les différents sites ou moteurs de recherche, les différents fichiers XML sont récupérés automatiquement. Par ailleurs, certains agrégateurs permettent aussi de stocker les différents résultats recueillis.

Même si certains agrégateurs permettent le filtrage par mots clés, cette fonctionnalité reste assez rudimentaire. Il est en effet difficile d'identifier tous les bons mots clés. Il est aussi impossible de pondérer ces mots et le filtrage est binaire : soit l'information contient les mots clés et elle est sélectionnée soit elle ne les contient pas et elle est ignorée. Or cette fonction de filtrage est souvent utile car même s'il existe des flux dont le sujet est souvent très focalisé et que ces flux contiennent des informations qui peuvent potentiellement toutes intéresser l'utilisateur, il existe aussi des cas où la recherche de l'utilisateur est tellement précise qu'aucun flux ne correspond totalement à la recherche ou des cas où certains flux ne délivrent qu'occasionnellement de l'information pertinente. De façon plus globale, comme nous l'avons évoqué précédemment un sujet ou un objectif de veille d'information est souvent complexe et liée aux connaissances de l'expert et ne peut se résumer à un filtrage binaire sur de simples mots clés.

3.2. *Les systèmes de filtrage adaptatif*

De nombreux systèmes de filtrage (ou recommandation) automatique (Bodilla et al, 2013) basés sur les retours de pertinence des utilisateurs ont été conçus. Parmi ces systèmes, on peut distinguer, les systèmes de filtrage collaboratif et les systèmes de filtrage basés sur le contenu. Dans le cas du filtrage collaboratif, les retours de pertinence d'un utilisateur pour une information sont utilisés pour évaluer la pertinence de cette information pour les autres utilisateurs. Nous nous focalisons ici sur le cas du filtrage basé sur le contenu où l'on considère un utilisateur isolé. Le principe de ces systèmes est le suivant : à partir d'un ensemble d'exemples

[5] http://www.netvibes.com/fr
[6] https://feedly.com
[7] http://www.rssowl.org/

d'informations pertinentes et non pertinentes pour un utilisateur et un sujet de veille donné, la tâche consiste à évaluer la pertinence d'un nouveau document. Ces systèmes s'appuient sur des systèmes de classification automatique (Joachims, 1998) ou sur des systèmes de recherche d'information (Robertson et Walker, 2001), (Wu et al, 2001) qui intègrent une méthode de mise à jour du poids des termes comme la méthode Rocchio (Rocchio, 1971). Ils sont en mesure d'améliorer la qualité de la sélection au fur et à mesure des retours de pertinence des utilisateurs. Les conférences TREC 2000, 2001 et 2002 (Robertson et Hull, 2000), (Robertson et Soboroff, 2001&2002) ou TREC KBA[8] ont été l'occasion de comparer les performances de différents systèmes de filtrage sur différents corpus de textes et de se confronter à différents problèmes liés à cette tâche notamment à celui de la définition d'un seuil de sélection (Arampatzis, 2001) et à celui de « démarrage à froid » (Blerina, 2014) car en effet, la tâche est particulièrement difficile lorsque peu d'exemples de documents pertinents sont donnés au départ.

Malgré ces difficultés, les diverses expérimentations ont montré que ces systèmes sont en mesure d'apprendre à sélectionner les informations pertinentes de façon beaucoup plus fiable qu'un simple filtrage binaire basé sur des mots clés. Dans ces systèmes, de très nombreux mots clés ajoutés automatiquement avec différentes pondérations calculées automatiquement sont pris en compte et les pondérations sont combinées de diverses façons pour calculer un score global. Ces bons résultats ont conduit à quelques initiatives récentes pour une application de ce type de système à la veille d'information sur le web (Katakis et al., 2009), (Nanas et al., 2010), (Paliouras et al., 2008). Actuellement, à notre connaissance, deux applications web pour la veille d'information intègrent un apprentissage automatique : il s'agit de Prismatic[9] et Noowit[10]. Néanmoins, avec ces outils, soit le choix des sources d'information reste très limité, soit les temps de réponse et l'ergonomie rendent leurs utilisations difficiles et semblent indiquer que ces outils ne sont encore pas totalement aboutis.

4. Présentation de SpecificSearch

Considérant l'existence de systèmes capables d'apprendre à filtrer l'information sur la base de retours de pertinence et aussi celle d'un format de description de l'information sur le web (RSS) et considérant aussi l'essor de technologies permettant le développement d'interfaces web suffisament sophistiquées (comme les librairies JavaScript), il semble que les ingrédients nécessaires au développement de moteurs de recherche accessibles à tous et dédiés à la veille d'information pour le web soient maintenant présents. En nous appuyant sur le système de filtrage Echo

[8] http://trec-kba.org/
[9] http://getprismatic.com/
[10] http://www.noowit.com/

(Brouard, 2012), nous avons conçu et développé une application web permettant la veille d'information sur le web en s'appuyant sur les flux RSS. Nous décrivons dans cette partie les fonctionnalités, l'interface et l'architecture de cette application.

4.1. *Les différentes étapes dans l'utilisation de l'outil*

4.1.1. *Création d'une recherche*

Contrairement à la plupart des agrégateurs, SpecificSearch est centré « recherche » et non « flux ». L'utilisateur ne commence donc pas par s'abonner à des flux mais par créer une recherche, c'est-à-dire par définir un sujet d'intérêt (étape « ciblage » de la veille).

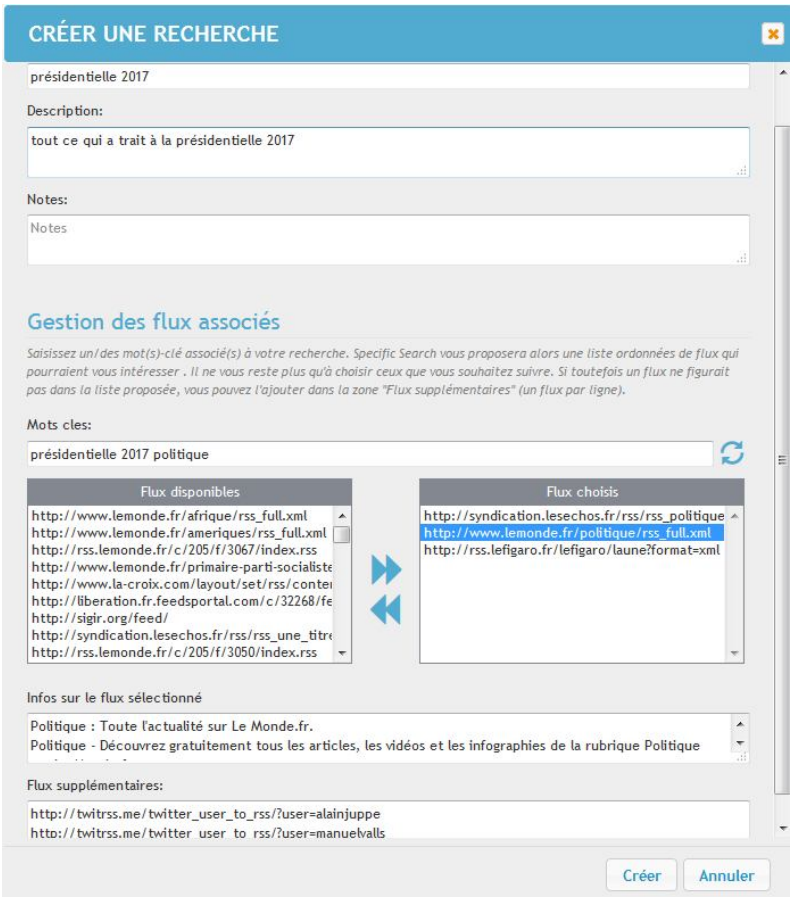


Figure 2. *Boîte de dialogue permettant la création d'une recherche (un sujet de veille), avec recherche de flux par mots clés. Il est aussi possible d'ajouter directement les URL s des flux dans le champ « flux supplémentaires ».*

Supposons que le sujet de la veille concerne tout ce qui a trait à la l'éléction présidentielle française de 2017. L'utilisateur commence par ouvrir la fenêtre de dialogue de création de recherche. Une fois cette fenêtre ouverte (cf Fig.2), il commence par nommer sa recherche puis ensuite sélectionne des sources d'informations à surveiller. A ce sujet, l'une des explications au succès très relatif des flux RSS semble être la difficulté pour un large public à sélectionner une source. En effet, il faut d'abord trouver un flux correspondant à la recherche (les annuaires de flux ne sont pas faciles à trouver, peu nombreux et très incomplets bien qu'en voie d'amélioration notamment avec feedly ou Instant RSS Search[11] basé sur une api Google). Puis il faut aussi souvent copier-coller une URL dans son outil. Afin d'éviter cette recherche et ce copier-coller d'URL pas toujours maîtrisé, SpecificSearch permet la recherche de flux par mots clés. L'utilisateur va par exemple taper les mots clés « présidentielle 2017 politique » dans le champ mots clés. De nombreux flux lui sont alors proposés. Il choisit parmi ces flux ceux qui lui semblent les plus pertinents et/ou ajoute manuellement des URL s de flux.

4.1.2. *Saisie des retours de pertinence et stockage des informations*

Une fois la recherche validée, les informations provenant des différents flux choisis sont affichées dans un onglet « Nouveaux résultats » (cf Fig.3).

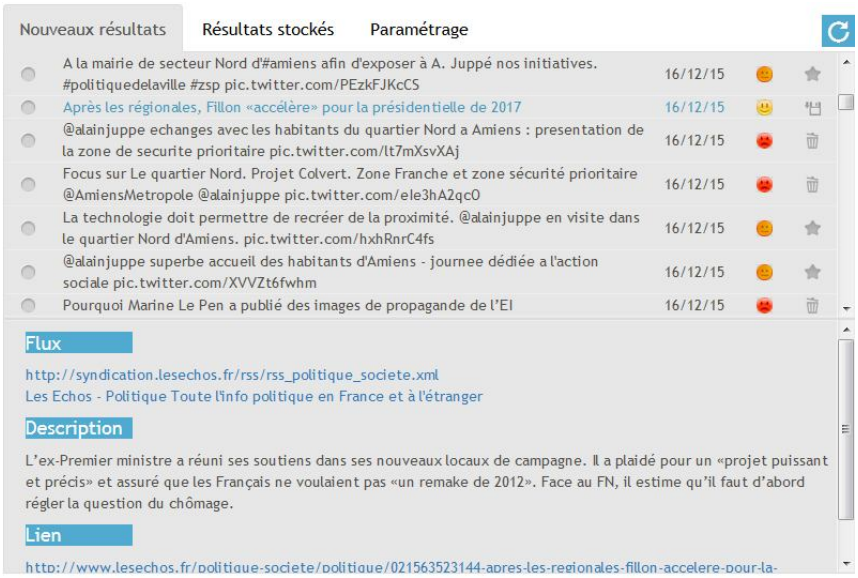


Figure 3. *Les nouveaux résultats sont affichés. L'utilisateur indique au moyen d'une émoticône si une nouvelle est pertinente ou pas. Indépendemment de son retour de pertinence, il peut aussi choisir de sauvegarder ou de supprimer une nouvelle.*

[11] http://ctrlq.org/rss/

L'utilisateur va alors commencer à apprendre au système à sélectionner les informations pertinentes en indiquant pour certaines nouvelles, au moyen d'une émoticône si elles sont pertinentes ou pas. Il peut aussi choisir de sauvegarder certaines nouvelles (elles seront alors déplacées dans l'onglet « Résultats stockés ») ou de les supprimer (elles seront alors déplacées dans l'onglet « Résultats supprimés » qui n'apparaît pas par défaut).

4.1.3. *Recalcul des scores de pertinence et visualisation des résultats*

Une fois les retours de pertinence donnés, l'utilisateur peut demander le calcul des scores de pertinence pour toutes les nouvelles non jugées. Une fois le calcul effectué par Echo (cf section 4.2.2), les nouvelles apparaissent alors classées par ordre de pertinence décroissante (cf Fig. 4).

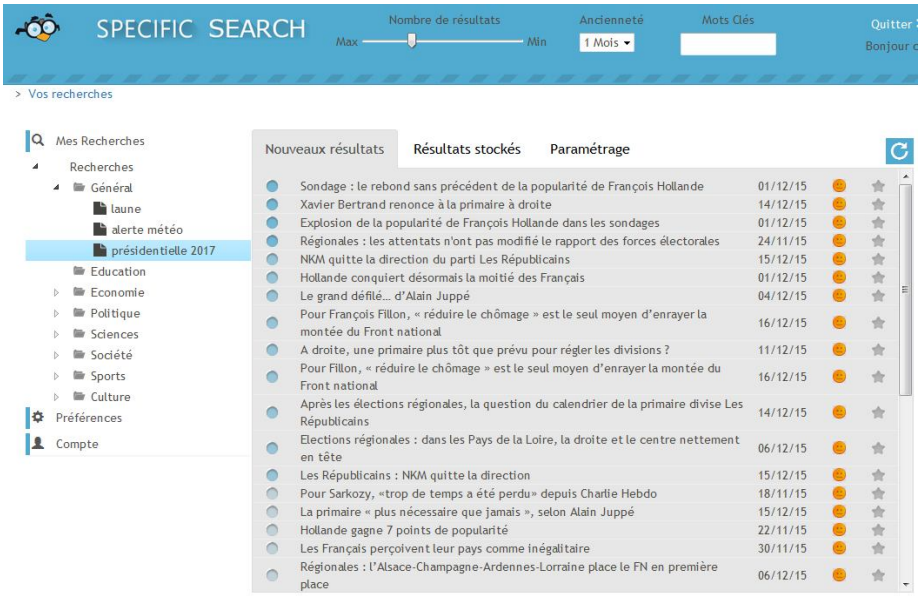


Figure 4. *Une fois le calcul des scores et des probabilités de pertinence effectués, les nouvelles sont affichées dans l'ordre des scores de pertinence décroissant. L'intensité de la coloration de la puce qui précède le titre de la nouvelle indique sa probabilité de pertinence. Il est aussi possible de filtrer les nouvelles sur cette probabilité, sur leur ancienneté ou encore sur de simples mots clés. Globalement, on trouve à gauche de l'interface, l'arborescence des recherches (sujets de veille) et au centre les informations propres à la recherche sélectionnée.*

Une probabilité de pertinence est indiquée au moyen de l'intensité de la coloration de la puce qui précède le titre de la nouvelle. L'utilisateur a la possibilité de n'afficher que les nouvelles dont la probabilité dépasse un certain seuil. Il peut aussi choisir l'ancienneté des nouvelles affichées et filtrer sur des mots clés. Sur la

base de ce nouvel affichage, l'utilisateur peut donner de nouveaux retours de pertinence et demander par un simple clic le recalcul des scores et un nouvel affichage et itérer ce processus autant de fois qu'il le souhaite.

4.2. *L'architecture de l'outil*

4.2.1. *Architecture générale*

SpecificSearch est une application web orientée client. Elle s'appuie sur un framework MVC PHP et sur la librairie JavaScript JQuery. Une représentation schématique de son architecture est donnée dans la figure ci-dessous (cf Fig.5).

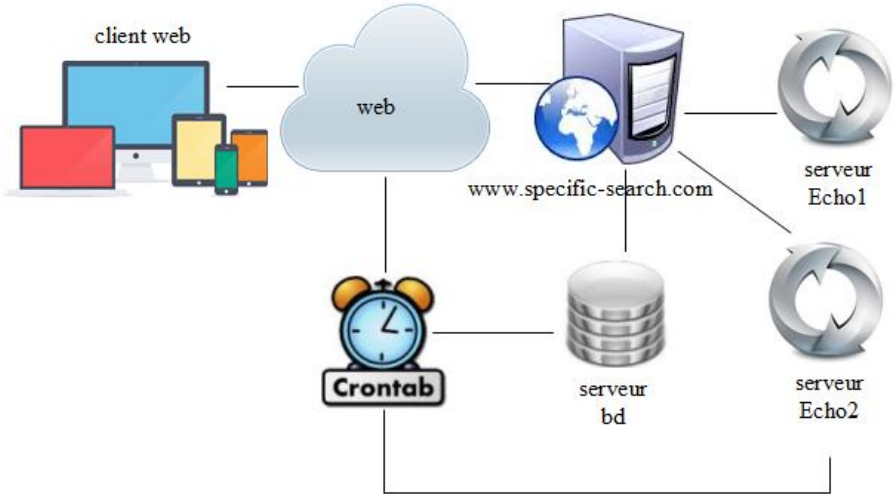


Figure 5. *Architecture de l'application web SpecificSearch. Le navigateur, le serveur web et le serveur de base de données sont les ingrédients classiques d'une application web. Par ailleurs, deux serveurs Echo dialoguent avec le serveur web. Echo1 calcule les scores de pertinence des nouvelles étant donnés les retours de pertinence de l'utilisateur. Echo2 calcule les scores de pertinence de flux pour des mots clés. Des scripts exécutés régulièrement et automatiquement récupèrent les nouvelles des différents flux sauvegardés dans la base de données et mettent à jour l'index des flux de Echo2.*

Les traitements qui ne nécessitent pas de données du serveur sont exécutés par le navigateur sur le poste client. Les traitements qui nécessitent l'intervention du serveur sont appelés au moyen de requêtes AJAX. Certains traitements comme la récupération des nouvelles sont effectués hors ligne auprès des différents serveurs web servant les différents flux RSS. Pour les calculs de pertinence, le serveur web fait appel au système Echo qui a été réécrit comme un serveur. Des scores de

pertinence sont calculées pour les nouvelles sur la base des retours de pertinence donnés mais aussi pour les flux dans le contexte de la recherche de flux par mots clés. Dans les deux cas, il s'agit bien du système Echo qui est à l'œuvre, mais deux serveurs Echo différents tenant compte d'un besoin d'interaction différent ont dû être mis en place. Ces deux types d'interaction qui constituent l'originalité de cette architecture sont détaillés dans la suite.

4.2.2. *Le système Echo*

Echo (Brouard, 2012) est un système de sélection d'information pertinente. Il peut être appliqué à différents types de problèmes, comme la recherche d'information (sélection d'un document à partir d'une requête), la classification supervisée (sélection d'une classe à partir d'un document) ou encore l'extension de requête (sélection d'un terme à partir d'un ensemble de termes). Il est basé sur une formalisation de la notion de pertinence qui combine les notions de spécificité et d'exhaustivité qui sont au cœur des modèles de pertinence en recherche d'information (Brouard, 2004). Echo peut être décrit comme un système de construction et d'exploitation de réseau associatif s'appuyant sur des mécanismes neuronaux simples. La construction du réseau s'appuie sur la règle de Hebb liant deux informations survenant simultanément (deux termes cooccurant dans le même document par exemple), son exploitation s'appuie sur une méthode de propagation et une mesure d'écho c'est-à-dire la mesure d'une quantité d'activation rétro-propagée vers les sources d'activation. Le système a été appliqué avec succès à différents problèmes dont celui de la classification de textes qui nous intéresse plus particulièrement dans le cadre de cette application. Il obtient sur cette tâche des performances légèrement inférieure aux SVM mais supérieures à la méthode des k plus proches voisins (meilleure méthode hors SVM) sur le corpus de référence Reuters-21578 (Brouard, 2012). Echo peut par ailleurs être appliqué à de grandes quantités de données. Ainsi Echo a été appliqué lors du deuxième challenge international "Large Scale Hierarchical Text Classification"[12] et dans le cadre de la compétition Kaggle[13] à la classification automatique de plusieurs millions de documents dans plusieurs centaines de milliers de classes. Les résultats obtenus par Echo dans le cadre de ces compétitions (1^{er}/17 et $9^{ème}$/115) ont montré son efficacité. Echo a par ailleurs l'avantage d'être totalement incrémental (il est inutile de reconstruire tout le réseau lorsque des exemples sont ajoutés ou supprimés). Enfin, les connexions dans le réseau pouvant s'interpréter comme des règles « SI ... ALORS », ses choix de classification sont facilement explicables à l'utilisateur.

[12] http://lshtc.iit.demokritos.gr/
[13] https://www.kaggle.com/c/lshtc

4.2.3. *L'API du système de calcul des scores des informations*

Une première forme du serveur Echo (Echo1) est utilisée pour calculer les scores de pertinence lors d'une demande de mise à jour de l'utilisateur. Dans cette forme, un seul type de requête est autorisé : le serveur prend en entrée une chaîne de caractères décrivant toutes les nouvelles. Cette chaine contient, pour chaque nouvelle, l'identifiant de la nouvelle, les mots clés qu'elle contient et le jugement de pertinence associé (pertinent, non pertinent ou pas de jugement). Le serveur retourne une chaîne de caractères correspondant aux scores et probabilités de pertinence de toutes les nouvelles auxquelles aucun jugement de pertinence n'a été associé. La probabilité de pertinence est calculée sur la base des distributions des scores calculés par Echo des documents pertinents et non pertinents (Brouard, 2012). Dès lors qu'un nouveau jugement de pertinence est donné tous les scores doivent être recalculés. Les volumes de données échangés dans ce cas restent relativement restreints (quelques centaines, voire quelques milliers de nouvelles tout au plus) car basés sur les retours de pertinence de l'utilisateur qui sont donnés manuellement.

4.2.4. *L'API du système de sélection des sources d'information*

Une autre forme du serveur Echo (Echo2) a dû être mise en place pour répondre au problème de la recherche de flux par mots clés. Contrairement à Echo1 où tout est reconstruit à chaque fois, Echo2 garde un index en mémoire associant mots clés et identifiants de flux. Deux types de requêtes sont possibles : les requêtes relatives à l'ajout de nouvelles dans l'index et des requêtes relatives au calcul de scores de pertinence de flux pour des mots clés particuliers. L'index est construit sur la base de toutes les nouvelles contenues dans la base de données (actuellement plusieurs centaines de milliers et potentiellement plusieurs millions) et est mis à jour régulièrement et incrémentalement avec les nouvelles informations recueillies. Les volumes de données échangés lors de la construction de l'index sont donc très importants. Ils sont par contre très réduits lors des requêtes de calcul de scores (quelques mots clés en entrée et une centaine d'identifiants de flux en retour).

5. Première évaluation

Bien que les performances du système Echo aient déjà été testées largement par ailleurs, nous avons réalisé une première évaluation de l'outil afin de vérifier que le système s'appliquait bien à des flux d'actualités (textes de taille réduite), répondait suffisament vite quand on l'intègrait à une application web (car l'apprentissage est réalisé en ligne à chaque demande de mise à jour de l'utilisateur) et ne nécessitait pas trop de retours de pertinences pour fournir des résultats intéressants en compraison d'un simple système de recherche par mots clefs. En particulier nous avons étudié l'évolution de la qualité des réponses avec les retours de pertinence et nous les avons comparés avec un simple filtrage sur mot clef. Cinq sujets de veille

ont été définis (portant respectivement sur la crise des migrants, les suites des attentats du 13 novembre, le football, la présidentielle de 2017 et le monde de l'éducation). Pendant un mois, les nouvelles de la une du journal « Le Monde », ce qui représente environ mille nouvelles, ont été étiquetées comme pertinentes ou non vis-à-vis de ces différents sujets. Puis des retours positifs et négatifs sur les premiers documents retournés par le système ont été simulés. Différentes quantités de retours, respectivement 5/10 (5 positifs, 10 négatifs), 10/20 et 15/30 ont été testées pour les 15 premiers jours. En fonction de ces retours, la précision à respectivement 5, 10, 15 documents retournés et la R-précision (c'est-à-dire la précision lorsque le nombre de documents retournés est égal au nombre de documents pertinents) a été calculée sur les 15 derniers jours. De façon à montrer que la tâche n'était pas triviale, les résultats du système ont été comparés avec un simple filtrage par mot clef (en choisissant celui donnant les meilleurs résultats, respectivement « migrant », « déchéance », « foot », « primaire » et « université »).

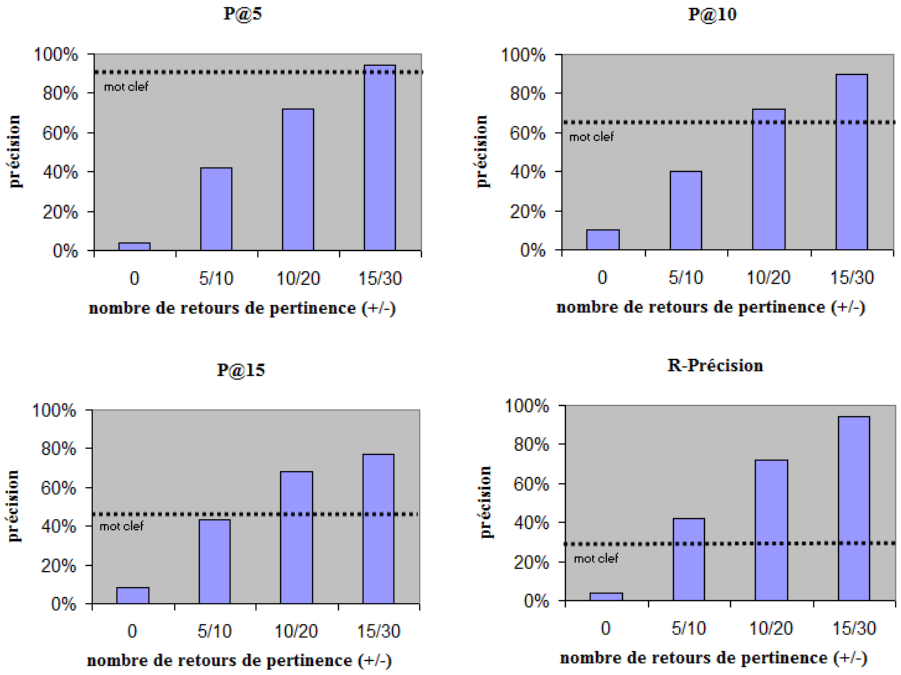


Figure 6. *Dans le cas des précisions à 15 et de la R-précision, dès lors qu'au moins 10 retours de pertinence positifs et 20 retours négatifs sont donnés par l'utilisateur, les résultats sont très nettement en faveur du système basé sur l'apprentissage automatique en comparaison d'un simple système de filtrage par mots clefs. Dans le cas des précisions à 5 et à 10, l'utilisateur doit donner plus de retours de pertinence pour obtenir de meilleur résultats.*

Il apparaît que la précision augmente bien avec le nombre de retours de pertinence donnés et devient vite supérieure à un filtrage par mot clef (cf Fig. 6). Plus précisément, il apparaît que le filtrage par mot clef ne permet pas de retrouver de nombreuses nouvelles pourtant pertinentes mais qui ne contiennent pas le mot clef (une nouvelle décrivant le résultat d'une équipe de football connue pourra par exemple ne pas contenir le mot « foot ») et que des nouvelles qui contiennent le mot clef peuvent ne pas être pertinentes (une nouvelle contenant le mot « primaire » peut concerner les primaires américaines et non la politique française). L'apprentissage automatique permet de retenir un plus grand ensemble de mots, de pondérer l'importance respective de chacun de ces mots et de combiner ces pondérations de façon optimale. Par ailleurs, les temps liés au calcul des scores et à la réponse de Echo1 se sont révélés inférieurs à la seconde et ceci même en considérant plusieurs centaines de documents exemples et tests. Ces résultats permettent de conclure que l'apprentissage en ligne qui incluait la préparation des données à envoyer au serveur Echo1, leur traitement par Echo1, la récupération des résultats et leur rangement dans la base de données est donc réalisable. Pour Echo2, la construction du réseau qui prend moins d'une minute pour plusieurs centaines de milliers de nouvelles est réalisée hors ligne.

6. Conclusion

Une interface homme-machine intégrant des fonctionnalités permettant de faciliter l'activité de veille sur le web a été proposée. Une architecture client-serveur intégrant un système d'apprentissage automatique a été conçue. Bien que la première évaluation soit relativement limitée et bien que la réalisation d'expérimentations en conditions réelles avec différents utilisateurs véritablement engagés dans un processus de veille sur une durée dépassant le mois semble incontournable, on peut considérer que le résultat obtenu montre déjà la faisabilité d'un tel outil (temps de réponse et qualité des résultats).

L'outil a été mis en ligne (http://www.specific-search.com) à la fin de l'année 2015. N'importe quel internaute peut maintenant se créer un compte et utiliser l'outil, laisser des commentaires, des souhaits d'évolution ou signaler des bugs. SpecificSearch est actuellement en bêta-test et nous comptons sur des retours nombreux et pertinents pour l'améliorer. Il est plus que probable que l'outil évoluera encore. Certaines évolutions concerneront notamment la mise en place d'outils d'annotation relatifs à l'étape de diffusion de la veille d'information (l'outil se limitant pour l'instant aux étapes de ciblage et de recherche), d'autres concerneront l'explication de la sélection d'une nouvelle ou encore l'auto-évaluation du système dans sa capacité à sélectionner l'information pertinente. Par ailleurs, nous sommes conscients que si le succès est au rendez-vous et que les données dans l'index deviennent plus volumineuses, l'infrastructure actuelle risque de ne pas être adaptée. Une version distribuée d'Echo est à l'étude.

7. Bibliographie

Arampatzis A., Beney J., Koster C.H.A, van der Weide T.P., Incrementality, Half-life, and Threshold Optimization for Adaptive Document Filtering. *Proceedings of the TextRetrieval Conference (TREC9),* NIST Special Publication, p.589-600, 2001.

Blerina L., Kostas K., Stathes H.,« Facing the cold start problem in recommender systems», *Expert Systems with Applications*, 41, p. 2063-2073, 2014.

Bobadilla J., Ortega F., Hernando A., Gutierrez R. « Recommender systems survey», *Knowledge-Based Systems*, 46, p. 109-132, 2013.

Brouard C., « Document Classification by Computing an Echo in a Very Simple Neural Network», *ICTAI*, 735-741, 2012.

Brouard C., Nie J.Y., « Relevance as Resonance: a New Theoretical Perspective and a Practical Utilization in Information Filtering », *Information Processing and Management*, 40, p. 1-19, 2004.

Cacaly S., Le Coadic Y-F, Pomart P-D., Sutter E., Dictionnaire de l'information (3e édition), Paris, A. Colin, 95 p., 2008.

Dan M., « Use of RSS feeds to push online content to users», *Decision Support Systems,* 54, p. 740-749, 2012.

Joachims T. « Text categorization with support vector machines:Learning with many relevant features », In Proceeding of ECML-98, 1998.

Katakis I., Tsoumakas G., Banos E., Bassiliades N., Vlahavas I.« An adaptive personalized news dissemination system», *Journal of Intel. Inform. Systems*, 32(2), p. 191-212, 2009.

Nanas N., Manolis V., Elias H.,« Personalised news and scientific literature aggregation», *Inform. Process. and Management*, 46, p. 268-283, 2010.

Noël E., « Veille et nouveaux outils d'information ». In DINET Jérôme, Usages, usagers et compétences informationnelles au XXIème siècle. Paris, Hermès; p. 257-284, 2008.

Paliouras G., Mouzakidis A., Moustakas V., Skourlas C., « PNS: A personalized news aggregator on the web », *Intelligent Interactive Systems in Knowledge-based Environments*, 104, p. 175-197, 2008.

Robertson S., Hull D. « The TREC-9 filtering track final report », *Proceedings of the TextRetrieval Conference (TREC9),* NIST Special Publication, 2001.

Robertson S., Walker S. « Microsoft Cambridge at TREC-9: Filtering Track», *Proceedings of the TextRetrieval Conference (TREC9),* p. 361-368, 2001.

Robertson S., Soboroff I. « The TREC-2002 filtering track report », *Proceedings of the TextRetrieval Conference (TREC11),* NIST Special Publication, 2003.

Rocchio J.J., « Relevance Feedback in Information Retrieval », *In The Smart Retrieval System, G. Salton (Ed.), Prentice* Hall, p. 313-323, 1971.

Wu L., Luang X., Guo Y., Zhang Y. « FDU at TREC-9:CLIR, Filtering and QA Tasks », In *Proceedings of the TextRetrieval Conference (TREC9),* p. 189-202, 2001.

Minimisation de l'influence négative dans les réseaux sociaux

Etat de l'art et Ouvertures de recherche

Zakia Challal[1], Kamel Boukhalfa[2]

1. LSI, Université des Sciences et de la Technologie Houari Boumediene
BP 32 El Alia 16111 Bab Azzouar Alger, Algérie
zchallal@usthb.dz

2. LSI, Université des Sciences et de la Technologie Houari Boumediene
BP 32 El Alia 16111 Bab Azzouar Alger, Algérie
kboukhalfa@usthb.dz

RESUME. L'influence dans un réseau social est la capacité d'un utilisateur à mener un autre utilisateur à adopter une idée, une opinion, à croire à une information, à utiliser des services ou des produits, etc. L'influence peut être négative si de mauvaises intensions s'imprègnent tels que de mauvaises idées, des rumeurs, des virus. Dans cet article, nous présentons les travaux proposés dans la littérature pour minimiser la propagation de l'influence négative dans un réseau social et protéger ainsi les utilisateurs du réseau. L'analyse des travaux proposés nous a mené à dresser quelques ouvertures de recherche.

ABSTRACT. The influence in a social network is the ability of a user to conduct another user to adopt an idea, an opinion, to believe in information, use services or products, etc. The influence can be negative if bad intensions are impregnated such as bad ideas, rumors, and viruses. In this article, we present the work proposed in the literature to minimize the spread of the negative influence in a social network and protect network users. The analysis of the proposed work has led us to draw some research openings.

MOTS-CLES : influence négative, réseau social.

KEYWORDS: negative influence, social network.

1. Introduction

Les utilisateurs des réseaux sociaux sont envahis par le flux d'information diffusé continuellement et ils sont généralement influencés par ce dernier. Ainsi, ils rediffusent l'information, commentent des publications, achètent des produits ou des services et parfois organisent des évènements, signent des pétitions, etc.

La diffusion d'information dans les réseaux sociaux a suscité l'intérêt de la communauté de recherche. Cette dernière s'est particulièrement intéressée à la problématique suivante : *comment faire pour arriver à une large diffusion et maximiser l'influence à travers le réseau ?* Beaucoup de travaux ont été menés dans ce sens en ne considérant que l'influence positive (Domingos et Richardson, 2001; 2002 ; Chen, Wang et Yang, 2009 ; Chen, Yuan et Zhang, 2010 ; Kempe, Kleinberg et Tardos, 2003 ; Li et al., 2014 ; Liu et al., 2012; Nguyen et Zheng, 2013 ; Wang, Camacho et Xu, 2009). Cependant, s'intéresser seulement à l'influence positive n'est pas pertinent car l'influence négative dans les réseaux sociaux est très présente, notamment les rumeurs, et leurs propagation dans le réseau conduit parfois à des résultats néfastes (Fan et al., 2013 ; Nguyen et al., 2012). Par exemple, la rumeur lancé sur Twitter affirmant la mort du président de la Syrie a conduit jusqu'à l'augmentation du prix du pétrole (Fan et al., 2013). Si une mauvaise information est lancée sur le réseau, elle est généralement plus rapidement propagée qu'une bonne information et elle a plus d'effet (Baumeister et al., 2001 ; Chen et al., 2010). Par exemple, une information sur la corruption d'un responsable se propage plus rapidement qu'une information sur ses travaux innovants. Dans ce cas, on voudra certainement minimiser l'influence au lieu de la maximiser. En marketing, par exemple, les entreprises souhaiteraient faire face aux contres publicités et protéger ainsi leur image de marque en minimisant leurs propagations. En politique, lors des élections, les politiciens veillent à minimiser les contres campagnes. En éducation, on doit minimiser l'influence négative exercée sur les enfants, etc.

Les travaux de la littérature sur la minimisation de l'influence négative dans les réseaux sociaux proposent des solutions qu'on peut classer en trois approches : *blocage de nœuds*, *blocage de liens* ou *utilisation de l'influence compétitive* pour faire une contre campagne qui limite la diffusion de l'influence négative. Nous présentons dans ce papier les différents travaux que nous avons trouvé dans la littérature sur la minimisation de l'influence négative tout en analysant les différentes approches proposées. Des ouvertures de recherches seront proposées à la fin de l'article.

L'article est organisé comme suit : nous présentons tout d'abord, dans la section 2, trois problématiques liées à la minimisation de l'influence qui sont : la propagation de l'influence, la détection d'influenceur et la maximisation de l'influence dans les réseaux sociaux. Dans la section 3, nous ferons une revue des travaux sur la minimisation de l'influence négative dans les réseaux sociaux. Une synthèse des travaux et pistes ouvertes seront présentées dans la section 4. Nous finirons par une conclusion et une vision sur nos travaux futurs.

2. Problématiques connexes

Nous définissons dans cette section les notions liées au problème de minimisation de l'influence négative dans un réseau social. Nous définissons les modèles de propagation d'influence utilisés, quelques métriques de détection d'influenceurs et le problème de maximisation d'influence lorsqu'il s'agit d'influence compétitive.

2.1. Modèles de propagation d'influence dans un réseau social

Les modèles de propagation d'influence dans un réseau social les plus utilisés sont : *Independent Cascade model (ICM)* et *Linear Threshold Model (LTM)* (Shakarian et al., 2015). Dans ces modèles, le réseau est représenté par un graphe orienté. A l'instant *t=0*, un ensemble de nœuds initiateurs de la diffusion d'une nouvelle idée sont actif. A un instant *t*, si un nœud adopte la nouvelle idée, il devient actif, sinon il est inactif. On suppose qu'un nœud inactif peut passer à l'état actif mais un nœud actif le restera tout au long du processus de propagation. Un nœud actif tente d'activer ces voisins. Le processus se poursuit jusqu'à ce qu'il n'ait plus d'activations possibles.

2.1.1. Independent Cascade Model

Dans ce modèle, une probabilité $p_{u,v}$ est associé à chaque lien *(u,v)* où *u*, et *v* sont deux nœuds du réseau. $p_{u,v}$ est la probabilité que *u* réussit à activer *v* (*u influence v*). Cette probabilité peut correspondre au taux de communication entre les deux nœuds, à la proximité géographique ou se basant sur un historique de processus de propagation antécédent (par apprentissage) (Shakarian et al., 2015).

2.1.2. Linear Threshold Model

Dans ce modèle, un poids $w_{u,v}$ est affecté à chaque lien *(u,v)* tel que la somme des poids des liens entrant à *v* est inférieur à *1*. Chaque nœud *v* est doté d'un seuil θ_v. A un instant *t*, les nœuds parents de *v* qui sont actif tentent de l'activer. *v* ne sera actif que si la somme des $w_{u,v}$ (*u* est un parent de *v* actif) est supérieur au seuil θ_v. Ce modèle correspond à dire qu'un utilisateur du réseau n'adopte une idée que si une proportion de ses relations l'ont déjà adopté (Shakarian et al., 2015).

2.1.3. Modèles basés sur le sujet

Dans ce modèle, l'analyse de l'influence porte sur un sujet spécifique. Par exemple, un utilisateur peut influencer un autre utilisateur sur le choix d'un produit, mais pas ou moins facilement sur une opinion politique. Une extension des deux modèles précédents *(ICM et LTM)* en prenant en compte le sujet de l'influence a été proposée par (Barbieri, Bonchi and Manco, 2012). Dans leur nouvelle version, la probabilité $p^z_{u,v}$ associé à chaque lien *(u,v)* représente la force de l'influence de *u* sur *v* par rapport à un sujet $z \in [1\text{-}K]$.

2.2. Détection d'influenceurs dans un réseau social

Les influenceurs dans un réseau social sont importants à détecter puisqu'ils constituent l'ensemble de départ dans un processus de diffusion d'influence. Si on cherche à maximiser l'influence dans le réseau, ces influenceurs seront privilégiés et seront les premiers porteurs d'information. Par contre, si on cherche à minimiser l'influence, ils seront à protéger, à bloquer ou à solliciter pour une contre campagne par exemple.

Plusieurs métriques ont été proposées dans la littérature (Cataldi et Aufaure, 2014 ; Herzig, Mass et Roitman, 2014 ; Subbian et al., 2014 ; Sun et Ng, 2012 ; Sun et Tang, 2011 ; Weng et al., 2010). Nous définissons les deux métriques les plus référencés dans les travaux de minimisation d'influence dans les réseaux sociaux.

- *Degré d'un nœud :* nombre de voisins d'un nœud. On parle aussi de *in-degree* ou *out-degree* pour designer respectivement, le nombre de voisins avec une relation entrante ou sortante (Sun and Tang, 2011).
- *Betweenness d'un nœud* : le nombre de plus court chemins entre chaque nœud vers tous les autres nœuds traversant ce nœud (Sun and Tang, 2011).

2.3. Maximisation de l'influence dans un réseau social

Les premiers travaux faisant référence au problème de maximisation de l'influence dans un réseau social sont (Domingos et Richardson, 2001 ; 2002 ; Kempe, Kleinberg et Tardos, 2003). Ils se sont posé la réflexion suivante : si on arrive à convaincre un ensemble d'utilisateurs du réseau social à adopter un produit ou une idée et le but est d'avoir une large diffusion dans le réseau, quel serait cet ensemble d'utilisateurs ? Plus formellement : ayant un réseau social représenté par un graphe *G*, un modèle de propagation *M*, un ensemble de nœuds initiateurs de l'influence *S* et soit $\sigma_M(S)$ la fonction qui estime la propagation de l'influence. Comment sélectionner *S* pour maximiser $\sigma_M(S)$ (Kempe, Kleinberg and Tardos, 2003).

Les auteurs de (Kempe, Kleinberg et Tardos, 2003; Li et al., 2014 ; Liu et al., 2012; Nguyen et Zheng, 2013) ont proposé des algorithmes gloutons qu'ils ont comparé à des heuristiques basées sur des mesures de centralité telles que le degré et le betweenness. Leurs propositions donnent de meilleurs résultats sauf que les algorithmes gloutons sont longs en termes de temps d'exécution. Dans (Chen, Wang et Yang, 2009 ; Chen, Yuan et Zhang, 2010), les auteurs proposent des solutions plus efficace en temps d'exécution.

3. Approches de minimisation de l'influence négative dans un réseau social

Les travaux de recherche sur la minimisation de l'influence négative dans les réseaux sociaux peuvent être classés selon leur approche en trois classes : approche

basée sur le blocage de nœuds, approche basée sur le blocage de lien et approche basée sur l'influence compétitive.

3.1. Approche basée sur le blocage de nœuds

Le principe est de bloquer un ensemble minimal de nœuds pour minimiser la propagation d'influence négative dans le réseau. Les nœuds sont généralement sélectionnés parmi les nœuds les plus influenceurs.

Plus formellement, le problème est définit comme suit : soit un réseau représenté par un graphe orienté $G = (V, E)$. V est l'ensemble de nœuds, $E \subset V \times V$ est l'ensemble des liens. Nous supposons qu'une information négative est propagée dans le réseau et un ensemble initial de nœuds I est influencé. Le but est de minimiser le nombre de nœuds influencés en bloquant un ensemble S de k nœuds, $S \subseteq \{V\}$ et k est une constante donnée. La fonction objective est :

$$Minimiser\ \sigma\{I|V \setminus S\}$$

où $\sigma\{I|V \setminus S\}$ représente le nombre de nœuds influencés par I quand les nœuds de S sont bloqués (Wang et al., 2013).

Dans (Wang et al., 2013), les auteurs supposent qu'une information négative est propagée dans le réseau social et un ensemble de nœuds est infecté (influencé). Leur but est de minimiser le nombre de nœuds infectés en bloquant k nœuds parmi les non infectés. Les auteurs proposent un algorithme glouton où à chaque itération on sélectionne un nœud parmi les nœuds non infectés qui maximise la décrémentation du nombre de nœuds infectés. L'algorithme proposé donne de meilleurs résultats que si on sélectionne les top-k nœuds non infectés selon leur out degree ou betweenness.

Dans (Yao et al., 2015a), les auteurs prennent en compte le sujet de l'influence en se basant sur un modèle de propagation d'influence basé sur le sujet (Topic-aware Independent Cascade Model) (Barbieri, Bonchi and Manco, 2012). Dans ce modèle, les probabilités d'influence d'utilisateur à utilisateur dépendent du sujet. Ils ont proposé deux méthodes prenant en compte le sujet : *Topic-aware out degree* et *Topic aware betweenness*. Les résultats des méthodes proposées sont comparés aux mesures de centralité déjà connues : *out degree* et *betweenness*. Les auteurs ont obtenu de meilleurs résultats en termes de limitation de la propagation d'influence.

3.2. Approche basée sur le blocage de liens

Le principe de cette approche est de bloquer un ensemble minimal de liens pour minimiser la propagation de l'influence négative dans le réseau.

Plus formellement, le problème est définit comme suit : soit le réseau $G = (V, E)$. Nous supposons qu'une information négative est propagée dans le réseau et un ensemble initial de nœuds I est influencé. Le but est de minimiser le nombre de nœuds

influencés en bloquant un ensemble S de k liens, $S \subseteq \{V\}$ et k est une constante donnée. La fonction objective est :

$$Minimiser\ \sigma\{I|V \setminus S\}$$

où $\sigma\{I|V \setminus S\}$ représente le nombre de nœuds influencés par I quand l'ensemble de liens de S sont bloqués (Yao et al., 2015b).

(Kimura, Saito and Motoda, 2009) sont les premiers à proposer cette approche. Ils pensent que le blocage de liens est plus fondamental que le blocage de nœuds puisque le blocage d'un nœud induit le blocage de liens associés. Les auteurs proposent un algorithme glouton qu'ils comparent aux heuristiques basées sur les mesures de centralité : out degree et betweenness. L'algorithme proposé est meilleur en termes de minimisation de la propagation d'influence. Les auteurs se basent sur le modèle de propagation d'influence Independent Cascade.

Les mêmes auteurs proposent dans (Kimura, Saito and Motoda, 2008) la même solution sous le modèle de propagation d'influence Linear Threshold où chaque nœud possède un seuil représentant la proportion de ses voisins adoptant une idée pour qu'il l'adopte lui-même. L'algorithme glouton proposé donne toujours de meilleurs résultats.

(Khalil, Dilkina et Song, 2013) formalisent le problème de minimisation de propagation de l'influence en bloquant un minimum de lien en un problème d'optimisation NP-difficile. Ils prouvent théoriquement que la fonction objective est *supermodulaire*. Ce qui permet de proposer une solution approximative avec un algorithme glouton qui approche la solution optimale avec un facteur de $(1- 1/e)$. Ils proposent ensuite un algorithme glouton qu'ils comparent à différentes heuristiques basées sur les mesures de centralité connues, entre autres: out degree et betweenness. L'algorithme proposé donne de meilleurs résultats.

Dans (Yao et al., 2015b), les auteurs proposent la même solution en démarrant d'un ensemble de nœuds déjà infectés (négativement influencés). En plus de l'évaluation du degré d'influence, les auteurs évaluent le temps d'exécution de l'algorithme glouton proposé ainsi des deux heuristiques (out degree et betweenness). Comme prévu, le temps d'exécution de l'algorithme glouton est dégradé par rapport aux heuristiques testées.

3.3. Approche basée sur l'influence compétitive

Le principe est de sélectionner un ensemble minimal de nœuds qui adopteront une contre campagne afin de minimiser l'effet de l'influence négative.

Plus formellement, le problème est définit comme suit : Soient le réseau $G = (V, E)$, un ensemble S_A de nœuds qui adoptent une contre compagne, un délai de détection de la contre compagne d. Le but est de trouver un ensemble S_C de k nœuds qui diffuseront une contre information pour sauver le maximum de nœuds. La fonction objective est :

$$Maximiser\ \sigma\{S_C ; S_A, d\}$$

où $\sigma\{S_C ; S_A, d\}$ représente le nombre de nœuds influencés par S_C. (Luo et al., 2014).

Dans (Budak, Agrawal and El Abbadi, 2011), les auteurs proposent un modèle de diffusion compétitive basé sur le modèle Independent Cascade. Il modélise la diffusion de deux informations, une représente une mauvaise information, l'autre représente la bonne information (la contre campagne). Etant donné un ensemble de nœuds qui commencent la diffusion d'une mauvaise information et un délai de détection de cette information, le but est de trouver un ensemble de nœuds qui optent une contre campagne et maximise sa diffusion pour protéger le maximum de nœuds d'être atteins par la mauvaise information. Les auteurs supposent que si la bonne et la mauvaise information arrivent au même temps à un nœud, c'est la bonne information qui prend effet. Et une fois, un nœud est influencé par une bonne ou mauvaise campagne, il ne change plus de campagne. Le processus se poursuit jusqu'à ce que tous les nœuds soient influencés. Les auteurs prouvent que le problème est NP-difficile et que la fonction objective est *submodulaire*. Ils proposent ensuite un algorithme glouton. Dans un second lieu, les auteurs supposent la méconnaissance de l'ensemble de nœuds diffusant la mauvaise information et du délai de détection de cette information, ce qui est plus réaliste. Ils proposent pour cela un algorithme de prédiction basé sur l'heuristique Hill Climbing.

Dans (Nguyen et al., 2012), étant donné un réseau où une désinformation est diffusée, on vise à décontaminer un minimum de nœuds de telle sorte à ce que la décontamination totale après un temps T atteint un taux β. T et β sont des paramètres en entrées. La propagation de la décontamination suit ensuite le modèle de diffusion (IC ou LT). Un algorithme glouton est appliqué pour trouver l'ensemble de nœuds à décontaminer. L'algorithme donne de bons résultats mais il est couteux en temps d'exécution. Pour cela, les auteurs proposent une heuristique basée sur la structure en communauté du réseau social. Ils appliquent alors l'algorithme glouton sur chaque communauté pour atteindre un taux de décontamination β dans chaque communauté.

Aussi, dans (Fan et al., 2013), les auteurs se basent sur la structure en communauté d'un réseau social et le constat que les communications sont plus dense dans une même communauté qu'entre communautés. L'influence est donc plus rapidement propagée dans une même communauté. Donc, si on cherche à protéger un ensemble minimal de nœuds pour minimiser l'infection des communautés voisines, ils seraient sélectionnés parmi les nœuds qui ont des relations avec les communautés voisines. Cela évitera la contamination des communautés voisines d'une part et décontamine la communauté infecté plus rapidement.

(Luo et al., 2014) se basent sur un modèle de diffusion à temps continue où le temps de diffusion entre deux nœuds diffère selon le taux de transmission entre ces derniers. Les auteurs montre que le problème tel que modélisé est NP-Difficile et que la fonction objective est sous modulaire.

De même que les travaux précédents, les auteurs de (Fan et al., 2013; Luo et al., 2014) proposent des algorithmes gloutons qu'ils comparent à des mesures de centralité tel que le degré. Leurs algorithmes donne de meilleurs résultats en termes du nombre de nœuds influencé mais n'ont pas été testé en terme de temps d'execution.

4. Synthèse et Analyse

Le tableau 1, résume les travaux cités dans la section 3. L'accent est mis sur la principale contribution de chaque travail, le modèle de diffusion sur lequel se base la solution proposée, l'algorithme proposé, les données d'expérimentation et les résultats obtenus en termes de minimisation d'influence et de temps d'exécution.

Tableau 1. Synthèse des travaux de recherche sur la minimisation de l'influence négative dans les réseaux sociaux.

Approche	Référence	Contribution	Modèle de diffusion		Algorithmes					Données d'expérimentation			Résultats	
					proposés		Testé avec							
			ICM	LTM	Glouton	autres	degré	betweenness	autres	Réseau	Nombre de nœuds	Nombre de liens	Minimisation de l'influence	Temps d'exécution
Bloquer des nœuds	(Wang et al., 2013)	Bloquer k nœuds parmi les nœuds non infectés	x		x		x	x	x	Enron email communication network	36692	367662	+	
	(Yao et al., 2015a)	prend en compte le sujet de l'influence.	x			x	x	x		Sina microblog	2000	14426	+	
										Facebook	4039	88234		
Bloquer des liens	(Kimura, Saito and Motoda, 2009)	Bloquer k lien pour minimiser la contamination par l'influence négative (initiateur de l'approche).	x		x		x	x	x	blog network	12047	79920	+	
										Japanese Wikipedia network	9481	245044		
	Saito and Mo-	Adaptation du travail précédent sous le modèle de		x	x		x	x		blog network	12047	79920	+	

		diffusion Linear Threshold							Japanese Wikipedia network	9481	245044		
	(Khalil, Dilkina et Song, 2013)	Formalisation et preuve théorique que la fonction objective est super modulaire		x	x		x	x	x	FOREST-FIRE network	500	1691	+	
										Meme-Tracker network				
	(Yao et al., 2015b)	Evaluation du temps d'exécution	x		x		x	x		Facebook data set	4039	88234	+	-
										Digger data set	8193	56440		
Influence compétitive	(Budak, Agrawal and El Abbadi, 2011)	La contre campagne est lancé après un délai r, délai de détection d'une campagne de désinformation	x		x	x	x		x	2009 Santa Barbara network	26455	53132	+	
										2008 Santa Barbara network	12814	184482		
										2008 the Monterey Bay network	6117	62750		
	(Nguyen et al., 2012)	Protection de nœuds : Décontaminer k nœuds les plus influenceurs	x	x	x		x		x	NetHEPT network	15233	31398	+	-
										Facebook	63000	1 .5 million		
	(Luo et al., 2014)	Sauver le maximum de nœuds avant un délai donné. Prend en compte le taux de transmission entre deux nœuds.			x		x		x	Facebook-like social network	1899	20296	+	
	(Fan et al., 2013)	Minimiser l'infection des communautés voisines. Les nœuds protec-			x		x		x	Enroll Email communication network	36692	367662	+	

		teurs sont définis parmi les nœuds qui ont des relations avec les communautés voisines.												
									e-print arXiv *Collaboration Network*	15233	58891			

En analysant le tableau 1, nous tirons les conclusions suivantes :

- La problématique de minimisation d'influence négative dans un réseau social est assez récente. Les premiers travaux publiés datent de 2008. Cependant, la problématique n'a pas attiré beaucoup l'attention de la communauté de recherche. Les travaux s'orientent plutôt vers l'influence positive et sa maximisation dans les réseaux sociaux.
- Parmi les trois approches proposées, les deux approches consistant à bloquer des nœuds ou des liens ne peuvent être appliquées que par un administrateur du réseau social qui possède ce privilège de blocage d'utilisateurs ou de relations. Ceci limitera l'application de ces approches par un responsable marketing, par exemple, qui veut limiter la diffusion d'une contre publicité
- Dans les trois approches (blocage de nœuds, blocage de lien, influence compétitive), le problème est modélisé comme un problème d'optimisation NP-difficile. Il s'agit dans tout les cas, de trouver un minimum de nœuds/ liens à bloquer ou un minimum de nœuds qui lanceront une campagne compétitive afin de minimiser le nombre de nœuds influencé négativement. Il est prouvé que la fonction objective est *sous*/*super modulaire*. Ceci garantit qu'une solution avec un algorithme glouton donne des résultats qui approchent la valeur optimale de (1-1/e), soit 63%. La plupart des travaux ont donc opté pour une solution avec un algorithme glouton. Cependant, ces algorithmes sont très longs en termes de temps d'exécution.
- La majorité des travaux comparent les résultats obtenus à des heuristiques basées sur des mesures très utilisées telles que le degré et le betweenness. L'algorithme glouton donne toujours de meilleurs résultats en termes de minimisation du nombre de nœuds influencés négativement. Peu de travaux ont fait une évaluation en termes de temps d'exécution. Ces derniers montrent que les algorithmes gloutons ne sont pas performants en comparaison avec les heuristiques citées précédemment.
- Les travaux proposés ne comparent pas leurs résultats aux travaux qui les précèdent. On n'a donc pas une idée sur la meilleure solution proposée

jusque là, ni en terme de minimisation d'influence ni en terme de temps d'exécution.

– Les données d'expérimentation sont généralement des données issues de réseaux sociaux réels. Seulement, la taille des réseaux testés reste limitée. Elle varie entre 500 et 63 000 nœuds tandis que les réseaux actuels comprennent des centaines de millions de nœuds.

5. Ouvertures de recherche

Après étude des différents travaux liés à la minimisation de l'influence négative dans les réseaux sociaux, nous avons pu dresser quelques pistes de recherche encore ouvertes :

– Les solutions proposées se sont fixé comme seul objectif, la minimisation de l'influence négative en négligeant le temps d'exécution, un paramètre très important dans les réseaux sociaux. Les algorithmes proposés sont des algorithmes gloutons qui donnent des résultats qui approchent la valeur optimale mais qui sont longs en termes de temps d'exécution. Il serait donc intéressant de proposer des solutions basées sur des heuristiques (plus rapide) au lieu d'algorithmes gloutons pour atteindre un compromis entre degré d'influence et temps d'exécution. Cela permettra aussi de faire des tests sur des réseaux plus larges approchant la taille des réseaux réels tels que Facebook.

– L'évaluation de l'influence négative a été exprimée, dans la plus part des travaux, par le nombre de nœuds infectés. Ce nombre peut ne pas être significatif si ces nœuds, par exemple, ont très peu de relations ou tout simplement n'ont rien à perdre s'ils sont infectés. (Luo et al., 2014) proposent, comme perspective, de donner un coût de perte à chaque nœud s'il sera infecté et au lieu de minimiser le nombre de nœuds infectés, on minimise le coût de perte dans le réseau.

– Plusieurs améliorations des travaux existant sont possibles :

- Prendre en compte l'aspect dynamique dans le réseau au cours de la diffusion d'information :
  - *Changement de topologie :* lors de la propagation de l'influence, des nœuds peuvent quitter le réseau (même parmi les influenceurs) et d'autres peuvent arriver et qu'on ne souhaitera pas qu'ils soient atteints dès leur arrivée.
  - *Changement d'opinion d'un nœud :* les modèles de propagation d'influence utilisés (ICM ou LTM) considèrent que si un nœud est influencé, il ne changera pas d'avis jusqu'à la fin du processus de propagation. Nous pouvons améliorer ces modèles pour mieux correspondre à la réalité où un utilisateur peut changer d'avis à n'importe quel moment.
- *Cibler une communauté :* dans certains cas, l'influence négative est propagée pour atteindre une communauté bien précise sur un sujet

bien précis. Il sera donc intéressant, dans ce cas, de prendre en compte ces deux aspects pour atteindre l'objectif en un temps réduit.

– Une question importante reste ouverte : en réalité, comment pourrions-nous convaincre un influenceur (probablement une star) pour adopter la bonne information et la diffuser dans le réseau social (Nguyen et al., 2012).

6. Conclusion

L'influence dans les réseaux sociaux est un sujet en plein essor. La problématique de minimisation de l'influence négative, par ses applications et ses avantages apportés aux réseaux sociaux, suscite l'intérêt de la communauté de recherche. Néanmoins, les travaux proposés manquent d'efficacité. Ils se sont focalisés sur le seul objectif de minimiser l'influence négative en proposant des algorithmes glouton long en termes de temps d'exécution.

Nous comptons traiter ce problème dans nos futurs travaux. Nous visons à proposer une solution algorithmique basée sur des méta-heuristiques au lieu d'algorithmes glouton pour minimiser la propagation d'influence en se fixant deux objectifs à la fois : minimiser le taux d'influence négatif dans le réseau et minimiser le temps d'exécution des algorithmes s'exécutant sur le réseau social (large échelle).

7. Bibliographie

Barbieri, N., Bonchi, F. and Manco, G. (2012). Topic-Aware Social Influence Propagation Models.*2012 IEEE 12th International Conference on Data Mining.*

Baumeister, R.F., Bratslavsky, E., Finkenauer, C., Vohs, K.D., 2001. Bad is stronger than good. Review of General Psychology 5, 323–370.

Budak, C., Agrawal, D. and El Abbadi, A. (2011). Limiting the spread of misinformation in social networks. *Proceedings of the 20th international conference on World wide web - WWW '11*.

Cataldi, M. and Aufaure, M. (2014). The 10 million follower fallacy: audience size does not prove domain-influence on Twitter. *Knowledge and Information Systems*, 44(3), pp.559-580.

Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincon, D., Sun, X., Wang, Y., Wei, W., Yuan, Y., (2010). Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate (No. MSR-TR-2010-137).

Chen, W., Wang, Y. and Yang, S. (2009). Efficient influence maximization in social networks.*Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09.*

Chen, W., Yuan, Y. and Zhang, L. (2010). Scalable Influence Maximization in Social Networks under the Linear Threshold Model. *2010 IEEE International Conference on Data Mining*.

Domingos, P. and Richardson, M. (2001). Mining the network value of customers. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*.

Fan, L., Lu, Z., Wu, W., Thuraisingham, B., Ma, H. and Bi, Y. (2013). Least Cost Rumor Blocking in Social Networks. *2013 IEEE 33rd International Conference on Distributed Computing Systems*.

Herzig, J., Mass, Y. and Roitman, H. (2014). An author-reader influence model for detecting topic-based influencers in social media. *Proceedings of the 25th ACM conference on Hypertext and social media - HT '14*.

Kempe, D., Kleinberg, J. and Tardos, É. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*.

Khalil, E.; Dilkina, B.; and Song, L. (2013), CuttingEdge: Influence minimization in networks. In *Workshop on Frontiers of Network Analysis: Methods, Models, and Applications at NIPS*, 2013**.**

Kimura, M., Saito, K. and Motoda, H. (2009). Blocking links to minimize contamination spread in a social network. *ACM Trans. Knowl. Discov. Data*, 3(2), pp.1-23.

Kimura, M., Saito, K. and Motoda, H. (n.d.). Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model. *PRICAI 2008: Trends in Artificial Intelligence*, pp.977-984.

Li, N. and Gillet, D. (2013). Identifying influential scholars in academic social media platforms.*Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*.

Li, S., Zhu, Y., Li, D., Kim, D., Ma, H. and Huang, H. (2014). Influence maximization in social networks with user attitude modification. *2014 IEEE International Conference on Communications (ICC)*.

Liu, B., Cong, G., Xu, D. and Zeng, Y. (2012). Time Constrained Influence Maximization in Social Networks. *2012 IEEE 12th International Conference on Data Mining*.

Luo, C., Cui, K., Zheng, X. and Zeng, D. (2014). Time Critical Disinformation Influence Minimization in Online Social Networks. *2014 IEEE Joint Intelligence and Security Informatics Conference*.

Nguyen, H. and Zheng, R. (2013). On Budgeted Influence Maximization in Social Networks. *IEEE J. Select. Areas Commun.*, 31(6), pp.1084-1094.

Nguyen, N., Yan, G., Thai, M. and Eidenbenz, S. (2012). Containment of misinformation spread in online social networks. *Proceedings of the 3rd Annual ACM Web Science Conference on - WebSci '12*.

Richardson, M. and Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing.*Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*.

Shakarian, P., Bhatnagar, A., Aleali, A., Shaabani, E. and Guo, R. (2015). The Independent Cascade and Linear Threshold Models. *SpringerBriefs in Computer Science*, pp.35-48.

Subbian, K., Sharma, D., Wen, Z. and Srivastava, J. (2014). Finding influencers in networks using social capital. *Soc. Netw. Anal. Min.*, 4(1).

Sun, B. and Ng, V. (2012). Identifying influential users by their postings in social networks.*Proceedings of the 3rd international workshop on Modeling social media - MSM '12*.

Sun, J. and Tang, J. (2011). A Survey of Models and Algorithms for Social Influence Analysis. *Social Network Data Analytics*, pp.177-214.

Wang, F., Camacho, E. and Xu, K. (2009). Positive Influence Dominating Set in Online Social Networks. *Combinatorial Optimization and Applications*, pp.313-321.

Wang, S.; Zhao, X.; Chen, Y.; Li, Z.; Zhang, K. & Xia, J. (2013), Negative Influence Minimizing by Blocking Nodes in Social Networks., *in* 'AAAI 2013.

Weng, J., Lim, E., Jiang, J. and He, Q. (2010). TwitterRank: Finding Topic-sensitive Influential Twitterers. *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*.

Yao, Q., Shi, R., Zhou, C., Wang, P. and Guo, L. (2015a). Topic-aware Social Influence Minimization. *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion.*

Yao, Q., Zhou, C., Xiang, L., Cao, Y. and Guo, L. (2015b). Minimizing the Negative Influence by Blocking Links in Social Networks. *Trustworthy Computing and Services*, pp.65-73.

Évaluation de l'influence dans un réseau multi-relationnel : le cas de Twitter

Lobna Azaza [1,2], **Sergey Kirgizov** [1], **Marinette Savonnet** [1], **Éric Leclercq** [1], **Rim Faiz** [2]

1. Laboratoire LE2I - UMR 6306 - CNRS - ENSAM
Univ. Bourgogne Franche-Comté
9, Avenue Alain Savary
F-21078 Dijon - France
Prenom.Nom@u-bourgogne.fr

2. Laboratoire Larodec
Université de Carthage
Tunis, Tunisie
Rim.Faiz@ihec.rnu.tn

RÉSUMÉ. *L'influence sur Twitter est devenue un sujet de recherche important. Certains utilisateurs révèlent plus de capacité que d'autres pour influencer les personnes avec lesquelles ils sont connectés. Ainsi, trouver les utilisateurs les plus influents peut permettre une diffusion efficace de l'information à grande échelle, action très utile dans le marketing ou les campagnes politiques. Dans cet article, nous proposons une nouvelle approche pour l'évaluation de l'influence dans les réseaux multi-relationnels tels que Twitter. Notre méthode est basée sur la règle de combinaison conjonctive de la théorie des fonctions de croyance qui permet de fusionner différents types de relations. Nous expérimentons notre méthode sur des données Twitter collectées lors des élections européennes de 2014 et déterminons les candidats les plus influents.*

ABSTRACT. *Influence in Twitter has become recently a hot research topic. Some users are more able than others to influence peers. Thus, studying most influential users leads to reach a large-scale information diffusion area, something very useful in marketing or political campaigns. In this paper, we propose a new approach for influence assessment on multi-relationnal networks such as Twitter. This is based on the conjonctive combination rule in belief functions theory in order to combine different types of interactions. We experiment the proposed method on a large amount of data gathered from Twitter in the context of the European Elections 2014 and deduce top influential candidates.*

MOTS-CLÉS : *Réseau multi-relationnel, Influence, Fusion d'information, Fonctions de croyance.*

KEYWORDS: *multirelational network, Influence, Information fusion, Belief theory.*

Congrès INFORSID, 34ème édition, Grenoble, 31 mai – 3 juin 2016

1. Introduction

Les réseaux sociaux en ligne tel que *Twitter* rassemblent les personnes et renforcent leurs relations avec de nouvelles formes de coopération et de communication. En raison de son immense popularité, *Twitter* est exploité comme une plate-forme pour le marketing et les campagnes politiques. L'une des caractéristiques de *Twitter* est la diffusion de l'information à travers les liens sociaux. Les liens entre les utilisateurs déterminent le flux de l'information et indiquent ainsi l'influence d'un utilisateur sur un autre. Certains utilisateurs, appelés influents, sont plus capables que d'autres de diffuser des informations à un grand nombre d'utilisateurs. Par conséquent, la détection des utilisateurs influents dans un réseau est une clé du succès pour parvenir à une diffusion d'information à large échelle et à faible coût.

L'influence sur *Twitter* est définie comme la capacité d'un utilisateur à provoquer une action chez un autre utilisateur (Leavitt *et al.*, 2009). Le terme "action"signifie les différentes interactions possibles entre les utilisateurs. Par conséquent, la mesure de l'influence sur *Twitter* n'est pas aussi simple puisque *Twitter* offre plusieurs relations (*retweet*, *réponse*, *mention*, *suivre*) et donc plusieurs formes d'interactions. Un utilisateur peut *suivre* un autre utilisateur, ce qui lui permet de voir les *tweets* et les informations de l'utilisateur qu'il suit. Il est également capable de *retweeter* un *tweet*, ce qui expose ce *tweet* à ses abonnés, qui peuvent aussi le *retweeter*. Un utilisateur peut *mentionner* un autre utilisateur en utilisant le préfixe "@"s'il veut lui adresser le *tweet*, ce même *tweet* pouvant être *retweeté* par un autre utilisateur. Enfin, un utilisateur peut *répondre* à un *tweet* et créer ainsi une conversation avec l'utilisateur du tweet intital. Ces différents types d'interactions sont ce qui fait de *Twitter* un réseau multi-relationnel (Wu *et al.*, 2013; Rodriguez, Shinavier, 2010).

L'évaluation de l'influence pose trois principaux défis. Le premier est la diversité des interactions sur lesquels nous pouvons baser les calculs de l'influence. Il est important de les combiner afin d'établir une mesure générale d'influence qui prend en compte les différents types d'interactions entre les utilisateurs. Le second défi est la considération de l'influence indirecte. L'influence est indirecte lorsqu'elle atteint un utilisateur à travers des utilisateurs intermédiaires. Par exemple, un utilisateur peut *retweeter* un *tweet* d'un autre utilisateur indirectement à travers un utilisateur intermédiaire. Il est donc nécessaire de mesurer l'influence en tenant compte des interactions directes et indirectes dans le réseau. Le troisième défi est relatif à l'incertitude lors de la combinaison d'interactions. Dans le cas des réseaux multi-relationnels, il est difficile d'attribuer des pondérations valuées aux différentes interactions avant de fusionner les données quantitatives.

Dans ce papier, pour mesurer l'influence d'un utilisateur, nous combinons, en tenant compte de l'incertitude dans le processus de la mesure, différentes interactions définies par des experts du domaine étudié (la communication politique dans notre cas). La mesure peut être établie entre un couple d'utilisateurs en tenant compte des différentes interactions entre eux deux, mais aussi étendue à une mesure d'influence globale d'un utilisateur dans le réseau.Pour cela, nous définissons un cadre théorique

sur la base de la règle de combinaison conjonctive de la théorie des fonctions de croyance et la règle de Smets (Smets, 1997) pour la fusion et la combinaison des informations. L'approche proposée est flexible et l'influence indirecte dans le graphe multi-relationnel peut être prise en considération. Une évaluation à travers des expérimentations est proposée, elle est basée sur des données *Twitter* collectées dans le cadre du projet TEE 2014 lors de la campagne pour les élections européennes de 2014.

Le reste de l'article est organisé comme suit. La section 2 présente un état de l'art. La section 3 décrit notre approche. La section 4 présente les résultats expérimentaux. Et enfin la section 5 conclut le papier.

2. État de l'art

Dans cette section, nous présentons des travaux sur l'évaluation de l'influence dans *Twitter* et rappelons les concepts de base de la théorie des fonctions de croyance sur lesquels se fonde notre approche.

2.1. L'influence dans Twitter

Dans la littérature, plusieurs approches ont été proposées pour classer les utilisateurs selon leur influence. Certaines approches sont basées sur la **topologie du réseau** et les mesures de centralité. D'autres approches ont établi un classement des utilisateurs en utilisant des algorithmes basés sur la **diffusion**. Une autre famille étend les approches topologiques pour assurer la **fusion d'information**. Dans la suite, nous présentons les travaux principaux pour chaque type d'approches.

Pour mesurer l'influence dans *Twitter*, de nombreux critères peuvent être pris en considération. Les auteurs dans (Leavitt *et al.*, 2009) utilisent trois caractéristiques pour mesurer l'influence : le nombre des *réponses*, *retweets* et *mentions*, en plus du nombre d'*abonnés*. Ils donnent des statistiques relatives à ces mesures mais ne proposent pas un score global de l'influence se basant sur toutes les relations prises en compte. (Cha *et al.*, 2010) utilisent les critères nombre d'*abonnés*, de *retweets* et de *mentions*. Ils calculent la valeur de chaque mesure d'influence pour 6 millions d'utilisateurs puis ils les comparent. Pour ce faire, ils trient les utilisateurs en fonction de chaque relation, puis, ils quantifient comment le classement d'un utilisateur varie selon les différentes relations. La corrélation de Spearman est utilisée comme une mesure de la force d'association entre deux ensembles du classement. Ils ont constaté que le nombre d'abonnés représente la popularité d'un utilisateur, mais il n'est pas lié à d'autres relations telles que les *retweets* et les *mentions*. Leur résultat suggère que le nombre d'*abonnés* seul révèle très peu sur l'influence d'un utilisateur. De même, cette méthode ne fournit pas une mesure globale de l'influence.

(Chen *et al.*, 2013) proposent une méthode de classement local, nommée Cluster Rank, prenant en considération le nombre de voisins et leur coefficient de cluster-

ing[1]. (Bakshy *et al.*, 2011) ont suivi une approche différente pour estimer les utilisateurs influents : ils utilisent les cascades de diffusion d'URL raccourcis et considérent que les utilisateurs qui produisent les cascades les plus longues sont les plus influents. (Brown, Feng, 2011) pensent que la localisation d'un nœud dans le réseau peut déterminer son influence. Considérant ce fait, l'algorithme de décomposition *k-shell* peut être utilisé (Seidman, 1983). Son principe est d'attribuer un indice de référence ks pour chaque nœud tel que les nœuds ayant les valeurs les plus faibles sont situés à la périphérie du réseau tandis que les nœuds avec les valeurs les plus élevées se trouvent au centre du réseau, ce sont ces nœuds qui auront le plus d'influence. Les auteurs ont adapté l'algorithme de décomposition *k-shell* aux caractéristiques du réseau *Twitter*. Qasem *et al.* (Qasem *et al.*, 2015) présentent une nouvelle approche pour la détection des utilisateurs influents. L'approche proposée détecte les utilisateurs qui augmentent la taille du réseau social en attirant de nouveaux utilisateurs dans le réseau.

L'inconvénient des algorithmes basés sur la topologie du réseau est de ne considérer que les informations de l'utilisateur, sans considérer l'interaction entre les utilisateurs à travers les séquences des relations. Avec *Twitter*, l'influence d'un utilisateur est impactée par la diffusion de l'information entre les utilisateurs.

D'autres recherches proposent de classer les utilisateurs en utilisant des algorithmes basés sur la **diffusion**, avec l'hypothèse commune selon laquelle un utilisateur est influent s'il pointe vers de nombreux voisins très influents. Dans (Weng *et al.*, 2010), les auteurs proposent TwitterRank, une extension de l'algorithme PageRank (Page *et al.*, 1999), afin de mesurer l'influence des utilisateurs en tenant compte des sujets associés aux tweets. Bien que l'idée soit prometteuse, les résultats expérimentaux montrent qu'il y a des utilisateurs qui *suivent* d'autres utilisateurs sans présence de similarité de sujets entre eux et leurs amis. La méthode a ignoré d'autres critères importants tels que les *mentions* et les *réponses*. Romero *et al.* (Romero *et al.*, 2011) proposent le IP-Algorithme basé sur l'algorithme HITS (Kleinberg, 1999). Les auteurs considérent l'influence comme le niveau de propagation du contenu dans le réseau (*retweets*). De plus, les auteurs estiment que l'influence d'un utilisateur ne dépend pas seulement de la taille de son audience, mais aussi de sa passivité. La passivité d'un utilisateur est le fait qu'il ne transmet pas l'information au réseau. L'algorithme a montré une meilleure précision que d'autres mesures d'influence tels que PageRank, le nombre d'abonnés et le nombre de *mentions*. Bien que la passivité semble être un facteur à prendre en compte dans le calcul de l'influence, ce travail a ignoré d'autres relations importantes telles que la *réponse*. Ashwini *et al.* (Ashwini, M.R., 2015) considèrent que *Twitter* est une plate-forme de diffusion d'information et étudient le problème de l'identification des utilisateurs influents. Ils proposent ProfileRank, un modèle de diffusion d'information basé sur la marche aléatoire qui estime l'influence des utilisateurs et la pertinence du contenu. ProfileRank est fondé sur le principe qu'un utilisateur influent crée du contenu pertinent. La limite de cette ap-

1. En théorie des graphes, le coeffcient de clustering mesure à quel point les voisins d'un sommet sont connectés.

proche est que l'influence est estimée en se basant seulement sur la relation *retweet* et la méthode ignore d'autres relations importantes.

Dans des travaux récents, la **fusion d'information** est considérée afin de contourner les limitations des méthodes existantes. Dans (Simmie *et al.*, 2013), les auteurs proposent la combinaison de deux modèles pour classer les utilisateurs influents : l'algorithme PageRank et HMC (Modèle de Markov Caché). Ils ont construit un HMM pour observer l'évolution de l'influence à travers le temps et utilisent les trois relations *retweet*, *mention* et *réponse*. Le modèle est évalué sur une enquête considérée comme une réalité du terrain. Le modèle proposé diffère des autres par la combinaison de trois relations. Toutefois, puisque le but est de classer l'influence des utilisateurs, l'influence d'un utilisateur donné ne révèle pas d'informations sur son degré d'influence (forte ou faible influence), le résultat du modèle est utile uniquement pour le classement des utilisateurs.

Aucun travail de recherche existant ne prend en compte la combinaison de plusieurs relations avec de l'incertitude. Or, il nous parait important, pour mesurer l'influence, de tenir compte des degrés d'incertitude sur les poids attribués aux différentes interactions selon leur importance. Dans cet objectif, nous proposons l'utilisation de la théorie des fonctions de croyance. Dans des recherches récentes, la théorie des fonctions de croyance est exploitée pour mesurer l'influence dans des réseaux pondérés (Cai *et al.*, 2013; Wei *et al.*, 2013) et complexes (Mo *et al.*, 2015) avec l'objectif commun de modifier les mesures de centralité existantes. Au meilleur de notre connaissance, ceci est la première fois que la théorie des fonctions de croyance est exploitée pour mesurer l'influence sur le réseau *Twitter* avec des patterns d'interactions au lieu des mesures de centralité.

2.2. Théorie des fonctions de croyance

La théorie des fonctions de croyance est considérée comme un outil général pour le raisonnement avec incertitude, et a été reliée à d'autres cadres tels que les théories des probabilités, des possibilités et des probabilités imprécises (Denoeux, Masson, 2012). La théorie des fonctions de croyance, aussi connue comme la théorie de l'évidence ou théorie de Dempster-Shafer, a été d'abord introduite par A. Dempster dans le contexte de l'inférence statistique, et a été developpée plus tard par G. Shafer comme un outil général pour la modélisation de l'incertitude épistémique (Kotz, N. L. Johnson eds., 1982).

Dans les paragraphes suivants, nous allons rappeler les concepts de base de la théorie des fonctions de croyance. Soient Ω un ensemble fini et 2^{Ω} l'ensemble de tous les sous-ensembles de Ω. Dans le contexte de la théorie de Dempster-Shafer, Ω est souvent appelée un cadre de discernement. La masse m est une fonction $m : 2^{\Omega} \rightarrow [0, 1]$ tel que :

$$\sum_{X \in 2^{\Omega}} m(X) = 1 \text{ and } m(\emptyset) = 0 \tag{1}$$

La masse $m(X)$ exprime la part de la croyance qui supporte le sous-ensemble X de Ω, $m(\emptyset) = 0$ car nous considérons que le cadre de discernement est exhaustif et exclusif (hypothèse du monde clos).

La théorie des fonctions de croyance permet, non seulement la représentation de la connaissance partielle, mais aussi la fusion de l'information (Nimier, Appriou, 1995). La fusion d'information est réalisée par la règle de combinaison conjonctive (Smets, 1997), elle suppose que toutes les sources sont fiables et consistantes. Considérant deux fonctions de masse m_1 et m_2, la règle de combinaison conjonctive est définie par :

$$(m_1 \textcircled{\cap} m_2)(C) = \sum_{A \cap B = C} m_1(A) m_2(B), \quad A, B, C \in 2^{\Omega} \tag{2}$$

Afin de prendre une décision, nous essayons de sélectionner l'hypothèse la plus probable, ce qui peut être difficile à réaliser directement avec les bases de la théorie des fonctions de croyance où les fonctions de masse sont données, non seulement pour les singletons, mais aussi pour les sous-ensembles du cadre de discernement. Ils existent plusieurs solutions pour assurer la prise de décision au sein de la théorie des fonctions de croyance, la plus connue est la probabilité pignistique (Smets, 1989). Contrairement aux fonctions de masse qui sont définies sur 2^{Ω}, la probabilité pignistique est une mesure de probabilité définie sur Ω. La probabilité pignistique a été proposée dans le modèle des croyances transférables (Smets, Kennes, 2008). Elle est basée sur deux niveaux : le "niveau crédal"où les croyances sont représentées par des fonctions de croyance et le "niveau pignistique"où les croyances sont utilisées pour prendre la décision et représentées comme des fonctions de probabilité appelées probabilités pignistiques et notées *bet* définies par :

$$\mathrm{bet}(x) = \sum_{x \in X \subseteq \Omega} \frac{m(X)}{|X|} \tag{3}$$

3. Approche proposée

L'objectif de notre approche est de trouver des critères de manifestation de l'influence et de pondérer ces critères. Afin de mesurer l'influence d'un utilisateur, nous utilisons la théorie des fonctions de croyance pour effectuer la fusion des informations issues des différentes formes d'interactions (patterns d'interaction directs ou indirects). La figure 1 donne un aperçu des différentes étapes de l'approche proposée. D'abord, l'information de *Twitter* est modélisée dans un graphe en sélectionnant les relations et les patterns pertinents puis le choix des degrés d'influence et l'initialisation des masses de croyance sont réalisés, cette étape dépend du domaine étudié. Dans le niveau crédal, nous associons les différentes fonctions de masse à chaque relation et pattern puis nous les combinons pour obtenir la masse de croyance de l'influence. Dans le niveau pignistique, nous calculons la probabilité pignistique afin de prendre la décision sur

le degré d'influence d'un nœud. Dans cette section, nous détaillons chaque étape du processus de l'évaluation.

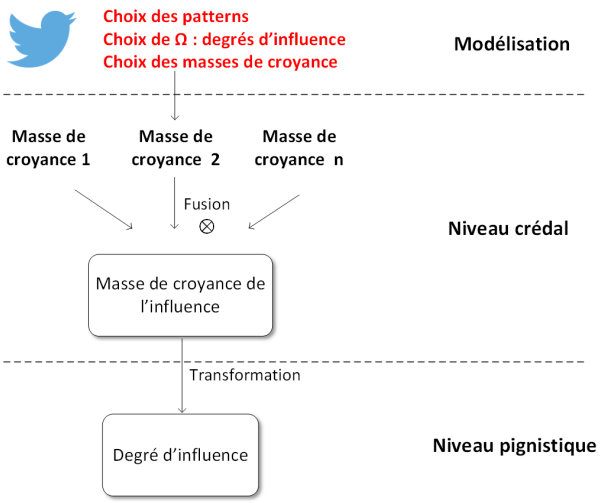


Figure 1. Étapes de l'approche proposée

3.1. Graphe de l'influence

Les réseaux sociaux sont généralement modélisés comme un graphe (Barnes, 1969) représenté par $G = (V, E)$ comprenant un ensemble V de sommets ou nœuds et un ensemble E d'arcs ou de liens. Dans le réseau *Twitter*, le graphe est hétérogène puisque nous avons différentes relations entre les nœuds et différents types de nœuds. Par exemple, il peut exister une relation *Suivre* entre deux utilisateurs, une relation *Retweet* entre un *tweet* et un utilisateur. Afin de modéliser cette hétérogénéité, un graphe multi-relationnel peut être utilisé (Rodriguez, Shinavier, 2010). Comme nous souhaitons évaluer l'influence d'un utilisateur sur d'autres, nous limitons le graphe à des nœuds homogènes (les utilisateurs), ainsi, nous travaillons sur un graphe multiplexe (Kanawati, 2015). Dans un graphe multiplexe, l'ensemble des liens E est divisé en classes disjointes : $E = \bigcup_{r \in R} E_r$, où R est l'ensemble de types de relations possibles. Par exemple, dans *Twitter* nous pouvons considérer :

R = {*Retweet*, *Mention*, *Réponse*, *Suivre*}

Nous définissons un pattern d'interaction p comme une séquence de relations, par exemple un *Retweet* d'une *Réponse* ou *Retweet* d'un *tweet* avec une *Mention*. Soit P l'ensemble des patterns d'interaction qui ont été identifiés pour modéliser l'influence dans un domaine spécifique, cet ensemble peut être donné par les chercheurs en sciences sociales par exemple. Notons par $R = R \bigcup P$ l'ensemble des relations y compris les patterns d'interaction.

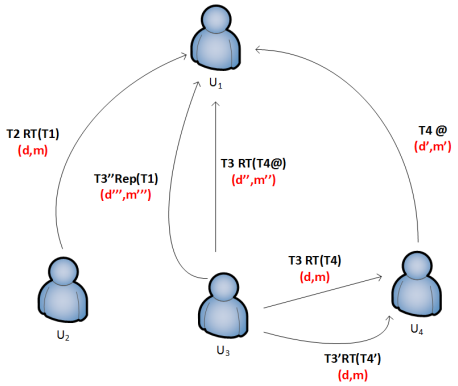


Figure 2. Graphe de l'influence

Dans ce contexte, nous introduisons le graphe de l'influence (Figure 2) comme un graphe multiplexe étiqueté. Les nœuds représentent les utilisateurs, et les liens sont les différentes relations entre eux. Par exemple, le lien entre u_4 et u_1 signifie que le *tweet* $T4$ émis par u_4 mentionne u_1, le lien étiqueté $T3RT(T4@)$ correspond au *tweet* $T3$ de u_3 qui est un *Retweet* (RT) du *tweet* $T4$ de u_4 qui mentionnait u_1, ce qui représente le pattern d'interaction *retweet* d'une *mention* entre les utilisateurs u_3 et u_1 effectué à travers l'utilisateur u_4. Nous trouvons d'ailleurs dans le graphe le tweet $T3$ de u_3 vers u_4. Les liens sont aussi étiquetés avec les degrés d'influence d (par exemple, Faible, Moyenne, Forte) et les masses de croyance m qui dépendent du type de la relation.

3.2. Fusion des masses dans le graphe de l'influence

En se basant sur la théorie de Dempster-Shafer discutée dans la section 2, nous fusionnons différentes fonctions de masse définies dans le graphe multiplexe.

Soit Ω un ensemble ordonné des degrés d'influence possibles :

$$\Omega = \{\text{Très Faible}, \text{Faible}, \text{Assez Moyenne}, \text{Moyenne}, \text{Assez forte}, \text{Forte}, \text{Très Forte}, \text{Extrêmement Forte}\} \tag{4}$$

Dans la théorie générale de Dempster-Shafer, 2^{Ω} est utilisé comme domaine des fonctions de masse, dans notre approche, nous utilisons seulement un sous-ensemble Λ de 2^{Ω}, précisément :

$$\Lambda = \{\text{Très Faible}, \text{Faible}, \text{Assez Moyenne}, \text{Moyenne}, \text{Assez Forte}, \text{Forte}, \text{Très Forte}, \text{Extrêmement Forte}, \Omega\} \tag{5}$$

Alors, les fonctions de masse sont définies comme suit : $m : \Lambda \rightarrow [0, 1]$

Table 1. Definition of the operation ⓐ

ⓐ	**T.Faible**	**Faible**	**A.Moyenne**	**Moyenne**	**A.Forte**	**Forte**	**T.Forte**	**E.Forte**	**Ω**
T.Faible	Faible	A.Moyenne	Moyenne	A.Forte	Forte	T.Forte	T.Forte	E.Forte	T.Faible
Faible	A.Moyenne	A.Moyenne	Moyenne	A.Forte	Forte	T.Forte	T.Forte	E.Forte	Faible
A.Moyenne	Moyenne	Moyenne	A.Forte	Forte	T.Forte	T.Forte	T.Forte	E.Forte	A.Moyenne
Moyenne	A.Forte	A.Forte	Forte	Forte	T.Forte	T.Forte	T.Forte	E.Forte	Moyenne
A.Forte	Forte	Forte	T.Forte	T.Forte	T.Forte	T.Forte	T.Forte	E.Forte	A.Forte
Forte	T.Forte	T.Forte	T.Forte	T.Forte	T.Forte	E.Forte	E.Forte	E.Forte	Forte
T.Forte	T.Forte	T.Forte	T.Forte	T.Forte	T.Forte	E.Forte	E.Forte	E.Forte	T.Forte
E.Forte	E.Forte	E.Forte	E.Forte	E.Forte	E.Forte	E.Forte	E.Forte	E.Forte	E.Forte
Ω	T.Faible	Faible	A.Moyenne	Moyenne	A.Forte	Forte	T.Forte	E.Forte	Ω

Pour chaque type de relation $r \in R$, une fonction de masse m_r est associée. Afin d'estimer le degré d'influence d'un nœud spécifique u, nous prenons en compte la structure locale du graphe de l'influence autour du nœud u et nous combinons les fonctions de masses de croyance des liens incidents en utilisant une version modifiée de la règle de combinaison conjonctive (2) :

$$(m \otimes m')(z) = \sum_{y \,ⓐ\, x = z} m(x)m'(y), \quad x, y, z \in \Lambda \tag{6}$$

ⓐ est une opération symétrique ⓐ : $\Lambda \times \Lambda \to \Lambda$, le tableau 1 représente un exemple de l'opération ⓐ. Cette fonction assure notre hypothèse : plus nous combinons des relations relatives à un utilisateur, plus il prend de l'importance en influence.

Comme plusieurs relations peuvent exister entre le nœud u et ses voisins, nous désignons par I_r l'ensemble de tous les liens ayant le type de relation r et nous obtenons l'ensemble des fonctions de masse suivant : $\{m_{r,i} : r \in R, i \in I_r\}$. Nous combinons les fonctions de masse afin d'obtenir une masse de croyance globale correspondante au degré d'influence du nœud u. L'ordre des combinaisons pouvant affecter nos résultats, nous devons choisir un ordre pour être consistant. Afin de simplifier les expressions nous écrivons $\bigotimes_{i \in \{1,2,3\}}$ au lieu de $m_1 \otimes m_2 \otimes m_3$. Ainsi, nous considérons l'ordre des combinaisons suivant :

1. Pour un type de relation donné r, nous combinons les masses des relations de type r afin d'obtenir r-pré-résultat avec $\hat{m}_r$ défini comme suit : $\hat{m}_r = \bigotimes_{i \in I_r} m_{r,i}$
2. Après nous combinons tous les r-pré-résultats en utilisant : $\bigotimes_{r \in R} \hat{m}_r$

En fonction de l'opération ⓐ, une telle procédure peut finalement converger vers une certaine masse stationnaire.

Une fois que nous avons la masse de croyance globale sur un certain nœud, nous utilisons une version modifiée de la probabilité pignistique définie dans l'équation 3 afin de prendre la décision à propos du degré de l'influence du nœud. Dans notre cas les masses de croyance sont définies sur Λ et la probabilité pignistique est calculée en répartissant uniformément la masse de Ω sur tous les autres éléments de Λ :

$$\text{bet}(x) = m(x) + \frac{m(\Omega)}{|\Omega|}, \quad x \in \Omega \tag{7}$$

Le code source est disponible sur github : https://github.com/kerzol/Influence-assessment-in-twitter.

3.3. Illustrations

Afin d'illustrer notre méthode, nous considérons les fonctions de masse suivantes associées à la relation *retweet* et au pattern de diffusion *retweet* d'une *mention* :

$$\text{Retweet} \mapsto \begin{cases} m_{\text{Retweet}}(\text{Faible}) = 0.4 \\ m_{\text{Retweet}}(\Omega) = 0.6 \end{cases} \qquad \text{RTmention} \mapsto \begin{cases} m_{\text{RTmention}}(\text{Moyenne}) = 0.7 \\ m_{\text{RTmention}}(\Omega) = 0.3 \end{cases}$$

Les masses de croyance $m_{\text{Retweet}}(\Omega)$ et $m_{\text{RTmention}}(\Omega)$ représentent l'ignorance partielle. Nous avons affecté une masse de croyance plus importante au pattern d'interaction *Retweet* d'une *Mention* car nous considérons que l'existance de ce pattern est très significative en terme d'influence.

Cas : Retweet + Retweet d'une Mention

Après initialisation des masses de croyance pour les relations, nous suivons le processus de l'approche proposée pour mesurer l'influence obtenue suite à la combinaison d'un *Retweet* avec le pattern *retweet* d'une *Mention*. D'abord nous utilisons l'opération $\textcircled{@}$ donnant les correspondances entre les degrés d'influence (tableau 1), après nous calculons la combinaison conjonctive. La fonction de masse combinée des deux relations est donnée dans le tableau 2:

Tableau 2. Combinaison d'un Retweet avec un Retweet d'une Mention

$\otimes$	$Faible$	Ω
	0.4	0.6
$Moyenne$	$Assez\ Forte$	$Moyenne$
0.7	0.28	0.42
Ω	$Faible$	Ω
0.3	0.12	0.18

Nous obtenons : $m(\text{Faible}) = 0.12 \quad m(\text{Moyenne}) = 0.42$

$m(\text{Assez Forte}) = 0.28 \quad m(\Omega) = 0.18$

Finalement, pour prendre la décision sur le degré d'influence, nous calculons la probabilité pignistique en utilisant l'équation 7 (Tableau 3). Par exemple, pour le degré Faible, nous procédons comme suit pour obtenir la probabilité pignistique :

$$\text{bet}(\text{Faible}) = m(\text{Faible}) + \frac{m(\Omega)}{|\Omega|} = 0.12 + \frac{0.18}{8} = 0.1425$$

Tableau 3. Probabilité pignistique dans le cas d'un Retweet suivi d'un Retweet d'une Mention

Très Faible	0.0225
Faible	0.1425
Assez Moyenne	0.0225
Moyenne	0.4425
Assez Forte	0.3025
Forte	0.0225
Très Forte	0.0225
Extrêmement Forte	0.0225

On peut conclure que le degré d'influence d'un *Retweet* suivi d'un *Retweet* d'une *Mention* est Moyenne puisqu'il a la plus grande probabilité pignistique soit 0.4425.

4. Expérimentations et résultats

Les travaux de recherche menés se déroulent dans le cadre du projet TEE 2014 dont l'intitulé exact est "*Twitter* aux élections européennes : Une étude contrastive internationale des utilisations de *Twitter* par les candidats aux élections au Parlement Européen en mai 2014". Ce projet international, mené par la Maison des Sciences de l'Homme (MSH) de Dijon, réunit près de 45 chercheurs (majoritairement des politologues, sociologues, chercheurs en communication) de 10 laboratoires de recherche répartis dans 5 pays européens (France, Allemagne, Belgique, Italie et Espagne). L'objectif global de ce projet est d'observer et d'analyser la communication des politiques sur Twitter durant les élections européennes de mai 2014 dans les 5 pays d'étude.

4.1. Description des données

Pour collecter les informations de *Twitter*, nous avons utilisé notre outil *SNFreezer*[2] (Leclercq *et al.*, 2015). Trois types d'informations (généralisées sous le terme "source") peuvent être pris en paramètre dans cette collecte : des comptes utilisateurs, des *hashtags* et des mots ou phrases. Ces différentes sources ont été choisies par les politologues, et nous retrouvons parmi elles les noms des principaux candidats, leurs

2. https://github.com/SNFreezer

comptes *Twitter*, et les *hashtags* relatifs à ces candidats, leurs partis, ou plus généralement à l'élection étudiée. L'objectif de la collecte est de capter les *tweets* mentionnant les utilisateurs désignés, ceux contenant un certain *hashtag*, mot ou phrase, ou encore les *tweets* envoyés par les utilisateurs spécifiés. En plus, nous collectons les informations sur ces *tweets* tels que les *tweets retweetés*, les utilisateurs *mentionnés* dans les *tweets* et les *réponses* aux *tweets*. La collecte sur 50 jours consécutifs nous a permis de disposer de 37 millions de *tweets*.

4.2. Expérimentations et résultats

L'objectif de nos expérimentations est de mesurer l'influence des candidats sur *Twitter*. Afin d'étudier leur influence, nous affectons des masses aux relations considérés, puis pour chaque candidat, nous combinons les masses de croyances en itérant sur le nombre de *retweets*, *mentions* et *réponses*.

Le choix et l'affectation des masses dans l'étape de l'initialisation sont une question importante lorsque nous traitons de données réelles. Dans certains domaines tels que la politique, les utilisateurs ont un très grand nombre de relations. Avec une initialisation avec les mêmes valeurs que celles utilisées dans la section illustration, l'influence converge pour tous les candidats vers le plus haut degré d'influence Extrêmement Forte après un nombre d'itérations restreint ($\simeq$ 45 itérations), ce qui ne nous a pas permis de pouvoir comparer l'influence des candidats. Pour règler cette question, nous effectuons une mise à l'échelle et nous utilisons les affectations de masses suivantes :

$$\textit{Retweet} \mapsto \begin{cases} m_{\text{Retweet}}(\text{Très Faible}) = 0.55 \cdot 10^{-3} \\ m_{\text{Retweet}}(\Omega) = 1 - 0.55 \cdot 10^{-3} \end{cases}$$

$$\textit{Mention} \mapsto \begin{cases} m_{\text{Mention}}(\text{Très Faible}) = 0.45 \cdot 10^{-3} \\ m_{\text{Mention}}(\Omega) = 1 - 0.45 \cdot 10^{-3} \end{cases}$$

$$\textit{Réponse} \mapsto \begin{cases} m_{\text{Réponse}}(\text{Très Faible}) = 0.45 \cdot 10^{-3} \\ m_{\text{Réponse}}(\Omega) = 1 - 0.45 \cdot 10^{-3} \end{cases}$$

Nous avons appliqué notre approche sur le corpus français comprenant 616 candidats et 4 millions de *tweets*. Le tableau 4 montre les résultats obtenus avec notre approche pour trois candidats français Marine Le Pen, Florian Philippot et Jean-Luc Mélenchon. Nous pouvons conclure que le degré d'influence dans *Twitter* pour le candidat Marine Le Pen est Extrêmement Forte avec la masse de croyance de 0.8173448. Les résultats fournissent non seulement le degré d'influence d'un candidat, mais aussi donnent par les masses sur les différents degrés d'influence une indication de la croyance que nous avons dans les résultats donnés.

Tableau 4. Résultats pour 3 candidats français influents

	M. Le Pen	**F. Philippot**	**J.L. Mélenchon**
Ω	0	0	0
Très Faible	0	0.000011065	0.000030278
Faible	0	0.00007295998	0.0001832843
Assez Moyenne	0	0.0007035528	0.001403947
Moyenne	0	0.003033557	0.004954501
Assez Forte	0	0.008340205	0.01247841
Forte	0	0.02191526	0.02977818
Très Forte	0.1826552	0.5830090	0.7960571
Extrêmement Forte	0.8173448	0.3829144	0.1551143

4.3. Vers le classement de l'influence des utilisateurs

L'approche proposée peut être aussi exploitée pour classer les utilisateurs selon leur influence. Afin de classer les candidats selon leur influence et en partant de nos résultats, nous procédons comme suit :

1. Pour chaque candidat nous prenons le degré d'influence ayant la masse de croyance maximale (par exemple, pour Marine Le Pen nous choisissons Extrêmement Forte)

2. Nous classons les candidats selon leur "degré d'influence maximal"

3. Si deux candidats ont le même "degré d'influence maximal", nous comparons les masses de croyance du plus haut degré d'influence en utilisant l'ordre suivant pour les degrés d'influence : $\Omega < Très\ Faible < Faible < Assez\ Moyenne < Moyenne < Assez\ Forte < Forte < Très\ Forte < Extrêmement\ Forte$

Nous procédons ainsi puisqu'il est injuste de classer les candidats selon la masse de croyance maximale qu'ils ont dans les différents degrés. Nous pouvons avoir un utilisateur plus influent qu'un autre même s'il a une masse de croyance plus faible que lui sur le même degré. Ceci est dû au fait que, les masses de croyance du plus haut degré d'influence suivant ont augmenté et sont devenues assez importantes. Par exemple, Influence(Florian Philippot) = Influence(Jean-Luc Mélenchon) = Très Forte, et $m_{\text{Philippot}}(\text{Extrêmement Forte}) > m_{\text{Mélenchon}}(\text{Extrêmement Forte})$ bien que le candidat Florian Philippot a une masse de croyance sur le degré Très Forte plus faible que la masse de croyance du candidat Jean-Luc Mélenchon sur le même degré (voir tableau 4), il est classé avant Jean-Luc Mélenchon (tableau 5) puisqu'il a une masse de croyance plus élevée sur le degré Extrêmement Forte. Nous effectuons alors la procédure de la combinaison pour tous les candidats et déduisons leur classement selon leur degré d'influence. Les résultats sont présentés dans le tableau 5.

Le tableau 6 présente le classement obtenu en utilisant les critères utilisés par (Cha *et al.*, 2009). Ces critères sont le nombre de Retweets, Mentions et Réponses. Les résultats présentés ne montrent pas l'influence globale dans le réseau puisque nous trouvons différents classements pour chaque type de relation. Alors que notre méthode (Tableau 5) nous permet d'avoir un classement unique qui tient compte de

Tableau 5. Candidats français les plus influents selon notre approche

Classement	Candidats	Degré d'influence	Masse de croyance
1	Marine Le Pen	Extrêment Forte	0.8173448
2	Florian Philippot	Très Forte	0.5830090
3	Jean-Luc Mélenchon	Très Forte	0.7960571
4	Christine Boutin	Très Forte	0.9796956
5	Aymeric Chauprade	Très Forte	0.4171324655
6	Nicolas Dupont-Aignan	Très Forte	0.5293170700
7	José Bové	Très Forte	0.2925722297
8	Geoffroy Didier	Moyenne	0.2092645352
9	Raquel Garrido	Moyenne	0.2048485
10	Marielle De Sarnez	Assez Moyenne	0.2074260

Table 6. Candidats français les plus influents selon les différentes relations et le degré de centralité

Classement	*Retweet*	*Mention*	*Réponse*	*Degré de centralité*
1	Marine Le Pen	Marine Le Pen	Christine Boutin	Marine Le Pen
2	Florian Philippot	Christine Boutin	Marine Le Pen	Christine Boutin
3	Jean-Luc Mélenchon	Jean-Luc Mélenchon	Florian Philippot	Florian Filippot
4	Aymeric Chauparde	Florian Philippot	Jean-Luc Mélenchon	Jean-Luc Mélenchon
5	François Asselineau	Nicolas Dupont-Aignan	Louis de Gouyon Matigon	Nicolas Dupont-Aignan
6	Corinne Morel-Darleux	José Bové	Nicolas Dupont-Aignan	Aymeric Chauparde
7	Nicolas Dupont-Aignan	Aymeric Chauparde	Jean-Sébastien Herpin	José Bové
8	Louis Aliot	Raquel Garrido	Julien Rochedy	Geoffroy Didier
9	Denis Payre	Jérome Lavrilleux	Geoffroy Didier	Raquel Garrido
10	Yannick Jadot	Marielle de Sarnez	Louis Aliot	Yannick Jadot

tous les critères considérés. La dernière colonne du tableau 6 montre le classement des candidats selon leur degré de centralité calculé en utilisant le nombre de voisins de chaque candidat dans le réseau. Le degré de centralité permet, pour chaque candidat, d'avoir un classement global sans indication sur leur degré d'influence contrairement à nos résultats présentés dans le tableau 5, l'influence mesurée est globale en prenant en compte les relations possibles dans la même mesure.

5. Conclusion

Dans cet article, nous avons proposé une approche pour l'évaluation de l'influence sur le réseau social *Twitter*. Cette approche répond à des limites des systèmes existants tels que la prise en compte de la combinaison des relations et l'incertitude engendrée par la fusion d'informations. Dans notre approche, nous avons proposé un graphe de l'influence nous permettant de considérer les différentes relations dans le réseau (*Retweet*, *Mention* et *Réponse*) ainsi que les séquences possibles de relations. En se basant sur la théorie des fonctions de croyance, nous avons établi une mesure d'influence globale pour les utilisateurs par combinaison des différentes relations. Nous avons expérimenté l'approche sur des données *Twitter* collectées dans le cadre du projet de recherche TEE 2014. Pour renforcer l'approche proposée nous souhaitons enrichir le modèle et développer d'autres patterns d'interaction en col-

laboration avec les politologues du projet TEE 2014. Nous souhaitons intégrer dans notre approche les *hashtags* et les *favoris* et ainsi pouvoir traiter des patterns plus complexes. Cependant, l'extraction des patterns du jeu de données nécessite des structures de données adaptées pour concevoir une application utilisable par les politologues et permettant d'évaluer et d'affiner leur modèle de l'influence. Par ailleurs, la méthode de classement des utilisateurs sera améliorée, par exemple avec des intervalles de confiance et nous allons comparer les résultats obtenues avec les résultats obtenus à partir des algorithmes connus dans la littérature tels que TwitterRank et HITS.

Bibliographie

Ashwini S. S., M.R. S. (2015). Profile ranking using user influence and content relevance with classification using sentiment analysis. *International Journal of Computer Science and Mobile Computing*, vol. 4, p. 1075–1080.

Bakshy E., Hofman J. M., Mason W. A., Watts D. J. (2011). Everyone's an Influencer: Quantifying Influence on Twitter. In *Proceedings of the fourth acm international conference on web search and data mining*, p. 65–74. New York, NY, USA, ACM. Retrieved from http://doi.acm.org/10.1145/1935826.1935845

Barnes J. A. (1969). Graph Theory and Social Networks: A Technical Comment on Connectedness and Connectivity. *Sociology*, p. 215-232.

Brown P. E., Feng J. (2011). Measuring user influence on twitter using modified k-shell decomposition. In *Fifth international aaai conference on weblogs and social media*, p. 18-23.

Cai G., Daijun W., Yong H., Sankaran M., Yong D. (2013). A modified evidential methodology of identifying influential nodes in weighted networks. *Physica A: Statistical Mechanics and its Applications*, vol. 392, n° 21, p. 5490 - 5500. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378437113005773

Cha M., Haddadi H., Benevenuto F., Gummadi K. (2010). Measuring user influence in twitter: The million follower fallacy. In *4th international aaai conference on weblogs and social media (icwsm).*

Cha M., Mislove A., Gummadi K. P. (2009). A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proceedings of the 18th international conference on world wide web*, p. 721–730. New York, NY, USA, ACM.

Chen D.-B., Gao H., Lü L., Zhou T. (2013). Identifying Influential Nodes in Large-Scale Directed Networks: The Role of Clustering. *PLoS ONE*, vol. 8, n° 10.

Denoeux T., Masson M.-H. E. (2012). Belief Functions: Theory and Applications. In *Proceedings of the 2nd international conference on belief functions, 9-11 may 2012*, p. 444.

Kanawati R. (2015). Multiplex network mining: a brief survey. *IEEE Intelligent Informatics Bulletin*.

Kleinberg J. M. (1999, September). Authoritative sources in a hyperlinked environment. *J. ACM*, p. 604–632.

Kotz S., N. L. Johnson eds. W. (1982). Belief functions. *Encyclopedia of Statistical Sciences 1 209*.

Leavitt A., Burchard E., Fisher D., Gilbert S. (2009, September). The Influentials: New Approaches for Analyzing Influence on Twitter. *Webecology Project*.

Leclercq E., Savonnet M., Grison T., Kirgizov S., Basaille I. (2015). SNFreezer: a Platform for Harvesting and Storing Tweets in a Big Data Context. In A. Frame, A. Mercier, G. Brachotte, C. Thimm (Eds.), *Twitter and the european parliamentary elections: researching political uses of microblogging*, p. 1–16. DE, Peter Lang.

Mo H., Gao C., Deng Y. (2015, April). Evidential method to identify influential nodes in complex networks. *Systems Engineering and Electronics, Journal of*, vol. 26, n° 2, p. 381-387.

Nimier V., Appriou A. (1995). Utilisation de la théorie de dempster-shafer pour la fusion d'informations. *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*, p. 137-140.

Page L., Brin S., Motwani R., Winograd T. (1999). The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th international world wide web conference*, p. 161–172.

Qasem Z., Jansen M., Hecking T., Hoppe H. (2015). On the detection of influential actors in social media. In *Signal-image technology and internet-based systems (sitis), 2015 11th international conference on*, p. 421–427.

Rodriguez M. A., Shinavier J. (2010). Exposing multi-relational networks to single-relational network analysis algorithms. *Journal of Informetrics*, vol. 4, n° 1, p. 29–41.

Romero D. M., Galuba W., Asur S., Huberman B. A. (2011). Influence and passivity in social media. In *Proceedings of the 20th international conference companion on world wide web*, p. 113–114.

Seidman S. B. (1983). Network structure and minimum degree. *Social Networks*, vol. 5, n° 3, p. 269 - 287.

Simmie D., Vigliotti M., Hankin C. (2013). Ranking twitter influence by combining network centrality and influence observables in an evolutionary model. In *Signal-image technology internet-based systems (sitis), 2013 international conference on*, p. 491-498.

Smets P. (1989). Constructing the pignistic probability function in a context of uncertainty. In *Uai*, vol. 89, p. 29–40.

Smets P. (1997). Imperfect Information: Imprecision and Uncertainty. In A. Motro, P. Smets (Eds.), *Uncertainty management in information systems*, p. 225-254.

Smets P., Kennes R. (2008). The transferable belief model. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, p. 693-736.

Wei D., Deng X., Zhang X., Deng Y., Mahadevan S. (2013). Identifying influential nodes in weighted networks based on evidence theory. *Physica A: Statistical Mechanics and its Applications*, vol. 392, n° 10, p. 2564 - 2575. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378437113001076

Weng J., Lim E.-P., Jiang J., He Q. (2010). TwitterRank: Finding Topic-sensitive Influential Twitterers. In *Proceedings of the third acm international conference on web search and data mining*, p. 261–270. New York, NY, USA, ACM.

Wu Z., Yin W., Cao J., Xu G., Cuzzocrea A. (2013). Community detection in multi-relational social networks. In *Web Information Systems Engineering – WISE 2013*, p. 43-56.

Session SI Dédiés

Cadre d'évaluation de systèmes de reconnaissance d'entités nommées spatiales

Damien Palacio [1], **Christian Sallaberry** [2], **Guillaume Cabanac** [3], **Gilles Hubert** [3]

1. *Department of Geography, University of Zurich, Zurich, Suisse*
   damien.palacio@geo.uzh.ch
2. *LIUPPA ÉA 3000, Université de Pau et des Pays de l'Adour, Pau, France*
   christian.sallaberry@univ-pau.fr
3. *IRIT UMR 5505 CNRS, Université Paul Sabatier, Toulouse, France*
   {guillaume.cabanac,gilles.hubert}@univ-tlse3.fr

RÉSUMÉ. *La reconnaissance d'entités nommées est une tâche de l'activité d'extraction d'information dans des corpus textuels. Des systèmes de reconnaissance d'entités nommées spatiales sont très largement utilisés, mais souvent sans en connaître les forces et faiblesses. C'est pourquoi nous proposons le cadre d'évaluation SNERBM (Spatial Name Entity Recognition BenchMark) comme référentiel commun et nous l'expérimentons sur six systèmes existants de reconnaissance d'entités nommées spatiales. Ce cadre a pour objectif l'évaluation et la comparaison des performances de tels systèmes. Il permet également d'envisager le choix d'un système, ou encore la combinaison de différents systèmes, particulièrement adaptés aux catégories d'entité nommées spatiales (ville, barrage, montagne, par exemple) majoritairement présentes dans un corpus donné.*

ABSTRACT. *Named entity recognition is a task of information extraction from textual corpora. Spatial named entity recognition systems are widely used in this respect, but no one actually knows about their pros and cons. This is why we propose the SNERBM evaluation framework as a benchmark (Spatial Name Entity Recognition BenchMark), which we experimented on six existing systems dedicated to spatial named entity recognition. This benchmark enables the evaluation and the comparison of performances of such systems. In addition, it informs the selection of a system, or a combination of systems, best appropriate to operate on a given textual corpus featuring specific categories of spatial named entities (e.g., cities, mountains).*

MOTS-CLÉS : *reconnaissance d'entité nommée, entité nommée spatiale, cadre d'évaluation de système*

KEYWORDS: *named entity recognition, spatial named entity, system evaluation benchmark*

1. Introduction

La reconnaissance d'entités nommées (REN) dans des textes (Chinchor, 1998), consiste à identifier des syntagmes appelés entités nommées (c'est-à-dire des noms propres, des expressions de temps et des expressions numériques) catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, dates, quantités, distances, valeurs, acronymes, abréviations, etc. Parmi les entités nommées, les noms de lieux, que nous appellerons entités nommées spatiales (ENS), désignent des objets géographiques tels que des entités administratives (commune, par exemple), des éléments du relief, des éléments hydrographiques, etc.

Un système d'analyse d'ENS est généralement composé d'un module de reconnaissance (marquage d'ENS, tel que « Yosemite »), d'un module de classification (typage d'ENS, tel que « Yosemite Park »), d'un module de désambiguïsation (ENS vs. non ENS, tel que « Canari » vs. « canari » ou ENS vs. ENS différente, tel que « Paris » aux U.S. vs. « Paris » en France) et d'un module de géolocalisation (géocodage d'ENS, tel qu'avec les coordonnées du centre de « Yosemite Park » – latitude : 37,75, longitude : $-119{,}50$ ou de la géométrie correspondante). Les premiers modules contribuent à une construction par étape d'une représentation symbolique d'une ENS et le dernier détermine une représentation numérique d'une ENS (coordonnées géolocalisées).

Les systèmes de traitement automatique de la langue (TAL) supportent la reconnaissance et la classification d'entités nommées. Les systèmes d'analyse d'ENS supportent eux la reconnaissance, la classification, la désambiguïsation et la géolocalisation d'ENS (Marrero *et al*., 2009 ; 2013). Nous pouvons citer Clavin [1] (D'Ignazio, 2013), GeoDict [2], Geolocator [3] (Gelernter, Zhang, 2013), OpenCalais [4] (D'Ignazio, 2013), Unlock [5] (Grover *et al*., 2010) et Yahoo!PlaceSpotter [6] (Tobin *et al*., 2010 ; Anastácio *et al*., 2010 ; D'Ignazio, 2013) comme autant d'exemples de systèmes d'analyse allant de la reconnaissance jusqu'à la géolocalisation d'ENS. Par abus de langage, on parle souvent de systèmes de reconnaissance d'ENS (RENS). L'efficacité de tels systèmes peut être évaluée au travers de campagnes d'évaluation telles que MUC, CoNLL ou ACE (Marrero *et al*., 2009). Ces campagnes mesurent l'efficacité des systèmes pour la reconnaissance d'entités nommées de type lieu (ENS), nom de personne, nom d'organisation… dans des corpus de documents associés. Peu de systèmes participant à ces compétitions, l'évaluation de systèmes d'analyse d'ENS commerciaux ou libres a été peu étudiée (Marrero *et al*., 2009).

Les systèmes de recherche d'information géographique (RIG) intègrent des systèmes d'analyse d'entités nommées spatiales (ENS) (Leidner, 2007 ; Andogah, 2010 ;

1. http://clavin.bericotechnologies.com
2. http://www.datasciencetoolkit.org/developerdocs#commandline
3. https://github.com/geoparser/geolocator
4. http://www.opencalais.com
5. http://edina.ac.uk/unlock
6. http://developer.yahoo.com/boss/geo/docs

Sallaberry, 2013). Au delà de la seule RENS, les RIG indexent les ENS et proposent des langages d'interrogation supportant l'appariement des critères spatiaux exprimés dans un besoin d'information avec les données spatiales indexées. Ainsi, le système de RIG SPIRIT (Vaid *et al*., 2005), par exemple, comprend des processus dédiés à la reconnaissance, à la désambiguïsation et à la géolocalisation d'ENS.

Nous proposons le cadre d'évaluation de systèmes de reconnaissance d'entités nommées spatiales, dénommé SNERBM (*Spatial Name Entity Recognition BenchMark*). Ce cadre est ouvert à l'évaluation de tout nouveau système et son corpus de test peut également être étendu par de nouvelles données. La proposition contribue aux travaux de la communauté à plusieurs titres : (*i*) ce *benchmark* est un outil qui permet d'évaluer et d'améliorer un système de RENS ; (*ii*) il permet également à un utilisateur non spécialiste de comparer des systèmes de RENS et de choisir le plus performant pour une catégorie d'entité ; (*iii*) de plus, il propose une évaluation par catégorie d'ENS qui permet d'envisager des combinaisons avantageuses de différents systèmes ; (*iv*) enfin, il est ouvert et, par conséquent, invite la communauté à se l'approprier et à l'enrichir.

Cet article est organisé comme suit. Dans la section 2, nous décrivons la problématique relevant des campagnes d'évaluation de systèmes de recherche d'information et de systèmes de REN. Nous présentons notre cadre d'évaluation nommé « benchmark SNERBM » en section 3. Ensuite, en section 4, nous expérimentons le *benchmark* en comparant six systèmes de RENS en termes de qualité de réponse et de rapidité. Enfin, nous concluons par une discussion et des perspectives.

2. L'évaluation de systèmes de reconnaissance d'entités nommées (REN)

De nombreuses campagnes d'évaluation proposent des plateformes pour évaluer des systèmes de REN ou de RIG (Leidner, 2007 ; Andogah, 2010 ; Nouvel, 2012), comme indiqué dans le tableau 1 :

– MUC : Message Understanding Conference (Chinchor, 1998),

– MET : Multilingual Extraction Task (Chinchor, 1998),

– IREX : Information Retrieval and Extraction eXercise (Sekine, Eriguchi, 2000),

– CoNLL : Computational Natural Language Learning conference (Tjong Kim Sang, De Meulder, 2003),

– HAREM : Avaliaçao de sistemas de Reconhecimento de Entidades Mencionadas (Santos *et al*., 2006),

– GeoClef : Geographic Cross Language Evaluation Forum (Mandl *et al*., 2009),

– ACE : Automatic Content Extraction program (Strassel *et al*., 2008),

– EVALITA : Evaluation of NLP and Speech Tools for Italian (Lenzi *et al*., 2013),

– TREC-CS : Text Retrieval Conference – Contextual Suggestion (Voorhees, 2001 ; Dean-Hall *et al*., 2013).

Ainsi, différents types de *benchmarks* (référentiels) ont été expérimentés (tableau 2). Les ressources disponibles comprenant des ENS pré-marquées sont composées de

Tableau 1. Campagnes d'évaluation de systèmes de REN

Campagne	Année	Corpus	Système	Cible	Représentation	
					symbolique	numérique
MUC-7	1997	Reportages	NER	Personne, lieux, organisation, temps, mesure	X	
MUC-7/MET-2	1998	Journaux	NER	Personne, lieux, organisation, temps	X	
IREX	1999	Journaux	NER	Personne, lieux, organisation, temps	X	
CoNLL	2003	Journaux	NER	Personne, lieux, organisation, divers	X	
HAREM	2006	Journaux	NER	Personne, lieux, organisation, temps, mesure	X	
GeoCLEF	2008	Journaux	GIR	Lieux, thème	X	X
ACE	2008	Journaux	NER	Personne, lieux, organisation, géopolitique	X	
EVALITA	2011	Journaux	NER	Personne, lieux, organisation, géopolitique	X	
TREC-CS	2013	Open web, ClueWeb12	GIR	Lieux, thème	X	X

petits échantillons de documents (par ex., la campagne CoNLL propose un jeu de 231 documents en anglais avec 1 668 ENS annotées manuellement et la campagne ACE propose 400 documents en anglais avec aucune ENS annotée). Par ailleurs, les ressources dédiées à l'évaluation de systèmes de RIG proposent des jeux de données regroupant sans distinction les documents à la fois thématiquement et spatialement pertinents (par ex., la campagne GeoCLEF propose un jeu de documents associé à 100 *topics* géographiques (besoins d'information comportant des critères spatiaux et thématiques) et aux jugements de pertinence correspondants (*Qrels*), mais aucune ENS pré-annotée n'est proposée).

D'autres approches, généralement dédiées à l'évaluation d'un système particulier, utilisent des corpus plus petits dont les ressources annotées ne sont que rarement mises à disposition (tableau 3). Bucher *et al*. (2005) ont ainsi proposé d'adapter des techniques d'évaluation existantes pour évaluer le système de RIG SPIRIT. Marrero *et al*. (2009) présentent une plateforme qui a permis l'évaluation des systèmes de REN Supersense, Afner, Annie, Freeling, TextPro, YooName, ClearForest et Lingpipe. Tobin *et al*. (2010) décrivent une approche dédiée à l'évaluation de système de RENS et plus particulièrement des modules de désambiguïsation et de géolocalisation des systèmes Unlock et Yahoo!PlaceMaker. Anastácio *et al*. (2010) ciblent également l'évaluation de systèmes de RENS. Les auteurs comparent les méthodes de calcul de portée spatiale supportées respectivement par les systèmes Yahoo!PlaceMaker, GIPSY, Web-a-Where et GraphRank. D'Ignazio (2013) compare les services d'analyse spatiale des systèmes de RENS OpenCalais, Clavin et Yahoo!PlaceSpotter. Enfin, Gelernter et Zhang (2013) évaluent le système de RENS Geolocator. Les auteurs mesurent la qualité du module de reconnaissance de toponymes de Geolocator ainsi que de son module d'analyse et de géolocalisation.

Notre objectif est d'évaluer différents systèmes de RENS suivant un même référentiel. Les *benchmarks* cités précédemment sont rarement mis à disposition. Or, nous avons pu nous procurer et travailler sur les *benchmarks* TREC-CS (Dean-Hall *et al*.,

Tableau 2. Compléments relatifs aux campagnes d'évaluation de systèmes de REN

Campagne	Année	Documents	Langues	Entités nommées	Bibliographie	Ressources annotées
MUC-7	1997	158 000	anglais	-	[Chinchor'98]	jeu de test
MUC-7/MET-2	1998	500	chinois, japonnais	-	[Chinchor'98b]	jeu de test
IREX	1999	1 371	japonnais	-	[Sekine'00]	jeu de test
CoNLL	2003	1 499	anglais, allemand	11 503	[Tjong'03]	jeu de test
HAREM	2006	1 202	portugais	5 132	[Santos'06]	jeu de test
GeoCLEF	2008	200 000	portugais, allemand, anglais	-	[Mandl'08]	jeu : thématique, géographique
ACE	2008	10 000	anglais, arabe	-	[Strassel'08]	jeu de test
EVALITA	2011	42 595	italien	1 924	[Bartalesietal'11]	jeu de test
TREC-CS	2013	30 144	anglais	-	[Dean-Hall'13]	jeu : géographique

Tableau 3. Campagnes d'évaluation ad hoc dédiées aux systèmes de REN spécifiques

Bibliographie	Documents	Langues	Entités nommées	Systèmes évalués	Ressources annotées
[Bucher'05]	21 094	anglais	-	RIG : SPIRIT	-
[Marrero'09]	1	anglais	100 EN	REN : Supersense, Afner, Annie, Freeling, TextPro, YooName, ClearForest, Lingpipe	-
[Tobin'10]	1 032	anglais	13 077 ENS	RENS : Unlock, Yahoo PlaceMaker	-
[Anastacio'10]	6 000	anglais	1 100 ENS	RENS : Yahoo PlaceMaker, GIPSY, Web-a-Where, GraphRank	-
[Ignazio'13]	75	-	-	RENS : OpenCalais, Clavin, Yahoo PlaceSpotter	-
[Gelernter'13]	1 306	espagnol, anglais	799 ENS	RENS : Geolocator	-

2013) et GeoparsingQT (Gelernter, Zhang, 2013). Le premier ne propose que des ENS de type nom de grandes villes d'Amérique du nord. Le second est plus intéressant car il traite de quinze catégories différentes d'ENS.

Aussi, nous proposons de construire et d'expérimenter un *benchmark* ouvert, basé sur GeoparsingQT. En effet, GeoparsingQT comporte un jeu d'ENS constitué par des géographes et utilisé par l'équipe de développeurs du système Geolocator (Gelernter, Zhang, 2013) pour mesurer les éventuels effets de bord engendrés par chaque passage à une version supérieure du système de RENS. À l'image de *PABench*, pour *Points Of Interest (POI) Alignment Benchmark* (Morana *et al*., 2014 ; Berjawi *et al*., 2015), dédié à l'évaluation de systèmes d'appariement de POI issus de différents services de description et de géolocalisation (Geonames, Google Maps, Bing Maps, par exemple), nous proposons un référentiel ouvert et évolutif, validé par des utilisateurs.

Nous prenons comme cas d'application un ensemble de systèmes de RENS existants : Clavin, Geodict, Geolocator, OpenCalais, Unlock et Yahoo!PlaceSpotter. À ce jour, ces systèmes n'ont pas été évalués ni confrontés au sein d'un même *benchmark*.

3. Le cadre d'évaluation SNERBM

Plus d'une dizaine de systèmes de reconnaissance d'ENS ont été proposés dans la littérature lors des vingt dernières années (Lieberman *et al*., 2010 ; Lingad *et al*., 2013). Certains ont été évalués sur des critères de qualité de réponse, de temps de réponse, ou

les deux (tableau 4). Cependant, les cadres d'évaluation mis en œuvre reposent sur des corpus et des métriques différents. Comme illustré au tableau 3, peu d'études ont visé la comparaison de systèmes et aucune n'a mis des ressources annotées à disposition. Par conséquent, il est impossible à ce jour de connaître les performances relatives des systèmes suivant un même référentiel.

Tableau 4. Compléments relatifs aux campagnes d'évaluation ad hoc dédiées aux systèmes de REN

		Evaluation	
Système	**Cible**	**Effectiveness (qualité)**	**Efficiency (temps)**
Afner	REN	[Marrero'09]	
Annie	REN	[Marrero'09]	
Clavin	RENS	[Ignazio'13]	
ClearForest	REN	[Marrero'09]	
Freeling	REN	[Marrero'09]	
Geolocator	RENS	[Gelernter'13]	
GIPSY	RENS	[Anastacio'10]	
GraphRank	RENS	[Anastacio'10]	
LingPipe	REN	[Marrero'09]	
OpenCalais	RENS	[Ignazio'13]	
SPIRIT	RIG	[Bucher'05], [Vaid'05]	[Vaid'05]
Supersense	REN	[Marrero'09]	
TextPro	REN	[Marrero'09]	
Unlock	RENS	[Tobin'10]	
Web-a-Where	RENS	[Anastacio'10]	
Yahoo!PlaceMaker	RENS	[Tobin'10], [Anastacio'10]	
Yahoo!PlaceSpotter	RENS	[Ignazio'13]	
YooName	REN	[Marrero'09]	

Ce type de problème a trouvé des réponses en recherche d'information (RI). La RI s'appuie sur une longue tradition d'évaluation, notamment, via des campagnes d'évaluation de systèmes de recherche d'information (SRI). Ces campagnes définissent et implémentent des *benchmarks* pour évaluer et confronter les performances de SRI (Voorhees, 2002 ; 2007). Par exemple, la campagne TREC implémente le *benchmark* TREC-CS visant à évaluer la qualité (*effectiveness*) des recommandations de lieux, sans toutefois évaluer le temps de réponse (*efficiency*) des systèmes.

En ce qui concerne les systèmes de RENS, aucun *benchmark* existant ne s'est imposé comme référentiel commun. Le principal frein à l'adoption d'un *benchmark* est certainement le verrouillage (la non diffusion) des corpus et du code du logiciel d'évaluation. Dans cet article, nous proposons un *benchmark* d'évaluation des systèmes de RENS, basé sur un corpus ouvert. Nous avons appelé ce cadre d'évaluation SNERBM. Les principaux apports de cette proposition sont :

– *la couverture.* La qualité des systèmes est doublement évaluée : l'*effectiveness* mesure la qualité de la RENS tandis que l'*efficiency* mesure la réactivité des systèmes. Ces deux indications permettent d'identifier le système offrant le meilleur rapport *effectiveness–efficiency*.

– *la neutralité.* Le *benchmark* proposé est ouvert et peut être étendu par des jeux de tests complémentaires afin de prendre en compte des cas de figure non étudiés jusqu'alors. Il s'agit de traiter le maximum de cas possibles et de ne favoriser aucun système au regard du corpus employé ou des métriques d'évaluation.

– *la réutilisabilité.* Nous comparons dans cet article N systèmes. Le *benchmark* permet de reproduire nos résultats et de positionner une variante de système ou même un nouveau système par rapport à ces N performances de référence (*baselines*).

Le *benchmark* SNERBM s'appuie sur la collection de test GeoparsingQT (Gelernter, Zhang, 2013) et comprend :

– des catégories d'ENS : 15 catégories de type *Nom unique aux États-Unis*, *Nom et contexte associé*, *Abréviation administrative*, *Nom sans typologie associée*, *Nom et niveau administratif associé*, *Nom avec diacritique*, *Séquence de noms avec caractéristiques communes*, *Nom avec typologie associée*, *Nom officiel, court ou dérivé*, *Nom retors avec caractères spéciaux*, *Abréviation*, *Nom retors avec caractères numériques*, *Surnom*, *Nom historique*, *Autre type de référence* (tableau 5).

– 244 phrases : selon le modèle *sentence case*. Chaque phrase est associée à une seule catégorie (tableau 5).

– des jugements de pertinence (appelés *qrels*, pour *query relevance judgments* dans le vocabulaire TREC) : pour chaque phrase, les ENS sont géolocalisées par un ponctuel et/ou une géométrie déterminés par des experts (figures 1 et 2), respectivement à l'aide des ressources Geonames [7] et Google Map API V3 Tool [8].

Par exemple, pour la phrase « Rhodesia and Northern Rhodesia were renamed Zimbabwe and Zambia » de la catégorie « Historical Places / Nom historique », le fichier des Qrels contient 2 polygones (figure 1, éditée sous *kml.appspot* [9]). De même, pour la phrase « He traveled to a cape named Big Island in North Carolina » de la catégorie « Names Which Are The Same for Different Feature Types / Nom avec typologie associée », le fichier des Qrels contient 1 polygone (figure 2).

Ainsi, SNERBM permet de mesurer la performance d'un système de RENS suivant deux volets : la qualité des résultats (*effectiveness*) et la rapidité du système (*efficiency*).

3.1. Critères d'efficacité (effectiveness)

Pour mesurer la qualité des résultats fournis par un système de RENS, nous lui soumettons chaque phrase du jeu de test et évaluons les ENS détectées selon une démarche TREC classique. Nous capitalisons sur ce cadre de référence pour calculer des indicateurs de précision, de rappel et de F-mesure (Manning *et al.*, 2008, chap. 8). Ainsi, comme le montre la figure 3, une phrase est soumise au système qui restitue

7. http://www.geonames.org
8. http://www.birdtheme.org/useful/v3tool.html
9. http://display-kml.appspot.com/

Figure 1. Deux géométries sont associées à la phrase « Rhodesia and Northern Rhodesia were renamed Zimbabwe and Zambia » dans les Qrels

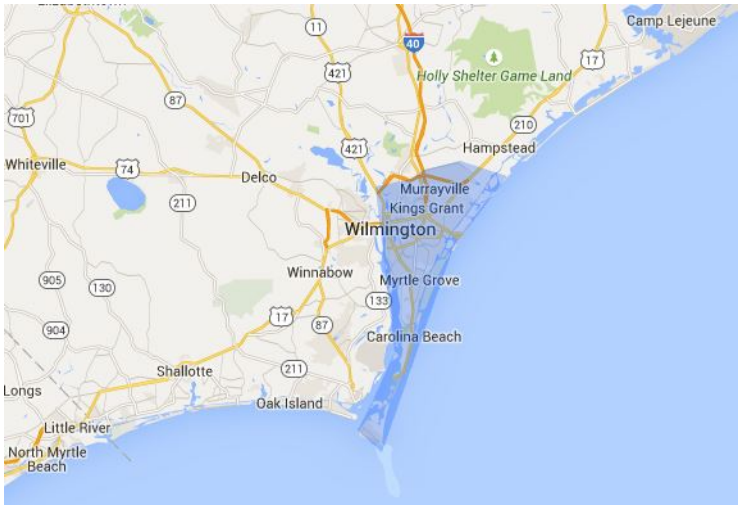


Figure 2. Une géométrie est associée à la phrase « He traveled to a cape named Big Island in North Carolina » dans les Qrels

alors aucune, une ou plusieurs ENS détectées et géolocalisées (points ou polygones correspondants). Une seconde phase de traitement, nommée « Intersections » sur la figure 3, compare ce résultat d'annotation avec les géométries définies dans les Qrels. Cette phase permet, pour une phrase donnée, de construire un résultat avec les caractéristiques suivantes :

– s'il existe une intersection entre une géométrie du résultat et une géométrie des Qrels alors : création d'un n-uplet (numéro phrase i, ..., numéro document d correspondant dans Qrels, ..., nom du système de RENS), par exemple, [phrase_31, ..., doc_34, système_Clavin] qui signifie que ce ponctuel (ou polygone) renvoyé par le système Clavin est en intersection avec un polygone associé dans les Qrels;

– s'il n'existe pas d'intersection entre une géométrie du résultat et une géométrie des Qrels alors : création d'un n-uplet (numéro phrase i, ..., numéro document d fictif (numéro phrase $i \times 1000$) inexistant dans les Qrels, ..., nom du système de RENS), par exemple, [phrase_31, ..., doc_31000, système_Clavin] qui signifie que ce ponctuel

Tableau 5. Exemples associés aux catégories du benchmark

Catégorie	Phrase	Commentaire
Abréviation	He climbed Mt. McKinley in Alaska.	Mount McKinley, Alaska
Abréviation administrative	He went to San Francisco, CA.	California
Autre type de référence	There was an accident at -120.9762, 41.25.	Longitude/Latitude Adin, CA
Divers	The French value their freedom.	People of France
Nom avec diacritique	Biên Hòa is a city in Vietnam.	City of Bien Hoa, Vietnam
Nom avec typologie associée	He went to the town of Big Island.	Town named Big Island
Nom et contexte associé	She was born in Paris in Idaho.	City of Paris, Idaho
Nom et niveau administratif associé	He went to Georgia, Kansas.	City of Georgia, KS
Nom et niveau administratif associé	He went to Tblisi, Georgia.	Country of Georgia
Nom historique	Ceylon became Sri Lanka.	Ceylon was renamed Sri Lanki
Nom officiel, court ou dérivé	The Republic of Korea is on a peninsula.	Country of South Korea
Nom retor avec caractères numériques	Green Creek Dam Nr. 5 was in need of repair.	Dam in Erath, Texas
Nom retor avec caractères spéciaux	They were vacationing in Trinidad and Tobago.	Country Trinidad and Tobago
Nom sans typologie associée	He climbed Rainier.	Mount Rainier
Nom unique aux États-Unis	She vacationed on the shores of Metacomet Lake.	Lake
Noms avec caractéristiques communes	The bicylce race went through Paris, Clarksville, and Hugo.	Cities in Texas
Surnom	There is a baseball team in the Big Apple.	New York

(ou polygone) renvoyé par le système Clavin n'est en intersection avec aucun des polygones associés à la phrase 31 dans les Qrels.

Enfin, ces n-uplets et les Qrels sont donnés dans le format idoine en entrées du programme `trec_eval` [10] de TREC qui calcule un ensemble de mesures d'évaluation (figure 3). Il en résulte les mesures de Précision, de Rappel et de F-mesure déterminées par TREC. Ces résultats nous permettent ensuite de construire des tableaux synthétiques de présentation des mesures par catégorie et par système de RENS.

Nous présentons ces mesures d'*effectiveness* de façon synthétique : toutes catégories confondues, d'une part, et pour chacune des quinze catégories, d'autre part.

3.2. Critères de performance (efficiency)

Nous proposons de mesurer le temps de traitement de l'ensemble des phrases toutes catégories confondues. Il est également intéressant de travailler sur le temps moyen de traitement d'une phrase, tout comme sur le temps moyen de traitement des phrases d'une catégorie donnée. Cet indicateur renseigne les utilisateurs de systèmes de RENS pour trouver l'équilibre adéquat entre qualité des résultats et temps de réponse acceptables pour les utilisateurs finals.

3.3. Mise à disposition du benchmark

Nous proposons la démarche d'utilisation totalement ouverte suivante :

– Le participant télécharge la liste de phrases associée au *benchmark* SNERBM ;

10. http://trec.nist.gov/trec_eval

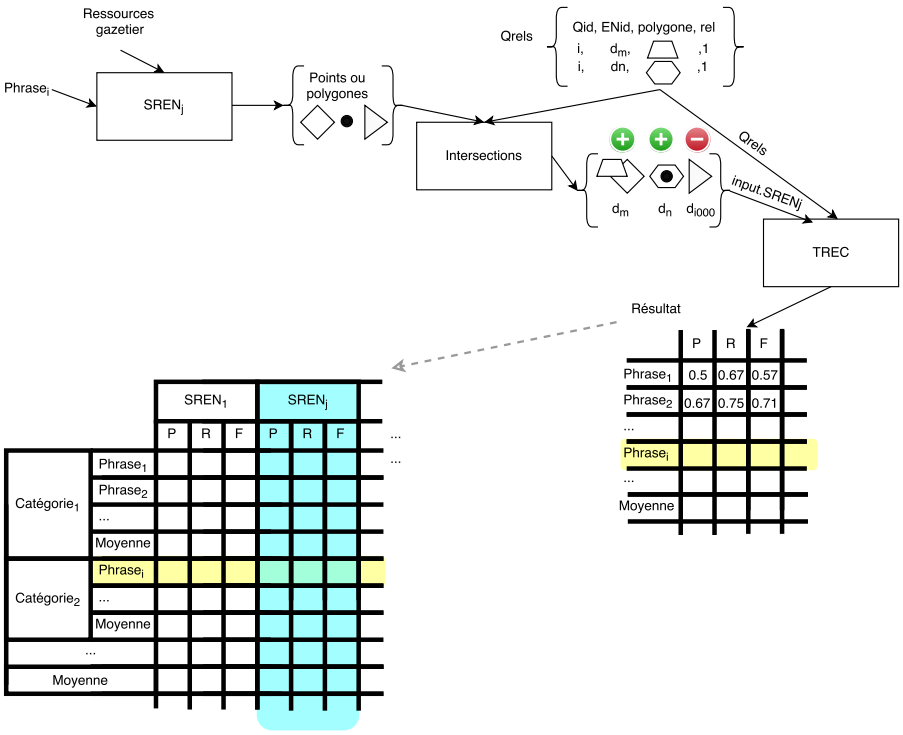


Figure 3. Processus d'évaluation des systèmes de RENS selon le benchmark SNERBM

– Pour chaque phrase, le participant calcule et produit, à l'aide de son système, des triplets (numéro de phrase, ENS, coordonnées géocodées) décrivant les ENS retrouvées et nous les communique afin d'obtenir un tableau synthétique des résultats ;

– À l'issue de chaque utilisation du *benchmark*, le participant est invité à proposer des catégories ou phrases supplémentaires qu'un groupe de modérateurs de SNERBM validera afin d'enrichir le jeu de données ou d'en créer de nouveaux.

Notons que, s'il le souhaite, le participant pourra télécharger les *qrels* et calculer ses résultats directement avec le programme `trec_eval`.

4. Évaluation de systèmes RENS avec SNERBM

Pour valider le cadre d'évaluation SNERBM et son utilisation comme référentiel commun, nous l'avons mis à l'épreuve en évaluant plusieurs systèmes existants : Clavin, Geodict, Geolocator, OpenCalais, Unlock et Yahoo!PlaceSpotter.

4.1. Mesure de l'efficacité (effectiveness)

Les résultats de l'évaluation du point de vue de l'*effectiveness* sont présentés dans le tableau 6. Pour un système donné, sont représentés la *Précision*, le *Rappel* et la *F1-mesure*. Le dernier jeu de mesures correspond à la meilleure combinaison de systèmes qui, pour chaque phrase, retient le système ayant proposé le meilleur résultat. Cette analyse de la performance globale (c.-à-d., sur l'ensemble des phrases du *benchmark*), au regard de l'*effectiveness*, classe les systèmes Yahoo!PlaceSpotter, Opencalais et Geolocator aux trois premières places respectivement.

Tableau 6. Analyse de la précision (P), du rappel (R) et de la F1-mesure (F) pour chaque système évalué

	Clavin			Geodict			Geolocator			Opencalais			Unlock			**Yps**			Meilleure combinaison		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Mesure	0,29	0,31	0,29	0,19	0,18	0,19	0,42	0,48	0,44	0,42	0,49	0,44	0,33	0,39	0,35	**0,66**	**0,66**	**0,65**	0,86	0,85	0,87
Classement	4	5	5	5	6	6	2	3	2	2	2	2	3	4	4	1	1	1			

Tableau 7. Analyse de la F1-mesure par système et par catégorie d'ENS

F1-Mesure	Clavin	Geodict	Geolocator	Opencalais	Unlock	Yps	Meilleure combinaison
Catégories d'ENS							
Abréviation	0,24	0,14	0,37	0,57	0,57	**0,91**	0,94
Abréviation administrative	0,87	0,60	0,93	0,73	0,00	**1,00**	1,00
Autre type de référence	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Divers	0,21	0,19	0,33	0,25	0,26	**0,50**	0,87
Nom avec diacritique	0,33	0,00	0,58	0,24	0,24	**0,80**	1,00
Nom avec typologie associée	**0,40**	0,20	0,33	0,00	0,13	0,20	0,75
Nom et contexte associé	0,16	0,30	0,12	0,29	0,17	**0,65**	0,66
Nom et niveau administratif associé	0,19	0,53	0,42	0,44	0,37	**0,84**	0,95
Nom historique	0,55	0,22	0,69	0,67	**0,88**	0,69	0,96
Nom officiel, court ou dérivé	0,82	0,51	0,82	0,83	0,62	**0,86**	0,97
Nom retor avec caractères numériques	0,00	0,00	0,17	0,00	0,00	**0,33**	1,00
Nom retor avec caractères spéciaux	0,20	0,05	0,60	0,67	0,33	**0,71**	1,00
Nom sans typologie associée	0,40	0,40	**0,42**	0,00	0,00	0,33	1,00
Nom unique aux États-Unis	0,57	0,00	0,69	**1,00**	0,92	0,50	1,00
Séquence de noms avec caractéristiques communes	0,31	0,17	0,40	0,41	**0,61**	0,46	0,56
Surnom	0,50	0,00	0,00	0,00	0,00	**0,67**	1,00

L'analyse des résultats par catégorie montre des catégories d'ENS correctement analysées et d'autres générant un fort taux d'échec. Le tableau 7 et la figure 4 présentent les meilleurs résultats obtenus pour chaque catégorie. Les différentes dénominations administratives, les abréviations et les noms officiels, par exemple, sont des catégories analysées avec une F1-mesure supérieure à 80 %. Sous la barre des 50 %, les références autres (de type GPS, par exemple) et les homonymes désignant des lieux différents sont difficiles à détecter et à analyser, y compris en présence de la typologie associée (nom avec typologie associée). De même, les noms avec des caractères numériques ou sans typologie associée sont difficiles à reconnaître et à analyser.

Pour un corpus hétérogène en termes de catégories, des combinaisons de systèmes sont envisageables afin d'améliorer la qualité des résultats de RENS. En effet, il existe presque toujours une complémentarité des systèmes qui permet d'envisager des combinaisons avantageuses (figure 4). Pour chaque phrase d'une catégorie donnée, nous avons observé les résultats des différents systèmes. Ainsi, la combinaison de

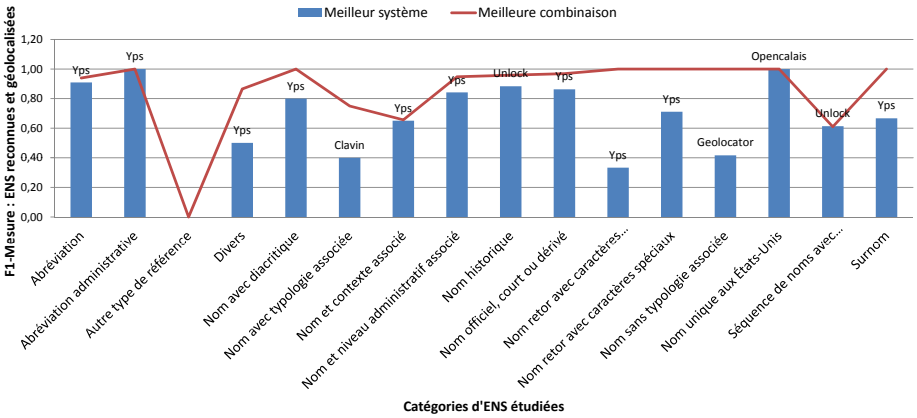


Figure 4. Meilleurs résultats par catégorie d'ENS

systèmes, pour une phrase, consiste à considérer le système donnant le meilleur résultat et, pour une catégorie, la moyenne des meilleurs résultats ainsi obtenus.

L'analyse qualitative des erreurs montre que tous les systèmes rencontrent des difficultés sur les mêmes phrases des catégories « Autre type de référence » et « Nom retors avec caractères numériques ». La catégorie « Nom retors avec caractères numériques » est singulière car chaque phrase comporte une énumération de noms de lieux. Une combinaison de systèmes permet dans ce cas d'améliorer considérablement la qualité des résultats : les systèmes sont bien complémentaires pour cette catégorie.

Pour un corpus donné, il sera donc intéressant d'analyser les résultats relatifs à différentes catégories (voir exemple figure 4) afin de préconiser un système de RENS particulier ou une combinaison de systèmes.

4.2. Mesure de la performance (efficiency)

Nous comparons les systèmes en termes de temps de traitement de la collection. Comme le montre la figure 5, d'importantes différences de temps de réponse sont observées : de 8 secondes à 51 minutes. Tous les traitements ont été effectués sur une machine Ubuntu 12.04 64 bits dotée d'un CPU simple cœur, de 4 Go de RAM et de 100 Go de disque dur.

Unlock est le système le plus lent. Ceci est dû au fait que le service web ne retourne pas directement les résultats mais propose un fonctionnement par lot (*batch*). Yahoo!Placespotter est le service web le plus rapide avec moins de 143 secondes pour l'ensemble de la collection. Clavin est plus rapide avec seulement 8 secondes de traitement. Contrairement à Yahoo!Placespotter, il est installé en local sur la machine qui réalise les traitements. Toutefois, Geolocator, qui est aussi installé sous la forme d'une application locale, nécessite 837 secondes pour la même collection.

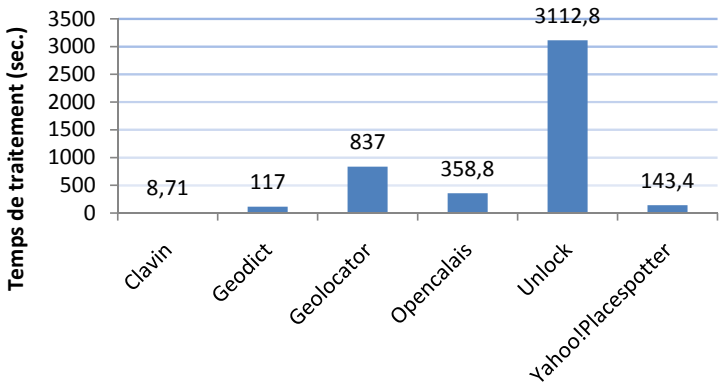


Figure 5. Temps de traitement de la collection

Le tableau 8 reprend des caractéristiques de ces systèmes. Nous avons choisi ces prototypes afin de comparer des systèmes avec des architectures et des stratégies de mises en œuvre différentes.

Tableau 8. Avantages et limites de chaque système

Système	Avantages	Limites
Clavin	Open source, local, très rapide, peu d'espace requis : RAM 1GB, DD 4GB	
Geodict	Open source (server image disponible) pas de limitation en nombre de requête	Service web limité pour trop de requêtes simultanées
Geolocator	Open source, local, espace requis : DD 7GB	>3GB de mémoire RAM, en cours de développement (bugs)
Opencalais		Clé limitée à 50K requêtes par jour (4 par seconde)
Unlock	Nombre de requêtes illimité	Traitements asynchrones
Yahoo!Placespotter	Rapide, peut traiter des URL directement	2K par jour et par IP ou clé limitée à 100K par jour

4.3. Synthèse

Cette synthèse reprend les mesures de temps de réponse et de F1-mesure sous la forme d'un diagramme composé de barres horizontales (figure 6). Nous avons défini la T-Mesure (équation 1), normalisée entre 0 et 1, en étendant la formule proposée dans (Lee, 1997). Ainsi, l'équation 1 calcule la performance (*efficiency*) T_i d'un système i relativement aux durées d'exécution $t_{1 \leqslant j \leqslant n}$ des n systèmes testés. Étant donné l'écart très important des temps de réponse observé pour les différents systèmes (figure 5), nous avons intégré un seuil pour mieux discriminer les performances desdits systèmes : $\forall i \in [1, n] \quad t_i \leftarrow \min(t_i, \text{seuil})$. Dans notre cas, ce seuil a été fixé à 500 secondes.

$$T_i = 1 - \frac{t_i - \min_{1 \leqslant j \leqslant n}(t_j)}{\max_{1 \leqslant j \leqslant n}(t_j) - \min_{1 \leqslant j \leqslant n}(t_j)} \in [0, 1] \qquad (1)$$

Les systèmes Clavin, Yps et Geodict maximisent les deux critères (figure 6) : le plus grand cumul correspond au meilleur résultat. Aussi, le système Clavin, qui offre

des traitements très rapides, malgré des résultats moyens en termes de F1-mesure, est-il au coude à coude avec le système Yps. Ces résultats globaux sont à considérer avec prudence toutefois. En effet, il est clair que mélanger *effectiveness* et *efficiency* présente des biais : l'un ne « rattrape » pas vraiment l'autre. Le critère *effectiveness* devrait sans doute être privilégié par une pondération plus importante.

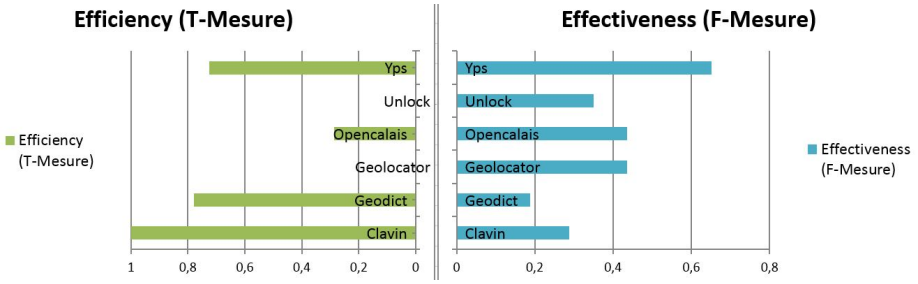


Figure 6. Synthèse des mesures de performance (efficiency) et d'efficacité (effectiveness) – plus le pourcentage est élevé, meilleure est la performance du système

5. Conclusion

Le cadre d'évaluation de système de RENS que nous avons appelé *benchmark* SNERBM est ouvert. Comme nous l'avons souligné, tout nouveau système peut être évalué selon le même protocole et comparé aux précédentes *baselines*. SNERBM est extensible, c'est-à-dire ouvert à la contribution : les contributeurs peuvent proposer de nouvelles phrases ou catégories d'ENS. La mise en ligne du *benchmark* SNERBM est en cours. De même, il nous paraît intéressant, à moyen terme, d'en proposer une version multilingue en conservant les mêmes *qrels* comme éléments de départ.

Enfin, ce travail présente un premier résultat de comparaison de systèmes de RENS disponibles en ligne. Au delà du comparatif global de systèmes, il pointe les catégories d'ENS correctement reconnues et analysées ainsi que celles encore mal gérées par les différents systèmes de RENS.

Remerciements. Nous remercions tout particulièrement Judith Gelernter de l'Université Carnegie Mellon et Doug Caldwell du U.S. Army Topographic Engineering Center (USATEC) pour avoir mis à notre disposition le jeu de test GeoparsingQT.

Bibliographie

Anastácio I., Martins B., Calado P. (2010). Using the geographic scopes of web documents for contextual advertising. In *GIR'10: Proceedings of the 6th Workshop on Geographic Information Retrieval*, p. 18:1–18:8. ACM.

Andogah G. (2010). *Geographically Constrained Information Retrieval*. Thèse de doctorat, University of Groningen, Netherlands.

Berjawi B., Duchateau F., Favetta F., Miquel M., Laurini R. (2015). PABench: Designing a Taxonomy and Implementing a Benchmark for Spatial Entity Matching. *GEOProcessing'2015: The 7th International Conference on Advanced Geographic Information Systems, Applications, and Services*, p. 7-16.

Bucher B., Clough P., Joho H., Purves R., Syed A. K. (2005). Geographic IR Systems: Requirements and Evaluation. In *ICC'05: Proceedings of the 22nd International Cartographic Conference.* Global Congressos. (CDROM)

Chinchor N. A. (1998). MUC/MET evaluation trends. In *Proceedings of the TIPSTER text program: Phase III*, p. 235–239. Association for Computational Linguistics.

Chinchor N. A. (1998). Overview of MUC-7. In *MUC-7: Proceedings of the 7th Message Understanding Conference.*

Dean-Hall A., Clarke C. L. A., Kamps J., Thomas P., Simone N., Voorhees E. (2013). Overview of the TREC 2013 Contextual Suggestion Track. In *TREC'13: Proceedings of the 22nd text retrieval conference.* NIST.

D'Ignazio C. (2013). *Big data, news and geography: Research update.* Consulté sur https://civic.mit.edu/blog/kanarinka/big-data-news-and-geography (MIT Center for Civic Media)

Gelernter J., Zhang W. (2013). Cross-lingual geo-parsing for non-structured data. In *GIR'13: Proceedings of the 7th Workshop on Geographic Information Retrieval*, p. 64–71. ACM.

Grover C., Tobin R., Byrne K., Woollard M., Reid J., Dunn S. *et al.* (2010). Use of the Edinburgh geoparser for georeferencing digitized historical collections. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, n° 1925, p. 3875–3889.

Lee J. H. (1997). Analyses of Multiple Evidence Combination. In *SIGIR'97: Proceedings of the 20th annual international ACM SIGIR conference*, p. 267–276. ACM Press.

Leidner J. L. (2007). *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. Thèse de doctorat, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, Scotland.

Lenzi V. B., Speranza M., Sprugnoli R. (2013). Named entity recognition on transcribed broadcast news at EVALITA 2011. In *Revised Papers from EVALITA'11: International Workshop on the Evaluation of Natural Language and Speech Tools for Italian*, vol. 7689, p. 86-97. Springer.

Lieberman M. D., Samet H., Sankaranarayanan J. (2010). Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proceedings of the 26th International Conference on Data Engineering, ICDE*, p. 201–212. IEEE.

Lingad J., Karimi S., Yin J. (2013). Location extraction from disaster-related microblogs. In *Proceedings of the 22nd International Conference on World Wide Web*, p. 1017–1020. ACM.

Mandl T., Carvalho P., Nunzio G. M. D., Gey F. C., Larson R. R., Santos D. *et al.* (2009). GeoCLEF 2008: The CLEF 2008 Cross-Language Geographic Information Retrieval Track Overview. In *Revised Selected Papers of CLEF'08: Proceedings of the 9th Workshop of the Cross-Language Evaluation Forum*, vol. 5706, p. 808–821.

Manning C. D., Raghavan P., Schütze H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marrero M., Sánchez-Cuadrado S., Lara J. M., Andreadakis G. (2009). Evaluation of named entity extraction systems. *Advances in Computational Linguistics. Research in Computing Science*, vol. 41, p. 47–58.

Marrero M., Urbano J., Sánchez-Cuadrado S., Morato J., Berbís J. M. G. (2013). Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces*, vol. 35, nº 5, p. 482-489.

Morana A., Morel T., Berjawi B., Duchateau F. (2014). Geobench: a geospatial integration tool for building a spatial entity matching benchmark. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, p. 533–536. ACM.

Nouvel D. (2012). *Reconnaissance des entités nommées par exploration de règles d'annotation*. Thèse de doctorat, Université François Rabelais de Tours, France.

Sallaberry C. (2013). *Geographical information retrieval in textual corpora*. Wiley-ISTE.

Santos D., Seco N., Cardoso N., Vilela R. (2006). HAREM: An Advanced NER Evaluation Contest for Portuguese. In *LREC'06: Proceedings of the 5th International Conference on Language Resources and Evaluation*, p. 1986–1991.

Sekine S., Eriguchi Y. (2000). Japanese Named Entity Extraction Evaluation – Analysis of Results. In *COLING'00: Proceedings of the 18th conference on Computational linguistics*, p. 1106-1110. Association for Computational Linguistics.

Strassel S., Przybocki M. A., Peterson K., Song Z., Maeda K. (2008). Linguistic Resources and Evaluation Techniques for Evaluation of Cross-Document Automatic Content Extraction. In *LREC'08: Proceedings of the International Conference on Language Resources and Evaluation*, p. 2706–2709.

Tjong Kim Sang E. F., De Meulder F. (2003). Introduction to the CoNLL-2003 Shared Task Language-Independent Named Entity Recognition. In *CoNLL-2003: Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147. Association for Computational Linguistics.

Tobin R., Grover C., Byrne K., Reid J., Walsh J. (2010). Evaluation of georeferencing. In *GIR'10: Proceedings of the 6th Workshop on Geographic Information Retrieval.* ACM.

Vaid S., Jones C. B., Joho H., Sanderson M. (2005). Spatio-textual Indexing for Geographical Search on the Web. In *SSTD'05: Proceedings of the 9th international Symposium on Spatial and Temporal Databases*, vol. 3633, p. 218–235. Springer.

Voorhees E. M. (2001). Overview of TREC 2001. In *TREC'01: Proceedings of the 9th Text REtrieval Conference.* NIST.

Voorhees E. M. (2002). The philosophy of information retrieval evaluation. In *CLEF'01: Proceedings of the Second Workshop of the Cross-Language Evaluation Forum*, vol. 2406, p. 355–370. Springer.

Voorhees E. M. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Communications of the ACM*, vol. 50, nº 11, p. 51–54.

Détecter et monitorer les séismes grâce aux capteurs embarqués dans les smartphones

Anne-Marie Lesas [1, 2]

1. Université de Nice – Sophia-Antipolis, MBDS,
1645, route des Lucioles, Sophia Antipolis, 06410 Biot, France
2. Aix Marseille Université, LSIS UMR 7296,
13397 Marseille, France
amlesas@yahoo.fr

RESUME. Dans cet article, nous présentons un algorithme original de détection des séismes utilisant le capteur accéléromètre embarqué dans le smartphone. Nos travaux s'inscrivent dans le cadre du projet SISMAPP d'études et de recherche proposé aux étudiants du MBDS (www.mbds-fr.org) de l'Université de Nice – Sophia-Antipolis (UNS) en partenariat avec le Centre Sismologique Euro-Méditerranéen (CSEM, www.emsc-csem.org). Notre objectif est de montrer que la connectivité du smartphone et les capteurs embarqués pourraient devenir de facto une station sismique mobile pouvant être aisément déployée à grande échelle et à faible coût.

ABSTRACT. In this paper, we present an original algorithm for detecting earthquakes using the accelerometer sensor embedded into the smartphone. Our work fits within SISMAPP research project proposed to MBDS (www.mbds-fr.org) students at University of Nice – Sophia-Antipolis (UNS) in partnership with the European-Mediterranean Seismological Centre (EMSC, www.emsc-csem.org). We aim to show that the smartphone connectivity and embedded sensors could turn into a mobile seismic station that can be easily and widely deployed at low cost.

MOTS-CLES : Surveillance des tremblements de terre, Application mobile, Capteurs embarqués, Big Data.

KEYWORDS: Earthquake monitoring, Mobile application, Embedded sensors, Big Data

1. Introduction

En moins d'une décennie, les téléphones mobiles ont évolué jusqu'à rivaliser avec les ordinateurs de bureau. De par sa popularité (près de 2 milliards de

smartphones vendus dans le monde fin 2015 [1]) et sa connectivité (réseau cellulaire, Wi-Fi, Bluetooth), le smartphone qui embarque toutes sortes de capteurs dont l'accéléromètre et le GPS, permet d'analyser et de collecter des informations géolocalisées et horodatées. Au mois d'août 2014, l'enregistrement d'un séisme de magnitude 6 survenu dans le Sud de Napa (Californie, US) a été capturé par hasard à 38 km de l'épicentre avec l'application « MyShake » développée dans le laboratoire de sismologie de l'université de Berkeley (www.seismo.berkeley.edu). Les enregistrements du séisme de Napa avec une application mobile ont démontré que les données en provenance des capteurs du smartphone sont d'une grande qualité lorsque l'appareil est positionné sur un support stable, et qu'elles pourraient être exploitées par les sismologues.

Depuis 2013 les étudiants du Master Mobiquité Bases de Données et intégration de Systèmes (MBDS) de l'Université de Nice – Sophia-Antipolis (UNS) contribuent au projet SISMAPP de prototypage d'outils numériques dédiés à une plateforme de gestion avant / pendant / après un séisme en partenariat avec le Centre Sismologique Euro-Méditerranéen (CSEM, www.emsc-csem.org) et le suivi des membres du laboratoire de recherche en Géophysique GéoAzur (www.geoazur.oca.eu) du Centre National de la Recherche Scientifique (CNRS) Unité Mixte de Recherche (UMR) de l'UNS.

Dans cet article, nous montrons comment une application mobile peut détecter et déclencher des alertes d'origine potentiellement sismiques et collecter des mesures en utilisant le capteur accéléromètre embarqué dans le smartphone. Les données collectées sont destinées à alimenter un entrepôt de données massives destiné à l'étude de modèles de prédictibilité : les travaux apparentés ainsi que les motivations pour le travail présenté dans cet article sont abordés dans la section 2. Dans la section 3, le lecteur est familiarisé avec les capteurs embarqués dans les smartphones. Nous décrivons notre algorithme basé sur le dépassement de seuils dans la section 4. À la fin de cet article, nous concluons en donnant quelques perspectives d'évolution de nos travaux.

2. Etat de l'art et motivations

2.1. Travaux apparentés

Depuis 2008, le projet Quake Catcher Network (QCN, www.qcn.stanford.edu), initié par les universités de Stanford et de Californie à Riverside, utilise le logiciel libre de calcul réparti « Berkeley Open Infrastructure for Network Computing » (BOINC) développé par l'université de Berkeley en Californie (USA). BOINC est une plateforme open source hébergeant des projets scientifiques dont le principe est d'utiliser une partie des ressources des systèmes informatiques des utilisateurs bénévoles connectés au réseau. La fiabilité de l'interprétation des données

1. Source Cbnews : http://www.cbnews.fr/digital/pres-de-2-milliards-de-smartphone-dans-le-monde-fin-2015-a1016742

remontées par le réseau QCN a été démontrée avec des données collectées par des capteurs inertiels fixés dans les parties inférieures des bâtiments à Christchurch en Nouvelle Zélande (Cochran et al., 2011) : ces données comparées aux données des stations sismiques GeoNet (www.geonet.org.nz) de la région lors du tremblement de terre survenu le 18 octobre 2011, ont montré que les mouvements du sol observés par GeoNet et QCN avaient des amplitudes comparables à une distance donnée de l'épicentre du séisme.

Le CSEM propose deux programmes dans le cadre de QCN : le programme « QCN Sensor Monitoring Program » qui fonctionne en mode « déclenchement », c.-à-d. que l'information n'est transmise que lorsque l'accélération est significativement plus élevée que celle des secondes précédentes en se basant sur l'algorithme « Short-Term Average/Long-Term Average » (STA/LTA) (Cuenot, 2003), et le programme « QCN Continual Monitoring Program » fonctionnant en mode « continu » pour collecter des échantillonnages de données.

La plupart des applications mobiles dédiées aux tremblements de terre s'appuient sur les données en provenance des organismes de surveillance sismique tels que l'US Geological Survey (USGS), le CSEM, ou l'Earthquake Early Warning (Japan Meteorological Agency) dont les informations qui proviennent majoritairement de stations sismiques fixes implantées sur toute la surface de la planète sont accédées à la demande en temps décalé ou semi-réel (requêtes continues avec ou sans notification d'alerte). Les applications mobiles utilisant les capteurs embarqués des smartphones ont émergé récemment avec le prototype japonais i-Jishin (Naito et al., 2012) (Naito et al., 2013) et le projet « iShake » de Berkeley (Reilly et al., 2013). En janvier 2014, QCN a publié son application mobile utilisant l'accéléromètre du smartphone et fonctionnant avec la version Android de BOINC toujours avec un déclenchement au repos (en charge). Cette application ne consomme que 1 à 5 % des ressources (CPU) de l'appareil et seuls quelques kilo-octets de données sont transmis quotidiennement sur le réseau.

Des applications mobiles telles que l'application Japonaise « Yurekuru Call » proposée par le centre Earthquake Early Warning sont spécialisées dans la propagation d'alertes précoces. Des applications ou des réseaux sociaux (par exemple Facebook) proposent également un bouton « je suis en vie » qui permet aux familles d'être rassurées en cas de sinistre. En outre, il existe des applications mobiles (e.g. « LastQuake » proposée par le CSEM) et des portails web permettant une contribution participative (« crowdsourcing ») du grand public pour partager des témoignages, des photos et des vidéos...

2.1. Motivations pour nos travaux

La détection basée sur les capteurs embarqués dans les smartphone est un nouveau champ d'étude, par exemple, une étude scientifique menée par plusieurs chercheurs (Minson et al., 2015), mentionne qu'un séisme pourrait être détecté en 5 secondes avec seulement une participation communautaire de 0,2 % des utilisateurs de mobiles (~5000 appareils) dans un rayon de 5 km autour de l'épicentre d'une

zone fortement peuplée comme San Francisco ou San Jose (aux USA). L'article précise que la détection n'est possible que lors des tremblements de terre de forte magnitude : la précision des capteurs mobiles limite la détection à des séismes de magnitude supérieure à 2. Cependant, les technologies évoluent plus vite que les usages et les limitations d'aujourd'hui ne doivent pas restreindre la recherche.

« *La région Provence-Alpes-Côte d'Azur est la région de France métropolitaine la plus soumise au risque sismique* » (Virieux, 2014). Déjà sensibilisés par le tremblement de terre dévastateur survenu à Haïti en 2010 et ayant entraîné la destruction des locaux du Master MBDS délocalisé, la contribution du MBDS au développement d'outils numériques autour du projet SISMAPP de gestion des séismes coulait de source. Le travail présenté dans cet article sur la détection de mouvements d'origine potentiellement sismique à partir des mesures collectées sur le capteur accéléromètre embarqué dans le smartphone s'inscrit dans le cadre d'un ensemble de travaux réalisés autour du projet SISMAPP présenté dans un rapport antérieur (Lesas et al., 2014). Le projet SISMAPP a pour double finalité de permettre aux étudiants d'exploiter leurs compétences dans la mise en pratique des enseignements théoriques avec à terme, l'implémentation et le déploiement d'une plateforme régionale dédiée à la recherche et à l'étude des modèles de prédictibilité en partenariat avec les chercheurs en sismologie et les industriels spécialisés dans le traitement des données en provenance des capteurs.

Dans cet article nous nous focalisons sur la faisabilité de l'utilisation des capteurs du smartphone dans la détection des séismes à travers la présentation de notre implémentation d'un prototype preuve de concept. La valeur ajoutée de notre contribution partiellement présentée et centrée ici sur l'analyse des données en provenance du capteur accéléromètre embarqué tient essentiellement dans la flexibilité ainsi qu'à la possibilité de collecter non seulement localement mais aussi de transmettre les données à un serveur distant ce qui en fait un appareil de mesure portable particulièrement adapté pour la recherche. En effet, contrairement aux autres applications, la solution proposée permet de tester plusieurs configurations et d'enregistrer des notifications d'alerte potentiellement sismiques ainsi que des mesures collectées dans un laps de temps prédéfini avant et après les alertes. Les données sont enregistrées dans une (micro) base de données (BDD) et/ou envoyées vers un serveur (dont l'URL est également paramétrable) au format texte de type clé-valeur (JSON). Le serveur met simplement à disposition un web service de type « Representational State Transfer » (REST) acceptant du texte sans contrainte de format de façon à permettre les évolutions ultérieures sans aucun impact sur l'interface : cette architecture a été conçue dans le but de pouvoir recueillir des données hétérogènes de différents formats et de différentes origines pouvant faire l'objet d'un traitement map reduce (Hadoop) toujours dans le cadre du projet SISMAPP.

3. Les capteurs

Les capteurs sont des instruments de mesure qui transforment une information d'ordre physique, chimique ou organique en donnée manipulable et interprétable.

Les capteurs sismiques sont des capteurs qui mesurent des inerties par l'accélération perçue sur un axe de l'espace à une fréquence d'échantillonnage donnée.

3.1. Les capteurs MEMS [2]

Les capteurs MEMS sont des systèmes micro-électro-mécaniques économiques de très petite taille (tendant vers les nanotechnologies) qui utilisent des propriétés électromagnétiques, thermiques, optiques, chimiques ou biologiques pour effectuer et collecter une mesure.

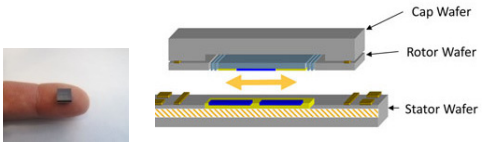


Figure 1. Capteur sismique MEMS Hewlett Packard (Source HP [3])

Les capteurs sismiques MEMS (e.g. fig. 1) sont généralement constitués d'accéléromètres qui mesurent les variations du champ de force exercé sur une petite masse témoin (seconde loi du mouvement de Newton : $F = m.a$), appelée aussi « masse sismique », se mouvant entre des lamelles fixes à la surface d'un circuit intégré en silicium. Il est ainsi possible de mesurer des accélérations horizontales et verticales en fonction de la position. Les capteurs triaxiaux retournent une valeur scalaire (x, y, z) qui correspond à la magnitude du vecteur d'accélération dans l'espace tridimensionnel.

3.2. Les capteurs de mouvements embarqués dans les smartphones

La plupart des smartphones embarquent en standard des capteurs inertiels mesurant les mouvements sur les 3 coordonnées spatiales relatives à l'appareil mobile (axe vertical, frontal et latéral). La restitution des mouvements dans le système géodésique se fait par translation des coordonnées (à l'aide d'un quaternion ou d'une matrice de rotation). L'inertie est mesurée par des capteurs physiques tels que l'accéléromètre qui renvoi un vecteur d'accélération en $m.s^{-2}$ (gravité incluse), le magnétomètre duquel est issu un vecteur de direction en micro-tesla [4], le gyroscope mesurant l'effet de Coriolis en rad/s restitué sous la forme d'un vecteur de vitesse angulaire par rapport aux axes, et le capteur de pression atmosphérique mesurée en hectopascals (millibars).

2. Micro-Electro-Mechanical Systems
3. http://www8.hp.com/us/en/hp-news/press-kit.html?id=1096990
4. Pour avoir une mesure correcte dans le temps, il faut normaliser le signal en prenant un échantillonnage de la vitesse angulaire, c.-à-d. la racine de la somme des carrés

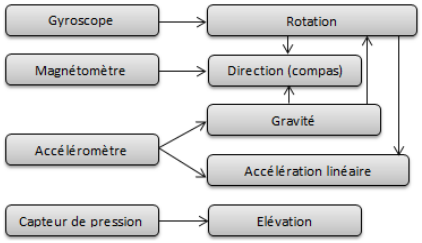


Figure 2. Capteurs fusion (Android)

La fusion des informations en provenance des capteurs physiques fournit une couche d'abstraction logicielle du matériel dont les relations illustrées par la fig. 2 montrent les dépendances de la source vers son utilisation entre les capteurs physiques (à gauche) et les capteurs « fusion » (à droite) issus de calculs combinant des mesures relevées sur les capteurs sources. Par exemple, le vecteur de rotation indique la direction avec l'azimut (autour de l'axe vertical), le pitch (autour de l'axe frontal) et le roll (autour de l'axe latéral) en degrés (fig. 3), la répartition de la gravité (g = 9,81 $m.s^{-2}$) sur les axes est mesurée $m.s^{-2}$ par combinaison des données du vecteur d'accélération et celui de la direction, l'accélération linéaire est calculée en soustrayant la gravité à l'accélération et l'élévation en mètres est déduite du capteur de pression. Les capteurs de mouvement sont présents sur tous les smartphones modernes, avec cependant des APIs qui diffèrent au niveau de la couche logicielle/middleware et du prétraitement des données (i.e. filtrage et fusion des données), par exemple, l'API Windows Phone n'est pas utilisable dans un service de tâche de fond [5].

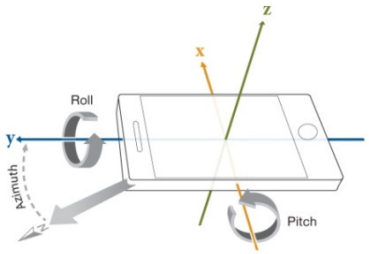


Figure 3. Axes des coordonnées géospatiales du smartphone [6]

5. Source de l'image : http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202962%28v=vs.105%29.aspx#BKMK_UnsupportedAPIs
6. Source : http://www.mathworks.com/matlabcentral/fileexchange/screenshots/9373/original.jpg

3.3. Sources d'erreurs et correction des mesures

L'une des préoccupations principales dans la manipulation des capteurs concerne l'élimination du bruit :

– l'accéléromètre présente un décalage constant entre la valeur produite et la valeur réelle qui se cumule au fil du temps et croît de façon quadratique lors des doubles intégrations,

– le décalage dans les données du gyroscope provoque une erreur angulaire qui croît linéairement dans le temps, mais une erreur du facteur d'échelles dans le calibrage, peut ajouter une dérive aux calculs basés sur les mesures faussées se répercutant dans les capteurs fusion,

– le magnétomètre est soumis aux interférences provoquées par la contamination magnétique environnante (proximité de matériaux ferreux)...

Mais la dérive engendrée par ces erreurs peut être atténuée significativement après détermination du coefficient de décalage (Shala et al., 2011) en fonction du modèle de capteur. La fusion des données multi-capteurs implémente également des filtres issus du traitement du signal (e.g. passe-bas, Kalman, Butterworth) qui permettent d'éliminer le bruit et de compenser des dérives.

4. Le service mobile SISMAPP

Le service mobile SISMAPP (Android) analyse en temps réel les données en provenance du capteur accéléromètre triaxial embarqué dans le smartphone afin de détecter des mouvements susceptibles de provenir d'un tremblement de terre. Pour ne pas impacter la consommation de la batterie (cf. tab. 1) et limiter la consommation des données, le service n'est déclenché que lorsque l'appareil est en charge et qu'il est connecté à un réseau Wi-Fi. L'analyse n'est déclenchée que lorsque l'état stationnaire du mobile est détecté.

Tableau 1. Consommation de batterie supplémentaire due à l'utilisation des capteurs et du GP

Utilisation en continu (Google Nexus S)	Durée pour 1% de conso. batterie	Durée pour 100% de conso. batterie	Conso. sup. (%)
Normale (78% de batterie)	00:03:28	05:46:40	-
Accéléromètre	00:03:01	05:01:40	+13%
Accéléromètre + GPS	00:01:38	02:43:20	+53%

Le service SISMAPP s'inspire du principe de détection d'événements par franchissement de seuils couramment utilisé dans les systèmes d'alerte. Cette technique permet de déterminer efficacement une anomalie sur un réseau distribué suffisamment dense (Cuenot, 2003). L'exécution en tâche de fond se base sur différents paramétrages que l'utilisateur peut modifier pour adapter la sensibilité en

fonction du résultat attendu : fréquence du capteur, accélération minimum pour un état stationnaire, accélération maximum pour un état stationnaire, variance minimum pour un état stationnaire, durée du temps de glissement des données analysées et collectée. Le paramétrage permet en outre de modifier les options de collecte et de notification : adresse du serveur et des web services, activation/désactivation du service ou de l'envoi des données, durée de la collecte, délais entre deux notifications, intervalle entre les localisations.

La méthode du franchissement de seuil sur une période est illustrée par l'algorithme (5) où les a_t sont les accélérations mesurées à l'instant t et $\delta(M)$ est le différentiel d'accélération qui déclenche le traitement :

$$\forall\, t > t_0, a \in [a_{t0} \ldots, a_{tn}], si\ |a_t - a_{t-1}| > \delta(M) \rightarrow \text{Traitement} \qquad (5)$$

4.1. Analyse et traitement des données en provenance de l'accéléromètre

Notre adaptation de la méthode du franchissement de seuil s'exécute sur une période glissante paramétrable très courte (~1s) dont le déclenchement est proposé dans l'algorithme 1 : lors du lancement de l'application, le service s'abonne aux événements de connexion et déconnection d'alimentation sur secteur, de changement de connectivité réseau (Wi-Fi), au service de localisation [7], et à l'événement d'activation et de désactivation du service accessible dans le setup de l'application. Lors de la réception des événements, la mise à jour d'indicateurs booléens conditionne le démarrage ou l'arrêt du traitement d'analyse et de collecte des données en provenance de l'accéléromètre.

Algorithme 1. Abonnement aux événements

```
1:  /* Abonnement aux événements qui conditionnent le traitement */
2:   AbonnerEvénements (alimentation, connectivitéRéseau, localisation, activationService)
3:
4:  /* Gestion du traitement en fonction des événements */
5:  RéceptionEvénement(Evénement)
6:  {
7:      Enregistrer l'état lié à l'événement
8:      si localisation alors
9:          /* Mise à jour de la localisation avec l'événement « localisation » */
10:         Mettre à jour la latitude et la longitude
11:     sinon
12:         si (enCharge ∧ connectéAuWiFi ∧ serviceActivé) alors
13:             /* Si les conditions sont réunie, l'analyse des données est démarrée */
14 :            début glissement ← horodatage
15:             durée collecte ← ∅
16:             maxX ← 0
17:             maxY ← 0
```

7 GPS et GPS fusion disponibles en fonction du réseau et de la couverture

```
18:                 maxX ← 0
19:                 stationnaire ← faux
20:                 collection ← faux
21:                 DémarrerLectureAccéléromètre(fréquence)
22:             sinon
23:                 /* Sinon, l'analyse est stoppée */
24:                 StopperLectureAnalyseAccéléromètre()
25:             finsi
26:     finsi
27: }
```

La lecture des données de l'accéléromètre se présente aussi sous la forme d'un abonnement aux événements de l'API capteurs à la fréquence choisie (cf. tab. 2) :

Tableau 2. Fréquence de lecture du capteur

Option de fréquence de lecture	Délais entre chaque lecture
SENSOR_DELAY_NORMAL	0,2 s
SENSOR_DELAY_UI	0,06 s (convient au rafraîchissement de l'affichage)
SENSOR_DELAY_GAME	0,02 s (convient aux jeux)
SENSOR_DELAY_FASTEST	Pas de délais : fréquence maximum

Les mesures des 3 axes x, y, et z du capteur accéléromètre (linéaire [8]) horodatées sont réceptionnées dans la méthode de rappel dans laquelle elles sont analysées et traitées selon l'algorithme 2 :

Algorithme 2. Analyse et traitement des données en provenance de l'accéléromètre

```
1:    /* Traitement d'analyse des données de l'accéléromètre */
2:  AnalyseValeursAccéléromètre(valeurs courantes de l'accéléromètre, horodatage)
3:  {
4:      x, y, z ← valeurs courantes de l'accéléromètre
5:      collecte ← {x, y, z, horodatage} ∪ collecte
6:
7:      /* Glissement de l'analyse sur les valeurs lues pendant la durée paramétrée */
8:      si (durée collecte > paramètre glissement) alors
9 :
10:         /* Vérification de l'état stationnaire : */
11:         si ¬stationnaire ∧ ¬collection alors
12:             stationnaire ← VérifierStationnaire(collecte)
13:         finsi
14:
15:         /* Lorsque l'état stationnaire a été détecté : */
```

8. Auquel on a soustrait la gravité.

```
16:             si stationnaire alors
17:                 /* Vérification du dépassement de seuil sur données collectées */
18:                 collection ← VérifierAlerte(collecte)
19:             finsi
20:
21:             /* Lorsque le dépassement de seuil a été détecté : */
22:             si collection alors
23:                 id_ collection ← id_ collection + 1
24:                 /* Envoyer notification d'alerte à l'utilisateur  */
25:                 TraiterAlerteUtilisateur(horodatage)
26:
27:                 /* Si le serveur est configuré, envoi de la notification au serveur */
28:                 TraiterAlerteServeur(horodatage, id_ collection, dernière localisation)
29:
30:                 /* Vérifier si la durée de collecte paramétrée est atteinte  */
31:                 si (durée collecte – paramètre  glissement) > paramètre collection alors
32:                         /* Traiter la collection et faire une pause */
33:                          collection ← faux
34:                         TraiterCollection(id_collection, collecte)
35:                         TraiterPause(horodatage)
36:                 finsi
37:             sinon
38:                     /* Supprimer les 1ères valeurs collectées pour ne garder */
39:                     /* que le temps de glissement paramétré et mettre à jour */
40:                     /* l'horodatage du début de la collecte...                */
41:                     TraiterGlissement(collecte)
42:             finsi
43:     finsi
44:
45:     durée glissement ← (horodatage – début glissement)
46:  }
```

L'algorithme 2 montre que les données horodatées des valeurs mesurées (x, y, et z) sont collectées dans une liste sur une période glissante ; le dépassement de seuil déclenchant une alerte et la sauvegarde de la collecte n'est vérifié qu'après détection de l'état stationnaire (cf. algo. 3) : s'il y a un dépassement de seuil (algo. 4), une notification d'alerte est déclenchée (dans la barre des notifications) et selon le paramétrage, envoyée au serveur. La collection est poursuivie sur la durée paramétrée avant d'être traitée selon l'algorithme 5. Les variables (état stationnaire et maximums des moyennes collectées à l'état stationnaire) sont réinitialisées, et le service est mis en pause pendant la durée paramétrée (ce qui évite les notifications intempestives et limite les dérives du capteur dans le temps). Si aucun franchissement du seuil d'alerte n'est détecté, les premières mesures sont supprimées de façon à ne conserver dans la collecte que la période de glissement.

Algorithme 3. Vérification de l'état stationnaire

```
1:    /* Vérification de l'état stationnaire */
2:    VérifierStationnaire(collecte)
```

```
3:   {
4:        ∀ x, y, z ∈ collecte :
5:        stationnaire ← faux
6:        avgX ← ∑ x > 0 / nb(x > 0)
7:        avgY ← ∑ y > 0 / nb(y > 0)
8:        avgZ ← ∑ z > 0 / nb(z > 0)
9:
10:       /* comparaison des valeurs au seuil minimum déclaré dans le setup */
11:     si (avgX < minAccel) ∨ (avgY < minAccel) ∨ (avgZ < minAccel) alors
12:
13:               /* si au moins une des valeurs moyennes mesurées sur un axe est          */
14:               /* inférieure à la valeur minimum, alors on considère l'état stationnaire */
15:               /* et on mémorise les valeurs moyennes maximum de l'état stationnaire */
16:                 stationnaire ← vrai
17:                 maxX ← avgX>maxX ?avgX :maxX
18:                 maxY ← avgY>maxY ?avgY :maxY
19:                 maxX ← avgX>maxX ?avgX :maxX
20:       finsi
21:  }
```

Afin d'exclure d'éventuelles valeurs parasites, l'état stationnaire est vérifié sur la moyenne de toutes les données collectées (période glissante) : si la moyenne mesurée sur un des axes est inférieure au seuil minimum pour un état stationnaire, c'est une condition suffisante pour considérer l'état stationnaire car cela permet de s'absoudre d'un éventuel décalage du zéro d'un des axes. Les moyennes maxima sont mémorisées pour calculer le delta d'accélération seuil d'alerte par rapport au delta configuré dans le setup (cf. algo. 4).

Remarque : Parce que la détection se fait en partant d'un état stationnaire et qu'une accélération positive est normalement suivie d'une accélération négative, seules les accélérations positives sont prises en compte.

Algorithme 4. Vérification du dépassement de seuil pour une « alerte »

```
1:    /* Vérification d'une alerte par une dérivée de la méthode du dépassement de seuil */
2:    VérifierAlerte(collecte)
3:    {
4:        ∀ x, y, z ∈ collecte :
5:        deltaX ← (∑ x > 0 / nb(x > 0)) – maxX
6:        deltaY ← (∑ y > 0 / nb(y > 0)) – maxY
7:        deltaZ ← (∑ z > 0 / nb(z > 0)) – maxZ
8:
9:      /* Si le delta moyen est supérieur à une valeur maximum paramétrée */
10:       /* c'est que le mouvement qui l'a provoqué est trop brusque être       */
11:     /* d'origine sismique...                                                */
12:       si (deltaX > maxAccel) ∨ (deltaY > maxAccel) ∨ (deltaZ > maxAccel) alors
13:
14:               stationnaire ← faux
15:
```

```
16:                 /* Il faut donc stopper une éventuelle collection */
17:                 si collection alors
18:                         /* Il faut donc stopper une éventuelle collection */
19:                         collection ← faux
20:
21:                         /* Mais la collection est tout de même traitée avant la pause */
22:                         TraiterCollection(id_collection, collecte)
23:                 finsi
24:                 TraiterPause(horodatage)
25:         sinon
26:                 /* Sinon, si le delta moyen est supérieur au delta d'accélération paramétré */
26:                 /* pour une alerte, la collection pour sauvegarde démarre                    */
27:                 si (deltaX > deltaAccel) ∨ (deltaY > deltaAccel) ∨ (deltaZ > deltaAccel) alors
28:                         collection ← vrai
29:                         début collection ← horodatage
30:                 finsi
31:         finsi
32:     }
```

Le déclenchement d'une alerte vérifie deux conditions (cf. algo. 4) : (i) que les deltas d'accélération moyens par rapport aux maxima moyens calculés à l'état stationnaire dans l'algorithme 3 ne dépassent pas un seuil trop élevé (e .g. dû à une reprise en main de l'appareil par l'utilisateur), et (ii) qu'au moins l'un des deltas mesurés sur les axes x, y, ou z dépasse un delta d'accélération minimum de déclenchement. Lorsque le delta d'accélération calculé dépasse un seuil trop élevé et qu'une collection était en cours, la collecte est stoppée mais les données sont quand même traitées pour une sauvegarde locale et/ou l'envoi au serveur (cf. algo. 5) car le dépassement pourrait avoir été engendré par la chute de l'appareil lors d'un séisme.

Algorithme 5. Traitement d'une collection de données consécutive à une alerte

```
1:     /* Vérification d'une alerte par une dérivée de la méthode du dépassement de seuil */
2:     TraiterCollection(id_collection, collecte)
3:     {
4:         si envoiServeur ∧ serveurRenseigné() alors
5:                 envoyerCollecte(id_collection, collecte, début glissement,
6:                                 début collection, horodatage, dernière localisation,
7:                                 type de capteur, fréquence, modèle de mobile)
8:         finsi
9:         enregistrerLocalement(id_collection, collecte)
10:    }
```

4.2. Traitement des alertes et des mesures collectées

Nous avons mentionné précédemment que lors d'un séisme, il se peut que les réseaux soient saturés ou endommagés. Dans la perspective d'une alternative à

l'incapacité de contacter le serveur et dans un contexte de recherche, la sauvegarde locale permet d'analyser les données collectées à postériori.

Les notifications d'alertes qui déclenchent la collecte et les mesures sont stockées localement dans une BDD embarquée (SQLite) accessible dans un dossier créé par l'application SISMAPP dans la mémoire de stockage du mobile (cf. fig. 4).

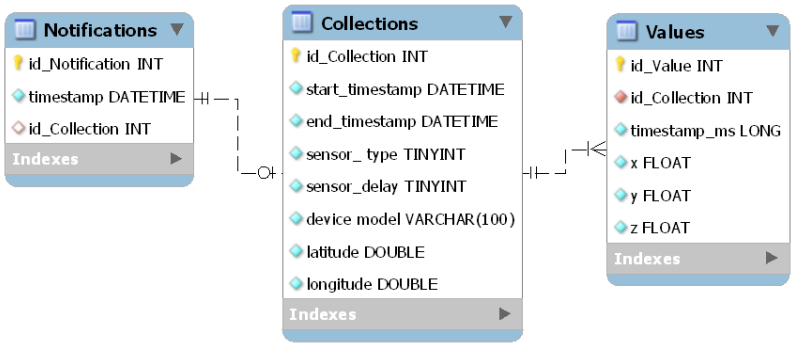


Figure 4. Modèle de la BDD locale du service SISMAPP

Les données collectées peuvent être supprimées à partir de l'application mobile ou bien récupérées pour être analysées (par exemple dans un tableur). Le temps *t* écoulé à l'occurrence *n* des valeurs collectées est donné en millisecondes ; il suffit alors de calculer en secondes les t_n d'un axe temporel horizontal démarrant à t_0 tel que : $t_0 = 0$ *et* $t_n = (t(n) - t(0)) * 10^{-9}$ et de sélectionner les séries de valeurs *x, y,* et *z* représentées en $m.s^{-2}$ sur l'axe vertical : la fig. 5 montre la visualisation graphique des données où le pic d'accélération correspond à une vibration du mobile (obtenue par un test) ayant déclenché une alerte et la sauvegarde de la collection.

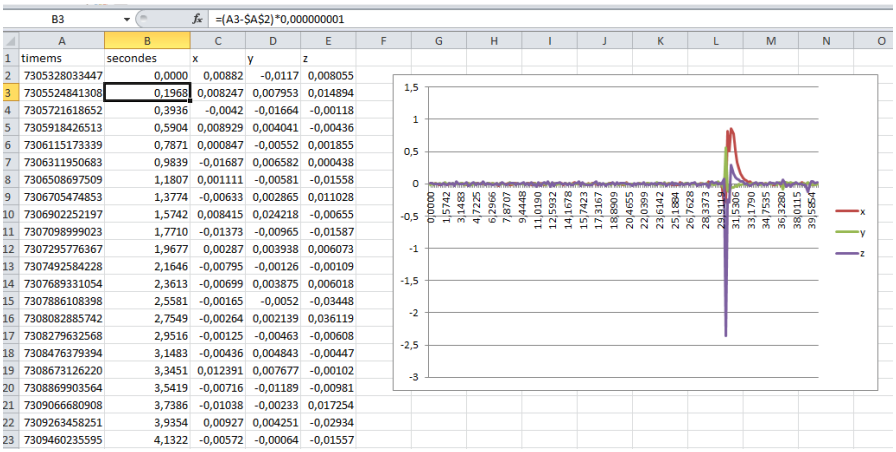


Figure 5. Visualisation graphique des données collectées

Si l'envoi au serveur a été paramétré et que le serveur est joignable, les notifications (en temps réel) et les collectes associées (à la fin de la collecte) peuvent être transmises dans le format clé-valeur (JSON) donné en exemple ci-après :

– Alertes : "{"notifications":{"id_device":1,"id_notification":4,
"timestamp":"2015-03-15_10:01:59","latitude":43.7055339,"longitude":
7.2820472, "id_collection":4}"

– Mesures : "{"collections":{"id_device":1, "id_collection":4,
"sensor_type":"Linear acceleration", "sensor_delay":3,
"start_timestamp":"2015-03-15_10:01:49", "end_timestamp":"2015-03-
15_10:02:19", "latitude":43.7055339, "longitude":7.2820472,
"device_model":"Android 21 - LGE NEXUS 5", "values":
[{"timestamp_ms":63382020019531, "x":0.0019194632768630981,
"y":1.209452748298645E-4, "z":0.0025548934936523438},
{"timestamp_ms":63382218017578, "x":0.0013000965118408203,
"y":0.006275080144405365, "z":0.02202129364013672},
{"timestamp_ms":63382416015625, "x":8.293241262435913E-4, "y":-
0.012210480868816376, "z":0.010548591613769531},
{"timestamp_ms":63382614013671, "x":-0.014498397707939148,
"y":0.00841280072927475, "z":0.02068042755126953},...]}

4.3. Expérimentation du prototype et limitations

4.3.1. Détermination des seuils

Dans un premier temps, la détermination des seuils de déclenchement des alertes s'est faite empiriquement par l'observation sur plusieurs modèles de smartphones (Google/Samsung Nexus S, Samsung Galaxy S3, Microsoft/Nokia Lumia 720, et Google/LG Nexus 5) : ne disposant pas de table vibrante, l'expérience consistait à détecter une très légère vibration (seuil minimum) du support (provoquée manuellement) sur un laps de temps arbitraire suffisant (1s) après détermination de l'état stationnaire. Le seuil maximum a quant à lui été déterminé en mesurant plusieurs reprises naturelles du mobile. À ce stade, une approche plus rationnelle de la détermination des seuils est envisagée en collaboration avec des partenaires industriels (par exemple dans le cadre d'un déploiement expérimental).

4.3.2. Perte de réseau en cas de destruction des infrastructures

Des applications comme « WhatsApp » (www.whatsapp.com), par exemple, ont démontré que la connectivité des smartphones avec le Bluetooth ou en Wi-Fi direct permet en agglomération de communiquer en pair-à-pair par réseau maillé lorsque les réseaux de télécommunication ne sont plus disponibles. Nous avons d'ailleurs prototypé les deux connectivités avec notre protocole qui permet de corroborer la présomption d'un séisme détecté par les capteurs du smartphone et d'alerter les utilisateurs lorsqu'un des utilisateurs a confirmé l'origine sismique. Cependant le système n'est fonctionnel que quand les appareils sont couverts par la connectivité utilisée.

4.3.3. Identification des faux positifs et corrélation des données

Nous partons du principe qu'il n'est pas possible d'éliminer les faux positifs individuels, c'est la raison pour laquelle nous portons toute notre attention sur la

corrélation des événements et le traitement massif des données. Plusieurs pistes sont à envisager ; l'étude des données collectées (en temps réel ou pas), à partir du moment où elles sont horodatées et géolocalisées, permet une analyse à postériori de laquelle pourraient être extraits des modèles de prédictibilité. Mais nous pensons également que la corrélation des sources d'informations et l'implantation d'un réseau de capteurs fixes communiquant avec le smartphone à des emplacements choisis pour leur faible exposition aux mouvements susceptibles d'engendrer des faux positifs est une solution adaptée pour les alertes précoces.

5. Conclusion

Dans cet article, nous avons montré comment grâce à un algorithme basé sur le franchissement de seuils, le smartphone avec ses capteurs inertiels embarqués et sa connectivité, peut contribuer avantageusement à la surveillance et la détection des séismes pour alerter les utilisateurs et collecter des données géolocalisées et horodatées pouvant être corrélées avec des événements sismiques dans l'objectif d'en déduire des modèles de prédictibilité.

Le service mobile SISMAPP est idéalement conçu pour dans un premier temps permettre la détermination d'un paramétrage efficace des seuils de sensibilité en laboratoire. Mais une phase de pilotage est nécessaire pour évaluer la fréquence des notifications et le volume des données pertinentes. L'exploitation du prototype SISMAPP dans le contexte de la recherche sismologique n'en est qu'à ses débuts, mais sa flexibilité en fait un outil de mesure portable qui pourrait trouver son utilité dans d'autres domaines, par exemple pour mesurer la résonnance des ondes d'origine explosive (e.g. dans des carrières de minerais).

Remerciements

Nous tenons à adresser ici tous nos remerciements à Rémy Bossu à la Direction du CSEM et à M. Frédéric Roussel du CSEM pour leurs conseils avisés, ainsi que l'ouverture d'un web service sur un serveur du CSEM nous permettant de tester l'envoi des données. Nous remercions également Anne Deschamps, Damienne Provitolo, et Emmanuel Tric du laboratoire GéoAzur pour leur intérêt et le suivi accordé au projet SISMAPP, ainsi que tous les étudiants qui ont participé au projet, le Pr. Serge Miranda pour avoir initié ce projet et le Pr. Omar Boucelma pour son encadrement et je remercie le comité de relecture et plus particulièrement le Pr. Florence Sedes pour ses critiques constructives et ses commentaires pertinents quant à la rédaction de cet article.

Bibliographie et références

Bossu R. (2015). *Earthquake Risk Mitigation Using Social Media and Sensor-Based Citizens' Participation*, internal report, CSEM, 2015.

Cochran E. S., Lawrence J. F., Kaiser A., Fry B., Chung A., Christensen C. (2011). Comparison between low-cost and traditional MEMS accelerometers: a case study from the M7.1 Darfield, New Zealand, aftershock deployment, *Annals Of Geophysics*, *Citizen*

Empowered Seismology, Special Section edited by R. Bossu and P.S. Earle, 2011, p.728-737.

Cuenot O. (2003). *Les algorithmes de détection automatique d'ondes sismiques*, probatoire CNAM, 2003.

Dolui K., Mukherjee S., Kanti Datta S. (2013). Smart Device Sensing Architectures and Applications, *International IEEE Computer Science and Engineering Conference (ICSEC), 2013*, Bangkok, Thailand.

Faulkner M., Olson M., Chandy R., Krause J. (2011). The Next Big One: Detecting Earthquakes and other Rare Events from Community-based Sensors, *International Conference on Information Processing in Sensor Networks (IPSN), 2011*, Chicago, Illinois, USA.

Kong Q. (2012). *Using Smartphones to Detect Earthquakes*, Berkeley Seismological Laboratory annual research report, University of California, 2012.

Lesas A.M., Ardoin A., Cano S. (2014). SISMAPP : *Une station sismique mobile dans le smartphone*, rapport technique, MBDS, Université de Nice – Sophia-Antipolis, 2014.

Minson S. E., Brooks B. A., Glennie C. L., Murray J. R., Langbein J. O., Owen S. E., Heaton T. H., Iannucci R. A., Hauser D. L. (2015). Crowdsourced earthquake early warning, *Science Advances*, vol. 1, n° 3, e1500036.

Naito S., Azuma H., Senna S., Yoshizawa M., Nakamura H., Fujiwara H., YoshidaM. (2013). *On-site experiment of seismic monitoring network by utilization inside sensors of mobile terminal*, National research Institute for Earth science and Disaster Prevention, Conference paper, Japan Geoscience Union Meeting, 2013.

Naito S., Azuma H., Senna S., Yoshizawa M., Nakamura H., Hao K. X., Fujiwara H., Hirayama Y., Yuki N., Yoshida M. (2013). Development and Testing of a Mobile Application for Recording and Analyzing Seismic Data, *Journal of Disaster Research*, vol. 8, n° 5, p. 990-1000.

Olson M., Liu A., Faulkner M. (2011). Rapid Detection of Rare Geospatial Events: Earthquake Warning Applications, *Distributed Events-Based Systems' conference, 2011*, New York, USA.

Reilly J., Dashti S., Ervasti M., Bray J. D., Glaser S. D., Bayen A. M. (2013). Mobile Phones as Seismologic Sensors: Automating Data Extraction for the iShake System, *ONS On Automation Science And Engineering*, vol. 10, n° 2, p. 242-251.

Shala U., Rodriguez A. (2011). *Indoor Positioning using Sensor-fusion in Android Devices*, Graduation work for a Degree of Master in Embedded Systems,School of Health and Society, Department Computer Science, Kristianstad University, Sweden, 2011.

Virieux J., Le Figaro, http://www.lefigaro.fr/sciences/2014/04/07/01008-20140407ARTFIG00398-le-sud-est-est-la-region-francaise-la-plus-exposee-au-risque-sismique.php.

Gouvernance des projets open source

Dr Ir Robert Viseur[1 2]

1. CETIC
Avenue Jean Mermoz, 28, B-6041 Charleroi (Belgique)
robert.viseur@cetic.be

2. UMONS Faculté Polytechnique
Rue de Houdain, 9, B-7000 Mons (Belgique)
robert.viseur@umons.ac.be

RÉSUMÉ. Les logiciels open source sont utilisés par la majorité des entreprises ; certaines n'hésitent pas à jouer un rôle moteur dans leur développement. Ces pratiques amènent des défis en matière de gouvernance des systèmes d'information et des projets open source. Le fork, comme scission d'une communauté, peut être la conséquence grave d'une mauvaise gouvernance. Après un état de l'art consacré aux concepts de gouvernance open source et de fork, nous proposons deux ensembles d'études de cas, le premier sur des grands projets analysés dans la littérature scientifique, le second, sur un ensemble de forks. Sur cette base, nous développons différents moyens permettant de limiter le risque de forks, identifions quatre logiques de gouvernance open source, discutons l'intérêt des stratégies d'inner source pour organiser une transition entre les stratégies propriétaires et open source, montrons les similitudes en termes de besoins en gouvernance entre projets d'open source innovation et insistons enfin sur l'impact du degré d'ouverture de la gouvernance sur la concurrence.

ABSTRACT. The open source software are used by the majority of companies; some of them do not hesitate to play a leading role in their development. These practices result in challenges related to the governance of information systems and open source projects. The fork, as division of a community, can be the serious consequence of poor governance. After a state of the art dedicated to the concepts of open source governance and fork, we propose two sets of case studies, the first ones about major projects discussed in the scientific literature, the second ones about a set of forks. On this basis, we develop various ways to limit the risk of forks, identify four logics of open source governance, discuss the interest of inner source strategies to organize a transition between proprietary and open source strategies, show similarities in terms of needs in governance between open source software and open source innovation projects, and further stress the impact of the governance on competition.

MOTS-CLÉS : open source, inner source, gouvernance, fork.

KEYWORDS: open source, innersource, governance, fork.

Introduction

Le logiciel libre est défini à partir de 4 libertés: la liberté d'exécuter le logiciel, de l'étudier, d'en redistribuer des copies et de le modifier (`gnu.org`). Créé en 1998, le terme « *open source* » est parfois préféré à celui de « *free software* » (logiciel libre). Il est défini par l'OSI (`opensource.org`) sur base d'une liste de 10 critères appelée Open Source Definition (OSD), incluant notamment l'accès au code source, la liberté de redistribution, l'autorisation de créer des oeuvres dérivées et la protection du nom des auteurs. Le terme « *open source* » est associé à un type de licence logicielle, à une approche du développement logiciel, à un type de communauté et à un type de modèle d'affaires (O'Mahony, 2007). Dans la suite de ce papier, les termes « logiciel libre » et « *open source* » seront considérés comme équivalents (l'acronyme FLOSS est parfois utilisé pour éviter les débats de terminologie). Les licences de logiciels libres et *open source* respectent les quatre libertés et dix critères, qui impliquent la mise à disposition du code source du logiciel (condition nécessaire mais non suffisante).

L'organisation des projets *open source* est à l'origine considérée comme organique, qualifiée de « bazar » par comparaison à la « cathédrale » des entreprises ou des organisations historiques (Raymond, 2001). Les communautés ont longtemps été vues comme auto-organisées. En pratique, la réalité est cependant plus contrastée, les structures de gouvernance se révèlant plus ou moins complexes, en fonction de la nature des projets, de la diversité des acteurs ou des interactions avec d'autres projets. L'intérêt du secteur privé s'est aussi accru avec le temps, comme le montrent la libération du code des logiciels Netscape et le lancement du projet Mozilla en 1998 (O'Mahony, 2007 ; Viseur, 2011, 2013a), la libération de Netbeans par Sun Microsystems en 2000 (Jensen et Scacchi, 2010), les investissements d'IBM dans Linux en 2001 (West, 2003) ou, plus récemment en France, le projet Capella porté par Thales. Le virage commercial de l'*open source* a été longuement commenté par Fitzgerald (2006).

Le *fork* est un mécanisme de fractionnement d'une communauté que l'on retrouve généralement dans le domaine du logiciel libre. Certains *forks* ont fait l'objet d'une médiatisation particulière (p.ex. LibreOffice.org / OpenOffice.org). En tant qu'échec d'une collaboration dans un contexte d'innovation ouverte impliquant une communauté d'utilisateurs, le *fork* constitue un sujet d'étude concret et qui peut être instructif quant aux limites d'un système de gouvernance (p.ex. manque de gouvernance ou divergence irréconciliable de stratégies).

Notre papier est organisé en 3 sections. La première section propose un état de l'art sur le concept de gouvernance *open source* et de *fork*. La seconde section présente deux ensembles d'études de cas. Le premier se base sur des études de grands projets analysés dans la littérature scientifique ; le second, sur un ensemble d'études de *forks* célèbres dans l'histoire de l'*open source*. La troisième section discute les résultats trouvés dans les études de cas.

1. État de l'art

1.1. Gouvernance open source

La gouvernance *open source* est rarement définie ; elle est souvent associée à différentes structures, règles, pratiques et normes incluant par exemple les licences ou les processus de communication (Markus, 2007). La diversité des projets *open source* en termes de tailles, de maturité, de licences ou de culture permet par ailleurs de supposer des mécanismes de gouvernance fort différents. La complexité du concept s'est par ailleurs encore accrue avec le développement de l'*open source* au delà du logiciel : *open content*, *open data*, *open hardware*,... Cette extension a été décrite et qualifiée d'*open source innovation* par Raasch (2009) puis par Pénin (2012), qui en a plus longuement défini le concept.

Markus (2007) définit la gouvernance *open source* comme l'ensemble des moyens mis en œuvre pour l'orientation, le contrôle et la coordination d'organisations et d'individus totalement ou partiellement autonomes pour le compte d'un projet de développement *open source* auquel ils contribuent collectivement.

La gouvernance d'un projet ou d'un ensemble de projets se structure généralement progressivement en suivant trois phases (de Laat, 2007). Dans une première phase du projet, la gouvernance est informelle et liée aux règles fixées par la licence (p.ex. GPL). Le *leadership* est exercé par des développeurs sur base de leurs performances (méritocratie). Dans une seconde phase, dès lors que la taille du projet augmente, un ensemble d'outils formels et explicites sont mis en place. Il s'agit de la modularisation du code source, de la division des rôles et de la délégation de la prise de décision, de la formation et la transmission des valeurs, de la formalisation des procédures et du régime d'exercice du pouvoir (autocratique ou démocratique). Dans une troisième phase, le projet, confronté au monde extérieur, doit assurer sa pérennité sur les plans matériel, financier et légal. Cela passe notamment par l'institutionnalisation des projets et la création de fondations (p.ex. Apache, Debian ou Mozilla).

La gouvernance d'un projet *open source* peut s'analyser sur plusieurs niveaux (Jensen et Scacchi, 2010). Le premier niveau d'analyse est *micro*. Il concerne les participants individuels au projet (p.ex. actions, ressources et interactions). Le second est *meso*. Il concerne les équipes de projets (p.ex. collaboration, *leadership*, contrôle et résolution de conflits). Le troisième est *macro*. Il concerne les écosystèmes inter-projets (p.ex. collaboration, autorité, contrôle et résolution de conflits).

Markus (2007) propose un ensemble d'éléments permettant de caractériser la gouvernance d'un projet *open source* : la propriété des actifs (p.ex. *copyrights*, licences ou marques), les objectifs du projet (p.ex. charte et vision), la gestion de la communauté, le processus de développement logiciel (p.ex. identification des besoins et affectation des tâches), la résolution de conflits et le changement de règles, et l'utilisation de l'information et des outils (p.ex. modalités d'accès aux outils et aux répertoires de code source).

Les licences *open source* constituent un des moyens de gouvernance. Ces licences sont généralement classées en deux grandes familles : les licences *copyleft* (dites aussi gauches d'auteur ou réciproques) et les licences permissives (dites aussi académiques ou non *copyleft*) (Alspaugh *et al.*, 2009 ; Fitzgerald, 2006 ; Lerner et Tirole, 2005 ; Montero *et al.*, 2005 ; Muselli, 2008). Une licence permissive (p.ex. BSD ou MIT) autorise l'utilisateur à placer le programme sous une nouvelle licence, libre ou même propriétaire (caractère appropriable). Une licence copyleft (p.ex. LGPL, GPL, AGPL ou MPL) « *lie l'octroi des droits à l'obligation de ne redistribuer le logiciel et ses modifications que sous la même licence que celle par laquelle le licencié a obtenu ces droits* » (Montero *et al.*, 2005). Elle confère dès lors au logiciel un caractère inappropriable. On parlera de copyleft faible, lorsque le *copyleft* s'applique uniquement à un composant logiciel, ou de *copyleft* fort, dans le cas où toute œuvre dérivée doit adopter la licence *copyleft* du composant logiciel (caractère contaminant).

Différents niveaux d'ouverture sont possibles au sein des communautés *open source*. Apache apparaît par exemple comme une communauté structurée avec une gouvernance transparente et ouverte. À l'opposé, le modèle *open core*, où seul un noyau générique et quelques modules sont *open source* (p.ex. Zenoss), le logiciel complet étant essentiellement propriétaire, laisse peu de place à la communauté (Viseur, 2013c). Aucune activité communautaire n'est d'ailleurs attendue et le pouvoir est concentré entre les mains de l'éditeur *open core*. L'Open Governance Index permet en pratique de quantifier le degré d'ouverture d'un projet en termes de transparence, de prise de décision, de réutilisation et de structure communautaire (Laffan, 2011, 2012). Cet indice comporte 13 métriques relatives à 4 domaines de gouvernance : l'accès au code source, le processus de développement, la création d'œuvres dérivées et la communauté.

Les pratiques *open source* ont connu une traduction en entreprise pour le développement de composants internes. Ce modèle est qualifié d'*inner source* (Stol *et al.*, 2011). Deux mises en œuvre sont distinguées : l'*inner source* basé infrastructure et l'*inner source* basé projet. Dans le premier cas, des porteurs de projets individuels sont invités à mettre à disposition leur projet sur une infrastructure de développement mutualisée inspirée par les plates-formes de développement *open source*. Dans le second cas, une équipe de développement interne prend en charge, dans un but de mutualisation, le développement et le support de composants critiques exploités dans différentes lignes de produits commercialisées par l'entreprise.

1.2. Fork

Bar et Fogel (2003) définissent le *fork* comme une situation se produisant « *lorsqu'un groupe de développeurs prend le code d'un projet de logiciel libre pour en démarrer un autre* ». Nyman et Mikkonen (2011), dans une étude de 566 projets hébergés sur Sourceforge.net et présentés par leurs administrateurs comme des *forks*, identifient des motivations à *forker* classables en quatre catégories : les motivations techniques (ajout de caractéristiques, spécialisation, portage, amélioration), les

changements de licence, les adaptations locales (langues ou particularités régionales) et la relance de projets abandonnés. Wheeler (2007) relativise la dangerosité présumée du *fork* et l'associe à un mécanisme de saine compétition. Il le compare au principe de la motion de censure dans un parlement ou à une grève. Le *fork* permettrait à la communauté des développeurs d'attirer l'attention des *leaders* sur des demandes non prises en compte. Nyman *et al.* (2011b) y voient même une « *main invisible* » garante du caractère durable et de la continuité des projets. La capacité à *forker* garderait par ailleurs « *les communautés vivantes, et les entreprises honnêtes* » (Moody, 2009). Elie (2006) voit dans le *fork* « *un droit essentiel* » mais insiste aussi sur le « *risque de se couper en même temps de la richesse du tronc commun* ». Il y voit souvent la conséquence de « *systèmes de régulation mal définis* ». Le mérite relèverait moins dans les communautés de logiciels libres de la compétence technique que du charisme et de la capacité à vivre dans le conflit. Notons que, sur le plan technique, le *fork* fait aussi partie des pratiques normales de certains outils de gestion de code source (p.ex. Git) ou de certains hébergeurs de projets (p.ex. Github).

2. Études de cas

2.1. Android, Apache, Mozilla, MySQL et Netbeans

Différents projets ont fait l'objet d'études sur la gouvernance ou sur les facteurs de succès (incluant des questions de gouvernance). Nous retiendrons les cas d'Android (Laffan, 2011, 2012), d'Apache HTTP Web Server (Viseur, 2016b), de Mozilla (Viseur, 2013a), de MySQL (Välimäki, 2003) et de Netbeans (Jensen et Scacchi, 2010). Ces derniers bénéficient d'études de cas déjà publiées dans la littérature scientifique. Ils couvrent une large diversité de projets et permettent une compréhension des mécanismes de gouvernance mis en œuvre, en termes notamment de prise de décision, de contribution ou de licence.

2.1.1. Android

Le système d'exploitation Android (`www.android.com`) a été lancé en novembre 2007. Il marquait l'entrée de Google sur le marché du mobile. Il est soutenu par l'Open Handset Alliance (`www.openhandsetalliance.com`), un consortium d'entreprises actives dans les technologies mobiles. La publication sous licence *open source* Apache a été réalisée en octobre 2008, parallèlement au lancement du téléphone HTC G1. La part de marché du nouvel OS a rapidement grandi par la suite, atteignant près de 80% du marché (source IDC, Q4 2014). Laffan (2012) souligne le caractère fermé du projet Android, avec un Open Governance Index de 23%, à comparer à Symbian (58%) et Meego (61%). Laffan met en particulier en évidence les processus de décision unilatéraux, le processus fermé d'accès au code source (« *code committer* »), le processus fermé de contribution au code source, l'opacité du processus de contrôle et de prise de décision autour de l'Android Compliance Program ou encore l'absence de volonté d'évoluer vers un

modèle de gouvernance plus ouvert. Des variantes de l'Android officiel existent (p.ex. CyanogenMod).

2.1.2. Apache HTTP

Le projet de serveur Web Apache HTTP (`httpd.apache.org`) a été démarré en 1995 sur base du code source du serveur du NSCA par un groupe de développeurs géographiquement distribués, connus sous le nom d'Apache Group. Il est publié sous licence Apache. En 1999, les membres de l'Apache Group ont fondé l'Apache Software Foundation, dont l'objectif était de fournir un support financier, légal et organisationnel au serveur Apache HTTP. La Fondation a par la suite évolué vers une structure multi-projets et a acquis le sponsoring d'entreprises IT de taille mondiale comme Hewlett-Packard, IBM et Microsoft. Elle se distingue notamment par sa structure d'incubation, permettant l'entrée de logiciels novateurs dans le portefeuille de projets et leur accès aux outils de gestion des projets (p.ex. gestionnaires de sources et gestionnaires de bugs). L'incubateur différencie nettement les logiciels matures, de haute qualité, et les nouveaux projets devant encore faire leurs preuves. Il permet ainsi un renouvellement dans la production logicielle (innovation) sans nuire à la réputation des logiciels de référence.

2.1.3. Mozilla

La Fondation Mozilla (`www.mozilla.org`) a été créée en juillet 2003 sur les cendres de la société Netscape, rachetée en mars 1999 par AOL Time Warner. Elle est à l'origine du logiciel Mozilla, issu de la libération de technologies de la société Netscape en 1998. Elle prend aujourd'hui en charge le développement du navigateur web Firefox ainsi que d'autres projets d'outils de développement ou de logiciels pour les utilisateurs finaux. Un important travail de réécriture et de modularisation du code source fourni initialement par Netscape ainsi qu'un travail sur l'ergonomie du logiciel ont permis à Firefox d'atteindre une part de marché autour des 25%. La communauté s'appuie sur une organisation assez hiérarchisée, incluant par exemple l'existence de responsables de modules (« *module owners* »), à l'origine de fréquents conflits. La licence MPL, écrite pour le projet afin de faciliter l'agglutination de codes sources sous des licences différentes et de protéger le travail des contributeurs, a changé à plusieurs reprises, en raison de réactions communautaires ou de dépendances à d'autres projets, selon un mode souvent collaboratif.

2.1.4. MySQL

MySQL (`www.mysql.com`) est une base de données développée depuis 1995. Le logiciel a été placé sous licence GPL en 2000, tandis que la société MySQL a été créée en 2001 pour valoriser la technologie. L'entreprise applique un modèle de double licence : une version *open source* cohabite avec une version commerciale (i.e. payante). L'entreprise tire dès lors ses revenus de la prestation de services mais également du paiement de licences (plus de 50 % de son chiffre d'affaires en 2003) par les clients achetant la version propriétaire. L'existence des deux branches du logiciel est garantie par la totale propriété du code source. Toutes les contributions sont à cette fin vérifiées et réécrites par l'entreprise. MySQL AB a dû faire face à un

fork suite au rachat de MySQL AB par Sun Microsystems puis de ce dernier par Oracle Corporation. Ce *fork* s'appelle MariaDB (`mariadb.org`) et a été initié par Michael Widenius, fondateur de MySQL ; il bénéficie du support de plusieurs entreprises importantes, dont Google, et est soutenu par un consortium baptisé Open Database Alliance (`www.opendatabasealliance.com`).

2.1.5. Netbeans

Netbeans (`netbeans.org`) est un projet d'environnement de développement pour le langage Java, mis en *open source* sous licence CDDL par Sun Microsystems en 2001. Le projet offre une large autonomie aux développeurs. Par contre, Sun Microsystems a conservé pendant longtemps une empreinte forte sur les structures décisionnelles chapeautant l'organisation. Un projet de fusion avec son principal concurrent *open source*, Eclipse, sponsorisé par IBM, a échoué pour cause de différences techniques et organisationnelles entre les deux sponsors ainsi que de risque de perte d'image (*leadership* technologique). L'ascendant a finalement été pris par Eclipse.

2.1.6. Synthèse

Les projets dont les caractéristiques sont synthétisées dans cette section présentent une grande diversité en termes de licences (licences Apache, licence GPL, licence MPL, licence CDDL, double licence,...), de modèles économiques (diverses formes d'édition logicielle et/ou de mutualisation des développements), d'organisation et d'évolution du rapport à la communauté (contrôle par un éditeur, autonomisation progressive d'une structure de mutualisation ouverte à la communauté,...). Les utilisateurs et contributeurs semblent rencontrer davantage de difficultés à faire accepter leurs codes sources ou à intégrer les organes de décision sur base du mérite lorsqu'un projet est contrôlé par une entreprise, ce qui peut se traduire par des *forks* (Android, MySQL).

2.2. Forks

2.2.1. Méthodologie

Plusieurs *forks* ont fait l'objet d'études plus ou moins approfondies dans la littérature. Citons la famille des systèmes d'exploitation BSD (Weber, 2004), Roxen (Dahlander et Magnusson, 2008), GCC (Fogel, 2004), CVS (Bart et Fogel, 2003), Spip (Elie, 2006) ou LibreOffice.org (Gamalielsson et Lundell, 2012). Ces résultats seront exploités. Nous avons procédé à l'étude de 26 *forks* de projets populaires ; ces projets sont 386BSD, Claroline, Compiere, CVS, Dokeos, Ext JS, FreeBSD, GCC, GNU Emacs, KHTML, Mambo, MySQL, NCSA HTTPd, NetBSD, OpenERP, OpenOffice.org, PhpGroupware, PhpNuke, Qt, RHEL, Roxen, Samba, Sodipodi, Sourceforge, Spip et Xfree86. Nous avons exploité des documents existants : livres, articles scientifiques, articles de presse, actualités postées sur des portails informatiques ou *open source*, communiqués et pages de sites de projets,... Chaque *fork* a fait l'objet d'une fiche descriptive, reprenant la chronologie du *fork*, les auteurs

impliqués et leurs motivations. Les résultats ont été synthétisés au sein d'un tableau récapitulatif, reprenant le nom du projet initial, le nom du *fork*, la (ou les) motivations(s) du *fork* et l'impact du *fork* sur le projet original (rapport technique interne). L'impact a été évalué au sens des issues possibles identifiées par Wheeler (2007).

2.2.2. Motivations à déclencher un fork

Sept motivations à *forker* ont été identifiées : l'arrêt du projet original (19%), les motivations techniques - nouvelle spécialisation, vues techniques divergentes ou objectifs techniques différents - (42%), le changement de licence (15%), le conflit autour de la propriété d'une marque (12%), les problèmes de gouvernance du projet (38%), les différences culturelles fortes (8%) et la recherche de nouvelles pistes d'innovation (4%).

En pratique, les études des *forks* qui précèdent montrent que les *forks* qui réussissent (et sont donc susceptibles de porter préjudice à l'éditeur original, lorsqu'il y en a un) démarrent généralement pour une raison importante. L'arrêt du support d'un produit libre populaire entraîne souvent un *fork* (cf. NCSA HTTPd, 386BSD, Red Hat Linux et Roxen). Le *fork* libre réussit généralement mais peut cohabiter avec une version fermée du produit (cf. Red Hat Linux, Roxen, Sourceforge). Un *fork* peut intervenir suite à l'apparition de divergences techniques. Les systèmes *BSD ont ainsi souvent adopté des spécialisations techniques distinctes (portabilité, sécurité,...). C'est la cause la plus fréquente (42%). La gouvernance du projet apparaît à l'origine du conflit pour environ un tiers des cas étudiés (38%). Le problème porte généralement sur le manque d'ouverture des équipes de développement : prise en compte des contributions extérieures (cf. OpenOffice.org), discussion des objectifs du projet (cf. Sodipodi), réticences face à un mode de développement communautaire (cf. OpenOffice.org, Dokeos, PHP Nuke),... La propriété de la marque apparaît également comme une source de conflit (cf. Claroline, Mambo et OpenOffice.org). Elle peut être liée à la question de la gouvernance (cf. propriété des actifs) car la marque permet en pratique à l'éditeur de conserver un contrôle sur l'évolution du projet. La marque cristallise dès lors les tensions entre un sponsor (p.ex. éditeur *open source*) et sa communauté. Le problème de licence apparaît parfois à l'origine d'un *fork*, qu'il n'affecte pas le type de licence (cf. Xfree86) ou qu'il entraîne au contraire une augmentation (cf. Ext JS) ou une réduction (p.ex. arrêt de la branche libre) de la liberté du logiciel. La mise sous GPL ou AGPL facilite par ailleurs les échanges entre projets, dès lors que la licence d'origine peut difficilement être changée. Le changement de licence n'est cependant pas un motif dominant à *forker* (15%). Les *forks* qui ont été suscités par Theo de Raadt, leader d'OpenBSD, peuvent être justifiés, au moins en partie, par des prises de position politiques ou idéologiques. Cette configuration paraît finalement assez marginale dans le paysage *open source*. Les chocs culturels -communauté vs entreprise (cf. KHTML), communauté vs administration (cf. Spip)- apparaissent par contre comme une cause possible (8%), illustrant la difficulté à aligner des stratégies communautaires et commerciales.

2.2.3. Conséquences des forks

Les études montrent que les *forks* qui se traduisent par une disparition totale du projet d'origine ne sont pas majoritaires (19%). À l'exception du serveur Apache, de X.Org, de Joomla ou d'Inkscape, une cohabitation apparaît dans plus de la moitié des cas étudiés (54%). Dans certains cas, les échanges de codes continuent d'ailleurs (cf. FreeBSD, NetBSD, OpenBSD). Une fusion ultérieure des projets (cf. GCC et EGCC) est possible. La divergence progressive peut par contre rendre la fusion malaisée (cf. Webkit et KHTML). L'échec total du *fork* intervient dans moins d'un cas sur cinq (19%).

3. Discussion

Nous identifions ci-dessous quatre logiques de gouvernance *open source* ainsi que différents moyens permettant de gérer le risque de *forks*, discutons ensuite l'intérêt des stratégies d'*inner source* pour organiser une transition entre les stratégies propriétaires et *open source*, montrons les similitudes en termes de besoins en gouvernance entre projets d'*open source innovation* et insistons enfin sur l'impact du degré d'ouverture de la gouvernance sur la concurrence.

3.1. Logiques de gouvernance

Plusieurs logiques de gouvernance émergent des différents cas traités dans cette recherche.

3.1.1. Logique individuelle

Par exemple : majorité des petits projets hébergés sur GitHub (`github.com`). Elle s'applique à la majorité des projets *open source* publiés, maintenus par une personne sur des dépôts publics. L'autorité y est exercée de manière informelle par l'auteur du logiciel. Les procédures de travail ne sont pas formalisées. La communauté est souvent petite et les contributions sont rares, réduisant la vitalité du projet mais aussi les sources de conflit.

3.1.2. Logique commerciale

Par exemple : MySQL ou Zenoss. Elle s'applique aux projets d'édition *open source* menés par des entreprises privées. Il s'agit généralement de projets valorisés selon le modèle de la double licence, associant une version communautaire du logiciel sous licence réciproque et une version commerciale payante techniquement différenciée. L'entreprise peut éventuellement chercher le soutien de la communauté (dans le cas du modèle *open core* la communauté est inexistante) mais veille à garder le contrôle sur le projet, notamment par la signature d'accords de contributeurs organisant le partage des droits patrimoniaux (Poo-Caamaño et German, 2015 ; Valimaki, 2003).

3.1.3. Logique communautaire

Par exemple : Apache HTTP ou Chamilo. Elle s'applique aux projets de grande taille, dont le succès a nécessité la mise en place d'une structure de soutien (p.ex. fondation ou association), permettant d'en assurer la pérennité, et la mise en place de procédures organisationnelles. La prise de décision est ouverte aux membres de la communauté sans volonté d'entraver la prise de responsabilité liée au mérite.

3.1.4. Logique industrielle

Par exemple : Eclipse ou Netbeans. Elle s'applique aux projets développés par des acteurs industriels ayant pour objectif une mise en commun de l'effort de développement (coopétition). La gouvernance peut paraître similaire à celle des grands projets à logique communautaire structurés en fondation. Elle peut cependant différer à la marge, par exemple par l'imposition de droits d'inscription limitant l'accès aux fonctions dirigeantes et permettant aux grandes entreprises de conserver un contrôle sur les objectifs du projet.

3.2. Gouvernance et inner source

Les logiques de gouvernance identifiées *supra* peuvent être rapprochées des différents modes d'*inner source*. L'*inner source* basé infrastructure peut ainsi être comparé à la logique individuelle (porteurs de projets individuels profitant d'une infrastructure de développement mutualisée) ; l'*inner source* basé projet, à la logique industrielle (objectif de mutualisation d'actifs critiques).

Un des défis auquel les entreprises font face réside dans la mise en *open source* des projets internes. Dès lors que les projets sont anciens, leur libération influence les modèles d'affaires et les stratégies mises en œuvre par l'entreprise (West, 2003). Sur le plan patrimonial, l'organisation d'une transition depuis une stratégie propriétaire vers une stratégie *open source* par le recours à des licences hybrides a déjà été appliquée par Sun Microsystems (Community Source et licence SCSL). Ce cas a fait l'objet d'une étude par Muselli (2007). L'application des principes de l'*inner source*, basé infrastructure ou basé projet, à l'intérieur de l'entreprise, pourrait également servir de transition vers une externalisation complète en *open source*. *Inner source* et *open source* partagent en effet un ensemble de préoccupations communes quant à la motivation des contributeurs ou de support aux utilisateurs du composant logiciel. Ce mode d'évolution pourrait cependant entraîner des habitudes de contrôle sur le processus de développement susceptibles d'ultérieurement provoquer des conflits avec la communauté *open source*.

3.3. Causes de forks liés à la gouvernance

Cette section compare les résultats des études de cas avec l'étude précédente de Nyman et Mikkonen (2011), discute ensuite la réduction du risque de *fork* et examine enfin du lien entre les *forks* et la logique de gouvernance.

3.3.1. Étude de Nyman et Mikkonen (2011)

Comparé à l'étude de Nyman et Mikkonen (2011), notre recherche regroupe plusieurs motivations sous le label de « motivation technique » et met en évidence trois causes supplémentaires : les questions de gouvernance, les difficultés liées aux différences culturelles et les conflits liés à la propriété d'une marque. Ceci étant, les changements d'orientation technique occupent aussi une place prépondérante dans notre étude. La reprise de projets arrêtés apparaît plus importante. Ces différences peuvent s'expliquer par le spectre plus large des causes prises en compte mais aussi par la nature différente des projets : Nyman et Mikkonen (2011) se basent sur un ensemble des projets pris sur Sourceforge.net, qui héberge de nombreux projets de taille réduite, alors que avons basé notre étude sur des projets populaires ayant déjà une communauté active, jouant un rôle de régulation et de responsabilisation.

3.3.2. Architecture de participation

Le risque de *fork* pour divergences techniques est élevé. Il peut cependant être limité en adoptant dès le départ une architecture modulaire. MacCormack *et al.* (2006) parlent d'architecture de participation, et y voient un moyen pour simplifier la compréhension du code et les contributions. Le modèle noyau-extension en est un exemple, permettant l'adaptation du logiciel sans en toucher le cœur. L'éditeur garantit alors les performances d'un noyau incorporant des fonctionnalités générales, tandis qu'intégrateurs et utilisateurs avancés en étendent les fonctionnalités par le développement d'extensions. Cette approche peut également limiter les conflits liés à l'organisation de l'équipe de développement puisque les intégrateurs ne doivent comprendre que les interfaces du logiciel permettant le développement d'extensions. La compréhension des spécificités du noyau n'est pas indispensable. Des conflits peuvent par contre apparaître dès lors que des extensions communautaires entrent en conflit avec des extensions propriétaires vendues par l'éditeur. Le modèle noyau-extension, avec la création d'interfaces de programmation d'applications (API) qu'elle sous-tend, rappelle les « *user toolkits for innovation* » décrites par Von Hippel (2001). Ces boîtes à outils permettent, sur des marchés composés de clients aux besoins hétérogènes, une forme d'externalisation vers les clients des tâches d'innovation nécessitant une connaissance pointue des besoins des clients. Le bénéfice attendu est une meilleure satisfaction des besoins et, dans un projet logiciel libre, un moindre risque de tensions autour des orientations du projet. Franke et Von Hippel (2003) considèrent par défaut les logiciels *open source* comme un « *user toolkit* » du fait de l'accès au code source ; cependant, l'existence d'Interfaces de Programmation d'Applications (APIs) documentées abaisse les barrières à la participation sur le projet *open source*.

3.3.3. Structures d'expérimentation

Samba illustre le coté « tueur d'innovation » de l'obligation de qualité liée à une importante base d'utilisateurs. Cet exemple souligne l'intérêt des incubateurs, tels que celui d'Apache, permettant l'expérimentation en marge du projet principal. D'une certaine manière, le *fork* Samba TNG joue d'ailleurs un rôle d'incubateur. Un

effet similaire peut être obtenu par la création de branches expérimentales au sein du dépôt de sources (cf. Linux par exemple).

3.3.4. Différences de culture

Les différences de culture entre membres d'une même communauté pourraient également constituer un motif de *fork*, dont la prévention mériterait des recherches plus approfondies (l'effet de ce type de *fork* n'est cependant pas obligatoirement négatif, s'il permet aux communautés de sortir des conflits et de recentrer les efforts sur le développement des logiciels). L'exemple de Claroline est parlant de ce point de vue. Dominé par des institutions d'enseignement, ce projet a finalement *forké* à deux reprises, la première avec Dokeos (entreprise), la seconde avec Chamilo (communauté / association).

3.1.5. Forks et logiques de gouvernance

La logique individuelle est généralement portée par une seule personne ; cette dernière joue un rôle de « dictateur bienveillant » et peut conduire à un *fork* du fait de divergences avec la communauté (p.ex. Sodipodi). La logique commerciale peut conduire à une opposition entre les objectifs de la communauté et celle de l'entreprise qui porte le projet, conduisant à la rupture (p.ex. MySQL). La logique industrielle peut se traduire par la mainmise de grands groupes sur le projet et conduire à une fronde des communautés dont les contributions sont handicapées (p.ex. OpenOffice.org). La logique communautaire est naturellement tournée vers les contributeurs extérieurs ; elle n'élimine cependant pas les risques de *forks*, notamment amicaux (p.ex. Samba ou la famille BSD). En pratique, aucune logique de gouvernance ne semble à même d'éliminer le risque de *fork* pour l'organisation qui porte le projet. De plus, les projets adoptent souvent des structures hybrides, qui rendent difficile l'identification des causes et de leurs effets. C'est par exemple le cas pour des éditeurs *open source* encourageant par ailleurs la création d'une communauté active de développeurs (p.ex. OpenERP/Odoo). Dans ce cas, la structuration progressive de la logique commerciale, suite par exemple à l'arrivée d'investisseurs, peut favoriser les tensions et conduire au *fork* (Viseur, 2013c).

La perception négative du *fork* en tant que risque mérite par ailleurs réflexion. Cette dernière provient essentiellement de *forks* inamicaux, fruits de tensions durables, conduisant à une scission de la communauté et à une perte en termes de mutualisation. Cette situation peut cependant constituer un processus salvateur d'auto-régulation en cas de mauvaise gestion du projet ou de divergence irréconciliable. Par ailleurs, certains *forks* apparaissent amicaux, conduisant à des choix techniques différents, parfois avec des échanges de code source (p.ex. famille BSD), ou à des expérimentations en marge du projet officiel (p.ex. Samba TNG).

3.4. Gouvernance au delà du logiciel

L'extension des pratiques *open source* au matériel conduit à des similitudes avec le domaine logiciel, tant pour les licences que pour les modèles d'affaires ou les

modèles de développement (Viseur, 2012). Dans le domaine des véhicules *open source*, par exemple, les licences réciproques (CC-BY-SA) et le modèle de la double licence sont expérimentés. Les logiques individuelles (p.ex. Wikispeed) ou commerciales (p.ex. OSVehicle) y co-existent, tandis que la faible maturité de ces projets semble ne pas encore avoir conduit à la création d'organisations structurant l'activité (Viseur, 2016a).

La problématique du *fork*, bien que moins fréquente, est également connue, comme le montre le conflit récent autour du projet Arduino (Orsini, 2015). En pratique, les cartes de prototypage Arduino sont depuis longtemps concurrencées par des cartes équivalentes : Freeduino, Sanguino, Seeduino,... Depuis 2015, cependant, un conflit entre fondateurs du projet autour du paiement de *royalties* et de l'exploitation de la marque Arduino a conduit à la cohabitation de deux entreprises concurrentes : Arduino SRL (ex-SmartProjects) et Arduino LLC. Cette dernière exploite la marque Genuino (hors USA).

3.6. Gouvernance et concurrence

En 1998 déjà, Eric Raymond soulignait la tendance de certains projets *open source* à occuper la totalité de la niche fonctionnelle qu'ils occupaient : « *Some very successful projects become `category killers'; nobody wants to homestead anywhere near them because competing against the established base for the attention of hackers would be too hard. People who might otherwise found their own distinct efforts end up, instead, adding extensions for these big, successful projects.* ». Or, les composants *open source* sont aujourd'hui massivement utilisés dans les développements logiciels. Ainsi, une étude de Black Duck Software (« *2015 Future of Open Source Survey Results* ») révélait que 78% des personnes interrogées basent leurs activités commerciales sur des logiciels *open source* et que les deux-tiers construisent des logiciels pour leurs clients qui sont eux-mêmes basés sur du logiciel *open source*. Le poids de certains projets tend à inquiéter les autorités de régulation de la concurrence, comme en Europe avec Android (82,8% de part de marché mondiale au Q2 2015 selon IDC), caractérisé par une gouvernance opaque : « *The Commission's in-depth investigation will focus on whether Google has breached EU antitrust rules by hindering the development and market access of rival mobile operating systems, applications and services to the detriment of consumers and developers of innovative services and products* » (EU, 2015).

Une gouvernance *open source* plus fermée représente donc potentiellement une menace pour la concurrence. De grandes entreprises adoptent maintenant le modèle *open source* (p.ex. Google avec Android ou Microsoft avec .Net). Ces entreprises disposent d'un pouvoir économique certain et l'*open source* peut les aider à disséminer davantage leurs technologies et standards. Adatto (2013) montre en effet comment la mise à disposition d'une implémentation de référence peut faciliter la diffusion d'une innovation. La modulation de l'ouverture au cours du temps peut dès lors conduire à des situations de pouvoir économique au travers de technologies et standards largement disséminés. Dans ce contexte, le *fork* apparaît comme un mécanisme d'auto-régulation susceptible de rétablir de la concurrence dans un

marché où elle est menacée par un grand groupe industriel. La disponibilité du code source d'Android permet ainsi l'émergence d'intégrateurs alternatifs (p.ex. Cyanogen) et d'éditions spécifiques à un constructeur (p.ex. Xiaomi). Les entreprises peuvent en effet récupérer et *forker* le code source, pour ensuite l'adapter en toute autonomie aux besoins de leurs propres clients. Les versions *forkées* d'Android représenteraient ainsi environ 20% de l'écosystème Android.

4. Conclusion

Cette recherche aura permis différents apports aux questions de gouvernance *open source*. Premièrement, la question de la transition d'une stratégie propriétaire classique vers une stratégie *open source* préoccupe depuis longtemps les entreprises. La question a déjà été traitée par Muselli pour les aspects juridiques (2007) ; cette recherche propose une exploitation de l'*inner source* comme stratégie progressive de transition vers l'*open source*, permettant à l'entreprise d'acquérir les outils et les expertises nécessaires à la conduite d'un projet *open source*. Deuxièmement, nous avons identifié plusieurs logiques de gouvernance, permettant à l'entreprise de rapidement identifier le degré de structuration et le type de gouvernance nécessaires en fonction de la maturité d'un projet *open source*. Troisièmement, nous avons identifié un ensemble de causes pour les *forks* et avons proposé différents moyens pour en réduire le risque de survenance (p.ex. mise en place d'une architecture modulaire ou de structures d'expérimentation). Quatrièmement, nous avons montré que les problématiques de gouvernance et les risques de *forks* se retrouvaient également au sein de projets d'*open source innovation* (p.ex. *open hardware*). Il est ainsi possible, pour les projets d'*open hardware*, d'exploiter l'expertise accumulée dans le passé avec les projets *open source*. Cinquièmement, nous avons montré que la gouvernance, loin de n'agir que sur les utilisateurs et les développeurs du projet, jouait également un rôle sur la concurrence dans le secteur ICT.

Cette recherche a permis d'identifier trois sujets méritant un approfondissement. Premièrement, l'adoption d'une architecture modulaire, favorisant la participation au projet et la réutilisation au sein d'autres projets, limite les possibilités d'innovation architecturale. L'organisation des changements d'interface (protocole, spécifications,...) passe par la gouvernance inter-projet. Les modalités pratiques de cette dernière mériteraient cependant d'être approfondies. Deuxièmement, l'innovation au sein des projets apparaît comme un problème fondamental, tant pour éviter le risque de *fork* que pour tirer pleinement parti de la mutualisation des développements et de la diversité des contributeurs. Cela peut passer par des mécanismes souples de gestion de sources et le développement de pratiques managériales encourageant une forme d'intraprenariat (comportement d'entreprenariat interne à l'entreprise) sur les projets ou écosystèmes auxquels les entreprises contribuent (Viseur et Pinchart, 2013b). L'encouragement de la prise d'initiative en matière de contribution ou de création de nouveaux projets s'accompagne d'enjeux pour l'entreprise, par exemple en termes de politiques internes de propriété intellectuelle. Troisièmement, le caractère *open source* d'un projet ne doit pas masquer son impact potentiellement problématique sur l'état de la

concurrence dans le secteur ICT. L'utilisation d'un indice de concentration pondéré par le degré d'ouverture de la gouvernance peut aider à identifier les projets majeurs potentiellement problématiques en termes de concurrence (Viseur, 2016c).

5. Références

Adatto, T. (2013), Standards ouverts et implémentations FLOSS (Free Libre Open Source Software): vers un nouveau modèle synergique de standardisation promu par l'industrie du logiciel, in Terminal : Technologie de l'Information, Culture, Société, n°113-114, pp. 137-170, 2013.

Alspaugh, T.A., Asuncion, H.U., Scacchi, W. (2009), Intellectual property rights requirements for heterogeneously-licensed systems, 17th IEEE International Requirements Engineering Conference (RE'09), pp. 24–33, Augustus 31 - September 4, 2009.

Bar, M., Fogel, K. (2003), Open Source Development with CVS, Paraglyph Press.

Dahlander, L., Magnusson, M., How do firms make use of open source communities?, Long Range Planning, vol. 41, n°6, December 2008, pp. 629-649.

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

Elie, F. (2006), Économie du logiciel libre, Eyrolles.

EU (2015), Antitrust: Commission sends Statement of Objections to Google on comparison shopping service; opens separate formal investigation on Android, 15 April 2015. Online: http://europa.eu/rapid/pressrelease_IP154780_en.htm (lu : 17 janvier 2016).

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

Fogel, K. (2004), How To Run A Successful Free Software Project - Producing Open Source Software, CreateSpace.

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

Gamalielsson, J., Lundell, B. (2012), Long-term sustainability of Open Source software communities beyond a fork: a case study of LibreOffice. In Open Source Systems: Long-Term Sustainability (pp. 29-47). Springer Berlin Heidelberg.

Jensen, C., Scacchi, W. (2010), Governance in open source software development projects: A comparative multi-level analysis. In Open Source Software: New Horizons (pp. 130-142). Springer Berlin Heidelberg.

Laffan, L. (2011), Open governance index-Measuring the true openness of open source projects from Android to WebKit, VisionMobile, London. Online : https://upload.wikimedia.org/wikipedia/commons/5/5f/VisionMobile_Open_Governance_Index_report.pdf (lu le 16 janvier 2016).

Laffan, L. (2012), A new way of measuring openness: The open governance index. Technology Innovation Management Review, 2(1). Online : http://timreview.ca/article/512 (lu : 16 janvier 2016).

Lerner, J., Tirole, J. (2005), The Scope of Open Source Licensing, Journal of Law, Economics, and Organization, vol. 21, issue 1, 2005, pp. 20-56.

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, vol. 52 (7), 2006, pp. 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.

Montero E., Cool Y., de Patoul F., De Roy D., Haouideg H., Laurent P., (2005), Les logiciels libres face au droit, Cahier du CRID, n°25, Bruylant, 2005.

Moody, G. (2009), Who Owns Commercial Open Source – and Can Forks Work?, Linux Journal, April 2, 2009.

Muselli, L. (2007), Les licences informatiques: un outil de modulation du régime d'appropriabilité dans les stratégies d'ouverture. Une interprétation de la licence SCSL de Sun Microsystems. In 12ème conférence de l'AIM, Lausanne (Suisse).

Muselli, L. (2008), Le rôle de licences dans les modèles économiques des éditeurs de logiciels open source, Revue française de gestion, n°181, pp. 199-214.

Nyman, L. Mikkonen, T., Lindman, J., Fougère, M. (2011), Forking: the Invisible Hand of Sustainability in Open Source Software", Proceedings of SOS 2011: Towards Sustainable Open Source.

Nyman, L., Mikkonen, T. (2011), To Fork or Not to Fork: Fork Motivations in SourceForge Projects, IFIP Advances in Information and Communication Technology, Vol. 365, pp. 259-268.

O'Mahony, S. (2007), The governance of open source initiatives: what does it mean to be community managed?. Journal of Management & Governance, 11(2), 139-150.

Orsini, L. (2015), Arduino vs. Arduino: What We Know About The Open-Source Hardware Fork, ReadWrite.com. Online : http://readwrite.com/2015/03/18/arduino-open-source-schism (lu : 07 février 2016).

Pénin J. (2012), Open source innovation: Towards a generalization of the open source model beyond software. Revue d'économie industrielle, (4), 65-88.

Poo-Caamaño, G., German, D. M. (2015), The Right to a Contribution: An Exploratory Survey on How Organizations Address It. In Open Source Systems: Adoption and Impact (pp. 157-167). Springer International Publishing.

Raasch, C., Herstatt, C., Balka, K. (2009), On the open design of tangible goods. R&D Management, 39(4), 382-393.

Raymond, E. S. (1998), Homesteading the noosphere. First Monday, 3(10).

Raymond, E.S. (2001), The Cathedral & the Bazaar (Musings on Linux and Open Source by an Accidental Revolutionary), O'Reilly Media.

Stol, K. J., Babar, M. A., Avgeriou, P., Fitzgerald, B. (2011), A comparative study of challenges in integrating Open Source Software and Inner Source Software. Information and Software Technology, 53(12), pp. 1319-1336.

Välimäki, M. (2003), Dual licensing in open source software industry, Systèmes d'Information et Management, vol. 8, n°1, 2003, pp. 63-75.

Viseur, R. (2011), Associer commerce et logiciel libre : étude du couple Netscape / Mozilla, 16ème conférence de l'AIM, Saint-Denis (France), 25 mai 2011.

Viseur, R. (2012), From Open Source Software to Open Source Hardware. In Open Source Systems: Long-Term Sustainability (pp. 286-291). Springer Berlin Heidelberg.

Viseur, R. (2013a). Identifying success factors for the mozilla project. In Open Source Software: Quality Verification (pp. 45-60). Springer Berlin Heidelberg.

Viseur, R., Pinchart, L. (2013b), Developing Free Software within a Major ICT Company, First International Workshop on Community Experience in Open Software Development (CommEx 2013, June 28th 2013, Koper-Capodistria (Slovenia).

Viseur, R. (2013c), Évolution 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. (2016a), Open Source Hardware on the Cutting Edge: the Case of Open Source Cars, 2nd International Workshop on the Sharing Economy, 28-29 janvier 2016, Paris.

Viseur, R. (2016b), Etude des facteurs de succès du projet open source Apache Http, Actes du 21ème colloque AIM 2016, Lille, 18-20 mai 2016.

Viseur, R. (2016c), Open Concentration Index: Measure of Market Concentration in Open Source Industry, University of Mons, Working paper.

Von Hippel, E. (2001), User toolkits for innovation, Journal of Product Innovation Management, vol. 18 (4), July 2001, pp. 247-257.

Weber, S. (2004), The success of open source, Harvard University Press, April 30, 2004.

West, J. (2003), How open is open enough?: Melding proprietary and open source platform strategies. Research policy, 32(7), pp. 1259-1285.

Wheeler, D.A. (2007), Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers !, www.dwheeler.com.

Session commune avec RCIS : Data and Knowledge Management

Designing Multidimensional Cubes from Warehoused Data and Linked Open Data

Conception de Cubes Multidimensionnels à partir de Données Entreposées et de Données Ouvertes Liées

Franck Ravat[1], Jiefu Song[1], Olivier Teste[2]

1. IRIT - Université de Toulouse, Université Toulouse I Capitole
2 Rue du Doyen Gabriel Marty F-31042 Toulouse Cedex 09
{ravat|song}@irit.fr

2. IRIT - Université de Toulouse, Université Toulouse II Jean Jaurès
1 Place Georges Brassens F-31703 Blagnac Cedex
teste@irit.fr

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec *INFORSID* 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

ABSTRACT. A Data Warehouse (DW) is widely used as a consistent and integrated data repository in Business Intelligence systems. Warehoused data provide only a partial view over the activities of an organization. In today's highly dynamic business context, Linked Open Data (LOD) should also be included in analyses to offer multiple perspectives to decision-makers. However, differences between DW and LOD models make it hard to analyze all useful data in a unified way, which complicates the analysis tasks. Moreover, the dispersion of warehoused data and LOD in different schemas leads to repetitive searches for relevant information among various data sources, which reduces the efficiency of analysis.

In this paper, we provide a new conceptual multidimensional model, named Unified Cube, which offers as much useful information as possible for analyses. The definition of Unified Cube is generic enough to represent both warehoused data and LOD according to an analysis subject and a set of analysis axes. Besides the concepts in multidimensional models related to one source, a Unified Cube supports some advanced modeling features depicting inter-source equivalent analysis granularities and aggregation paths spanning different sources.

We describe a two-stage process to build a Unified Cube from DWs and multidimensional LOD. As a first step, schemas published with specific modeling languages are transformed into a common conceptual representation named exportation cube. The second step consists of associating together related data scattered in different exportation cube. An algebraic linking operator is proposed to enable non-expert users to associate relevant data together according to their analytical needs. We propose an algorithm to automate the execution of the linking operator while guaranteeing the overall validity of a Unified Cube.

To demonstrate the feasibility of the proposed concepts, we develop a prototype implementing a Unified Cube built from a R-OLAP DW and two LOD datasets. We illustrate the advantages of our proposal by showing how analyses of both warehoused data and LOD are carried out through the implemented Unified Cube.

RESUME. De nos jours, les systèmes d'aide à la décision intègrent couramment un entrepôt de données (ED) intégrant le plus souvent les données internes d'une l'organisation. Les données entreposées ne contiennent pas toutes les informations nécessaires aux prises de décision. De ce fait, des données externes comme les Données Ouvertes Liées (DOL) pourraient être ajoutées afin d'offrir aux décideurs de nouvelles perspectives d'analyse. Néanmoins, analyser de manière unifié les données entreposées et les DOL constitue une tâche complexe pour les non-spécialistes. Cette difficulté est due à (i) une modélisation différente des données d'ED et de DOL, (ii) une dispersion sur plusieurs sources des données relatives à un même sujet d'analyse. Pour effectuer une analyse décisionnelle, un décideur est donc obligé d'effectuer une recherche sur différentes sources de données.

Face à ces problématiques, nous proposons, dans cet article, une nouvelle modélisation conceptuelle, appelé Cube Unifié, qui permet de regrouper toutes les informations utiles pour des analyses décisionnelles. Un Cube Unifié est suffisamment générique pour représenter aussi bien des données entreposées que des données ouvertes liées. Un cube unifié représente multi-dimensionnellement toutes ces données à l'aide d'un sujet d'analyse et d'un ensemble d'axes d'analyse composés de hiérarchies. L'intégration de ces différentes sources dans un modèle unifié nécessite d'étendre le concept de niveau de granularité d'analyse en y associant différents attributs provenant de différentes sources et en proposant des chemins d'agrégation supplémentaires reposant sur plusieurs sources.

En complément de cette modélisation, nous proposons un processus de construction d'un Cube Unifié en deux étapes. La première étape consiste à transformer différents schémas exprimés avec des langages de modélisation spécifiques en une représentation conceptuelle commune, appelé cube d'exportation. La deuxième étape permet d'associer et d'unifier les données provenant de différents cubes d'exportation. Un opérateur algébrique est proposé afin de traduire les besoins des concepteurs en des relations inter-données. Cet opérateur est implanté à l'aide d'un algorithme pour automatiser l'établissement des relations entre les données inter-sources et garantir la validité globale du Cube Unifié obtenu.

Afin de prouver la faisabilité de notre solution, nous avons développé un prototype implantant un Cube Unifié basé sur un ED R-OLAP (Relational On-Line Analytical Processing) et deux collections de données ouvertes liées. Nous montrons les avantages de notre proposition en identifiant et explicitant des analyses définies sur un cube unifié construit à partir de données entreposées et de DOL.

KEYWORDS: Multidimensional Modeling; Linked Open Data; Unified Business Analyses

MOTS-CLES: Modélisation Multidimensionnelle; Données Ouvertes Liées; Analyses Décisionnelles Unifiées

Organizational Memory: a model based on a heterogeneous network and an automatic information integration process

Une mémoire d'entreprise basé sur un réseau hétérogène et un processus d'intégration automatique d'information

Jérémy Bascans[2,3] — Max Chevalier[2] — Patrice Gennero[3] — Chantal Soule-Dupuy[1]

[1] IRIT, Université Toulouse 1 Capitole, Faculté d'Informatique
2 rue du Doyen-Gabriel-Marty, 31042 Toulouse, France
[2] IRIT, Université Toulouse 3 Paul Sabatier
118 route de Narbonne, F-31062 Toulouse Cedex 9, France
[3] SmartKiwi, Parc Technologique du Canal
10 avenue de l'Europe, 31520 Ramonville-Saint-Agne

{jeremy.bascans, max.chevalier, chantal.soule-dupuy}@irit.fr,
patrice.gennero@smartkiwi.net

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

RÉSUMÉ. Une mémoire d'entreprise est un espace de capitalisation d'informations diverses circulant au sein d'une entreprise. Les mémoires d'entreprise sont très importantes car elles permettent de stocker « les connaissances » de l'entreprise. Cependant, l'effort demandé à l'utilisateur pour intégrer/maintenir les informations dans la mémoire est très important. L'objectif de nos travaux est de représenter cette mémoire sous forme d'un réseau hétérogène d'informations sur lequel va reposer un processus automatique d'intégration des informations visant à assister les usagers tout en limitant leurs efforts. Nous avons développé un prototype, basé sur le modèle proposé, qui nous a permis de réaliser une évaluation de sa capacité à intégrer de nouvelles informations dans la mémoire.

ABSTRACT. Organizational memory is a space where various information circulating in a company are capitalized. From the users' point of view, organizational memories, which can be seen as an information system component, is very important since it store the "shared knowledge" of the organization. But, at the same time, the price of this knowledge is relatively high since users' participation, i.e. to integrate/maintain… the memory, is important. The aim of our work is to model such an organizational memory through a heterogeneous network on which will be based an automatic integration process of information to assist users in this task while limiting their effort. We developed a prototype

and evaluated through an experiment its ability to integrate new information into an organizational memory based on the proposed model.

MOTS-CLÉS : réseau hétérogène, mémoire d'entreprise, organisation de l'information, capitalisation de l'information, intégration automatique des informations.

KEYWORDS: heterogeneous network, organizational memory, organizing information, information capitalization, automatic information integration.

1. Résumé

L'information représente aujourd'hui un capital non négligeable pour les entreprises. De ce fait, les systèmes d'information (SI) dont le rôle principal est de permettre la collecte, le stockage, le traitement et la diffusion de ces informations, doivent évoluer avec un nouvel objectif de capitalisation et de partage de connaissances. Une évolution des SI pour l'intégration de mémoires d'entreprise (ME) a ainsi été constatée (Basaruddin *et al.*, 2011). Ainsi, les mémoires d'entreprises sont devenues une composante importante des systèmes d'informations. Cependant, l'implantation d'une ME nécessite parfois des modifications des processus métiers de l'entreprise en impactant directement la tâche des usagers (Zacklad, 2011, Doria, 2010). Par ailleurs, l'implantation d'une ME au sein d'une entreprise nécessite une forte implication des usagers qui doivent, manuellement ou presque, alimenter en information cette mémoire (Mas *et al.*, 2008). Or, (Ackerman *et al.*, 1996) a constaté dans ce contexte que les systèmes basés sur une action communautaire ne perdurent dans le temps que lorsque plusieurs personnes l'alimentent régulièrement. Aussi, en raison des efforts et de l'implication demandés, de tels systèmes tombent souvent à l'abandon.

Dans ce contexte, nous nous intéressons à l'organisation et l'intégration automatique des informations dans une mémoire d'entreprise. Cette organisation a pour objectif d'être générique et adaptative. Dans cet article, nous proposons une ME adaptable à toute structure d'entreprise et ayant pour objectif de limiter l'effort des usagers nécessaire pour (ré)organiser les informations. Pour ce faire, nous définissons cette mémoire sous la forme d'un réseau d'informations hétérogènes (Flakes *et al.*, 2002). Les informations sont organisées autour de la notion d'« Objets d'intérêt » (nœuds centraux du réseau), ce qui permet une organisation cohérente et contextualisée de ces informations.

Un processus d'intégration automatique sur la base de la composition de notre réseau a été proposé afin de prendre en compte l'hétérogénéité des informations. Ce processus est non intrusif. De plus, l'utilisateur n'a pas à connaître l'organisation de la ME, il n'a qu'à dire qu'il souhaite que telle ou telle information soit intégrée. De plus, il est impératif que, pour répondre aux pratiques et aux usages de toute entreprise, il soit possible d'ajouter de nouveaux processus d'intégration automatiques des informations.

Enfin, un prototype, sur la base des propositions, a été implanté. Ce dernier peut être extensible (type de liens, mesures…). Afin de vérifier si l'exploitation de

ces éléments et l'utilisation du réseau proposé permettent de garantir une cohérence maximale, nous avons mis en place une évaluation avec une collection d'« Objets d'intérêt » et de documents à intégrer provenant de données Wikipédia. Dans cette expérimentation, nous avons évalué la mémoire obtenue et démontré que notre prototype a correctement lié la majorité des documents avec les « Objets d'intérêt ». Cette évaluation met en évidence que dans un cadre général non déterministe, l'utilisation de l'organisation du réseau proposé permet l'intégration automatique de nouvelles informations tout en conservant une organisation les rendant intelligibles.

Ces travaux ont posé les principes de base pour la définition d'une mémoire d'entreprise et de nombreuses perspectives restent ouvertes. Nous souhaitons vérifier que la mémoire réponde à tous les verrous issus des objectifs imposés par cette mémoire, notamment en termes d'adaptabilité et généricité, en l'expérimentant via différents scénarios conçus avec plusieurs entreprises partenaires de SmartKiwi. Ensuite, il faudra proposer les moyens pour une entreprise de définir ses propres processus d'intégration automatique d'informations dans la mémoire. En outre, nous prévoyons d'analyser d'avantage la proportion des erreurs faites par le prototype dans l'évaluation. En effet, une grande partie pourrait être réduite par une nouvelle étude basée sur un échantillon analysé par plusieurs personnes (mesure de commensurabilité par des tests Kappa). Enfin, un des verrous majeurs pour la mise en place d'une ME, et surtout garantir sa pérennité, est de pouvoir gérer son évolution dans le temps de façon automatique, et la moins intrusive possible. Pour ce faire, nous envisageons de mettre en place un système d'apprentissage basé sur des interactions avec les utilisateurs qui pourront s'exprimer sur l'intégration proposée et faire des propositions.

Bibliographie

Ackerman M. S., McDonald D. W., « Answer Garden 2: merging organizational memory with collaborative help », *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, 1996, p. 97-105.

Basaruddin S., Haron H., Noordin, S. A., « Understanding Organizational Memory System for Managing Knowledge », *International Conference on Advancements in Information Technology ICAIT 2011*, IPCSIT vol. 20, 2011, Singapour, IACSIT Press.

Doria O. D., Zacklad M., « Améliorer la recherche d'information à l'aide de thésaurus « ad hoc » », *Document numérique*, vol. 13, n° 2, 2010, p. 13-40.

Flakes G. W., Lawrence S., Giles C. L., Coetzee F. M. « Self-organization of the web and identification of communities. » *IEEE Computer*, 2002.

Mas S., Bénel A., Cahier J. P., Zacklad M., « Classification à facettes et modèles à base de points de vue : Différences et complémentarité », *Actes du 36e congrès annuel de l'Association canadienne des sciences de l'information (ACSI)*, 2008, University of British Columbia, Vancouver, p. 5-7.

Zacklad M., Desfriches-Doria O., Bertin G., Mahe S., Ricard B., Musnik, N., *et al.*, « Miipa-Doc: Gestion de l'hétérogénéité des classifications documentaires en entreprise », *Actes de la onzième édition de la conférence internationale H2PTM (Hypertextes et Hypermédias)*, 2011, Hermès, p. 323-333.

Increasing Secondary Diagnosis Encoding Quality Using Data Mining Techniques

L'Augmentation de la Qualité de Codage de Diagnostic Secondaire en Utilisant des Techniques de Fouille de Données

Ghazar Chahbandarian[1], Nathalie Bricon-Souf[1], Rémi Bastide[1], Jean-Christoph Steinbach[2]

1.University of Toulouse, IRIT/ISIS
F-81100 Castres, France
{ghazar.chahbandarian, nathalie.souf, remi.bastide}@irit.fr

2. Department of Medical Information
Centre Hospitalier Intercommunal de Castres Mazamet
F-81100 Castres, France
jean-christophe.steinbach@chic-cm.fr

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS

RESUME. *Afin de mesurer l'activité médicale, les hôpitaux sont tenus de coder manuellement des informations concernant les séjours des patients hospitalisés en utilisant la Classification internationale des maladies (CIM-10). Cette tâche prend du temps et nécessite une formation importante pour le personnel, en particulier pour le codage des diagnostics associés (secondaires) qui ne sont pas toujours bien décrits dans les ressources médicales telles que la lettre de sortie et les dossiers médicaux. Nous proposons d'explorer des outils pour faciliter la tâche fastidieuse de codage de tels diagnostics.*

Notre approche exploite des techniques de fouille de données et plus précisément les arbres de décision dans le but d'explorer les bases de données médicales, et en particulier les diagnostics associés précédemment codés. On utilise les informations structurées comme l'âge, le sexe, le nombre de diagnostics, les actes médicaux présents dans les bases PMSI pour construire un arbre de décision facilement exploitable par un non spécialiste en informatique tel qu'un médecin, afin de souligner les diagnostics associés pour un séjour donné. Nous avons utilisé des données anonymisées extraites de la base de données PMSI de l'hôpital "Centre Hospitalier Intercommunal de Castres Mazamet", il contient environ 90.000 séjours d'hospitalisation entre 2011 et 2014.

Deux niveaux de granularité de diagnostic sont disponibles selon que l'on choisit de représenter le diagnostic de façon très précise (bas niveau de granularité) ou en se contentant de garder une information plus générale (haut niveau de granularité correspondant aux catégories de diagnostics. Les résultats indiquent qu'une amélioration de la performance pourrait être obtenue en utilisant le bas niveau de granularité de diagnostics et en équilibrant la répartition des exemples négatifs et positifs dans l'ensemble de l'apprentissage. En revanche, nous avons trouvé qu'il y a une variation entre les scores d'évaluation des diagnostics étudiés, par exemple, le score le plus élevé est 75% en utilisant la mesure F1 et le score le plus bas est 25% en utilisant la même mesure. En conséquence, des améliorations supplémentaires sont nécessaires pour obtenir une meilleure performance quel que soit le diagnostic codé. Cependant, le score moyen de tous les diagnostics associés étudiés est d'environ 80% en utilisant la mesure "accuracy", ce qui indique la prédiction des exemples négatifs est meilleur donc il pourrait être utile dans la prévention ou la détection des codages erronées dans les séjours hospitalisés.

.

ABSTRACT. In order to measure the medical activity, hospitals are required to manually encode information concerning an inpatient episode using International Classification of Disease (ICD-10). This task is time consuming and requires substantial training for the staff. We propose to help by speeding up and facilitating the tedious task of coding patient information, specially while coding some secondary diagnoses that are not well described in the medical resources such as discharge letter and medical records. Our approach leverages data mining techniques in order to explore medical databases of previously encoded secondary diagnoses and use the stored structured information (age, gender, diagnoses count, medical procedures…) to build a decision tree that assigns the proper secondary diagnosis code into the corresponding inpatient episode or indicates the impatient episodes that contains implausible secondary diagnoses. The results suggest that better performance could be achieved by using low level of diagnoses granularity along with adding some filters to balance the repartition of the negative and positive examples in the training set. The obtained results show that there is big variation in the evaluation scores of the studied diagnoses, the highest score is 75% using F1 measurement and the lowest 25% using F1 measurement which indicates further enhancements are needed to achieve better performance regardless of the encoded diagnosis. However, the average accuracy of all the studied secondary diagnoses is around 80% which indicates better negative predictions therefore it could be useful in the prevention or the detection of wrong coding assignments of secondary diagnoses in the inpatient stay.

MOTS-CLES : Fouille de données, apprentissage, arbre de décision, codage CIM-10.

KEYWORDS: Data mining, Machine learning, Decision tree, coding ICD-10

Data schema does matter, even in NoSQL Systems!

De l'importance d'un schéma de données dans les systèmes NoSQL

Paola Gómez[1], Rubby Casallas[2], Claudia Roncancio[1]

1. LIG, Université Grenoble Alpes
{paola.gomez-barreto,claudia.roncancio}@imag.fr

2. TICSw, Universidad de los Andes, Bogotá, Colombia
rcasalla@uniandes.edu.co

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

RESUME. La plupart de systèmes « NoSQL » n'utilisent pas de schéma de bases de données. Ces systèmes n'offrent donc pas de fonctions de gestion de schéma et la définition de la gestion des structures de données utilisées reviennent à l'application. Les données peuvent être semi-structurées et l'utilisateur a de nombreuses possibilités de structuration.

L'absence de schéma prédéfinie présente certainement des avantages de flexibilité. Cependant, ceci a aussi un coût en termes de volume des données stockées, de performances des requêtes, de clarté du code applicatif et, a fortiori, la mise au point et la maintenance des applications. Ceci est démontré dans cet article où nous présentons une étude de l'impact de la structuration des données dans la base de documents MongoDB.

Nous avons expérimenté avec une variété d'alternatives de structuration et une série de requêtes avec complexité croissante. Cet article présente notre analyse et conclusions suite à cette expérimentation.

ABSTRACT. A Schema-less NoSQL system refers to solutions where users do not declare a database schema and, in fact, its management is moved to the application code. This paper presents a study that allows us to evaluate, to some extent, the data structuring impact.

The decision of how to structure data in semi-structured databases has an enormous impact on data size, query performance and readability of the code, which influences software debugging and maintainability. This paper presents an experiment performed using MongoDB along with several alternatives of data structuring and a set of queries having

Session commune avec RCIS : IS Methods and Method Engineering

Progressive Integration of Method Components: A Case of Agile IS Development Methods

Intégration progressive de composants de méthodes : Application pratique avec des méthodes de développement agiles

Rébecca Deneckère[1], Elena Kornyshova[2], Adrian Iacovelli[1]

1. Centre de Recherche en Informatique
Université Paris 1 Panthéon-Sorbonne
90 rue de Tolbiac,
75013 Paris, France

2. CEDRIC
Conservatoire National des Arts et Métiers 2
rue Conté
75003 Paris, France

rebecca.deneckere@univ-paris1.fr, elena.kornyshova@cnam.fr, adrian.iacovelli@gmail.com

ARTICLE ACCEPTE ET PRESENTE A LA CONFERENCE INTERNATIONALE RCIS, CO-LOCALISEE AVEC INFORSID 2016 A GRENOBLE.

LA VERSION LONGUE DE L'ARTICLE, EN ANGLAIS, EST DISPONIBLE DANS LES ACTES DE RCIS.

RÉSUMÉ. L'ingénierie des méthodes situationnelles cherche à construire des méthodes adaptées à une situation donnée, soit par construction à partir de composants de méthodes déjà définis, soit par l'adaptation d'une méthode existante (en utilisant diverses techniques : configuration, extension, réduction, etc.). Cependant, ces techniques sont peu utilisées dans l'industrie car elles sont considérées comme compliquées et très lourdes à implémenter. Dans cet article, nous décrivons une expérience pratique d'intégration progressive de plusieurs composants de méthodes issus des méthodes agiles. Ce cas réel appliqué dans une entreprise de développement nous a permis d'étudier et d'observer l'introduction progressive des composants de méthodes en lieu et place d'une construction ou d'une adaptation en un coup. Nous discutons les leçons apprises de cette expérience et définissons différentes perspectives de recherche dans le domaine.

Congrès INFORSID, 34ème édition, Grenoble, 31 mai – 3 juin 2016

ABSTRACT. *Situational Method Engineering aims at constructing methods adapted to a given situation, either by a construction from a set of predefined method components or by a customization of an existing method using different techniques: configuration, extension, reduction, and so on. However, these techniques are still limited in practice, as considered complicated and heavy to implement. In this paper, we describe a practitioner experience of a gradual integration of different method components (issued from agile methods of software development). In a real case of a development company, we have practiced and observed the progressive introduction of agile method components instead of the construction or customization of methods in one go. We discuss the lessons learned from this experience and define different research perspectives.*

MOTS-CLÉS : *Ingénierie des méthodes situationnelles, Composants de méthodes, Méthodes agiles, Intégration progressive, rapport d'expérience.*

KEYWORDS: *Situational Method Engineering, Method Component, Agile Method, Progressive Integration, Experience Report.*

1. Introduction

L'ingénierie des méthodes situationnelles (IMS) statue qu'une méthode de développement de système doit être alignée avec le contexte du projet sur lequel elle s'applique. En effet, chaque situation est différente et nécessite un support méthodologique différent. Dans cet objectif, l'IMS propose des méthodes de construction spécifique et adaptable selon la situation du projet en cours en réutilisant des parties de méthodes existantes, appelés composants, stockés dans des bases de méthodes. Bien que plusieurs approches de construction existent, leur implémentation dans le contexte industriel est difficile. Les entreprises reconnaissent le bien fondé de ces approches et de ces techniques mais trouvent leur implémentation difficile et couteuse.

Une autre manière d'utiliser les techniques d'IMS de manière plus douce est d'intégrer les composants de manière progressive, un à la fois, et d'attendre que les utilisateurs soient à l'aise avec les premiers changements avant d'aller vers une autre modification. Nous proposons dans ce travail le résultat d'une expérimentation exécutée dans une entreprise utilisant déjà quelques techniques de gestion de projet mais souhaitant améliorer son processus de développement, a priori par l'introduction de techniques venant des méthodes agiles. Cette entreprise ne souhaitait pas un changement trop rapide en implémentant une méthode agile dès le départ. Le but de ce travail était de vérifier qu'une intégration progressive était possible et que cela induisait une meilleure acceptation de la méthode produite par les professionnels.

Nous avons tout d'abord étudié la méthode utilisée par l'entreprise dans le but d'identifier les techniques déjà existantes et celles qu'il serait souhaitable d'implémenter. Nous avons ensuite sélectionné un ensemble de composants correspondants à ces techniques. Un plan d'intégration a été défini sur le long terme (d'une durée de deux ans) pour s'adapter à l'acceptation du changement dans

l'entreprise. L'ingénieur de méthodes responsable de l'expérimentation faisait partie de l'entreprise en tant que développeur et a pu aider à l'intégration tout au long de l'expérience. De bons résultats ont été obtenus puisque tous les composants sélectionnés ont été intégrés dans la méthode de départ et que les professionnels ont acceptés les changements effectués.

2. Contexte organisationnel

L'entreprise capitalise plus de 10 ans de recherche et développement dans le cloud computing et le big data. Elle est spécialisée dans le développement de systèmes d'informations complexes, particulièrement dans le domaine de la santé et de la recherche biomédicale.

L'entreprise travaillait sur plusieurs projets en même tems et utilisait une méthode de développement essentiellement basée sur une méthode de développement classique. L'équipe avait une réunion hebdomadaire pour discuter des taches à effectuer pendant la semaine. De nouvelles taches pouvaient être ajoutées et les membres de l'équipe pouvaient discuter de leur faisabilité. L'équipe utilisait un googledoc pour sauver et partager les minutes des réunions. Une nouvelle version du googledoc était créée chaque semaine. L'outil Redmine[1] était utilisé pour gérer le projet mais son usage était limité à la définition des taches de haut niveau et à la décomposition de ces taches en sous taches. La durée de réalisation des taches était également intégrée dans cet outil.

L'organisation de la gestion de projet avait plusieurs problèmes. Le premier était la définition des taches. Les taches identifiées étaient de très haut niveau et pas assez détaillée. De plus, leur formulation était très informelle. Le suivi de projet était un autre souci. Le temps de réalisation des taches était spécifié dans le google doc mais cela était fait à un niveau très haut et la hiérarchie des taches n'était pas toujours à jour. Le troisième problème était le manque d'un outil spécifique pour aider dans les taches de gestion de projet. Un autre problème concernait l'intégration des nouveaux besoins et les retours utilisateurs. Il n'y avait pas de réunion centrées sur les utilisateurs, chaque nouveau besoin ou retour utilisateur était soit traité en temps réel sans gestion des priorités, soit mis de côté pour une période indéterminée, ce qui engendrait des complications sur le suivi de ces taches. Tous ces problèmes étaient reliés au manque de méthode ou d'outil de gestion de projet. L'idée d'intégrer une nouvelle méthode de développement agile dans l'entreprise ne souleva pas l'enthousiasme dans l'équipe. En effet, la méthode utilisée n'était pas parfaite mais elle fonctionnait et donnait des résultats. Les membres de l'équipe pensaient que cela leur demanderait trop d'efforts et de temps pour un gain peut-être minime. C'est à ce moment que nous sommes intervenus pour leur proposer une intégration progressive des composants de méthodes agiles qui leur manquaient.

[1] http://www.redmine.org/

UIPLML: un outil basé sur les patrons pour l'ingénierie des systèmes d'information multi-plateformes

Thanh-Diane Nguyen[1], Jean Vanderdonckt[1], Ahmed Seffah[2]

1-Louvain School of Management, Université catholique de Louvain, Place des Doyens, 1, B-1348 Louvain-la-Neuve, Belgium
{thanh-diane.nguyen, jean.vanderdonckt}@uclouvain.be

2-Innovation and Software, School of Business and Management, Lappeenranta University of Technology, P.O. Box 20, FI-5385, Lappeenranta (Finland)
ahmed.seffah@lut.fi

Article accepté et présenté à la conférence internationale RCIS, Co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

RÉSUMÉ. Les patrons de conceptions (pattern en anglais) sont un outil qui a fait ses preuves à la fois dans la communauté IHM et en génie logiciel pour l'ingénierie des systèmes interactifs. Cependant, tel que documenté aujourd'hui, la grande majorité des patrons de conception des interfaces utilisateurs ne supportent pas le développement et l'adaptation des interfaces dites multi-contextes d'utilisation (multiplateformes, caractéristiques et expériences des utilisateurs et multi-environnement logiciel). L'utilisabilité n'est pas aussi prise en compte explicitement. Cette faiblesse rend aussi difficile la tâche des développeurs et gestionnaires des systèmes d'information de choisir le modèle de conception adéquat pour un cas particulier d'utilisation. Cet article introduit le langage UIPLML (User Interface Pattern Langue Markup Language) qui exploite un format standardisé basée sur le XML qui permet d'apporter une solution satisfaisante à l'ensemble de ces faiblesses. Ce langage utilise le concept de patron génératif qui va au-delà de la simple documentation des patrons ; il propose d'intégrer dans la description des patrons, la stratégie de génération de code et d'adaptation de l'interface. Pour illustrer ce langage UIPLML, une application est présentée en utilisant le patron de conception : Master/Détails pattern. L'exemple décrit, en plus d'une définition complète, ce pattern incluant l'utilisabilité et la génération d'un fichier XML pour faciliter son implémentation dans le contexte des multiplateformes.

ABSTRACT (As it appears RCIS Version): Information systems are more accessible in the form service (Information systems as a service) by anybody from everywhere, anywhere, and at any time, from almost any device and computing platform. The continuous growth and the heterogeneity of these devices create various user experiences depending on the device and challenge designers and developers to creating methods and tools for engineering of usable, yet accessible, information systems. Instead of repeating a similar development life cycle, design patterns concentrate design solutions with embedded usability and accessibility. However, when an appropriate pattern is selected, the developer is responsible for adequately program the corresponding code, which is a tedious and error-prone task. In order to address these challenges, this paper motivates, presents, and defines UIPLML (User Interface Pattern Language Markup Language), a XML-compliant markup language for defining user

interfaces patterns for multiple contexts of use, e.g., for different users carrying out a task on different devices in different environments. A meta-model with new expressiveness enable multi-facet pattern matching. To validate it, four UIPLML pattern databases have been created: a base of 237 entries for multi-platform systems, a base of 42 entries for context-aware interfaces, a base of 10 entries for culturally-aware interfaces, and a base of 52 entries for accessibility issues. One particular pattern, i.e. the master/detail, is supported by a software for generative pattern-based approach in which application parameters and contextual data govern automated user interface XML creation which, in turns, generates code for multi-context information systems.

MOTS-CLÉS *: interface utilisateur multiplateformes; contexte d'utilisation, génération de code,*

KEYWORDS*: Generative design pattern; multi-platform user interfaces, context of use, code generation, interactive information systems engineering*

Résumé étendu

L'émergence d'une multitude de plateformes offre aussi une grande diversité d'accès aux systèmes d'information tout en supportant une très grande richesse en matière d'interaction. Les interfaces utilisateurs (IUs) doivent s'adapter à ces variations de contexte (utilisateurs, plateformes, environnements logiciel, organisation) afin de permettre à l'utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction.

Cependant, nous voyons émerger de plus en plus des plateformes différentes supportant des règles ergonomiques très variables de conception et d'évaluation d'IUs. Les développeurs et les designers d'IUs ont deux solutions pour résoudre ce problème. La première consiste à développer une interface pour chaque plateforme en respectant les critères d'utilisabilité propre à la plateforme. Cette solution n'est bien sûr pas techniquement viable compte tenu qu'elle nécessite une armée de développeurs. La seconde, celle que nous préconisons consiste à développer et adopter une approche de conception dirigée par les modèles qui utilisent les patrons de conception comme moyen d'encapsuler les règles ergonomiques et comme outil d'instanciation de modèles et de génération de code.

Les principes généraux de cette approche de conception dirigée par les modèles et basée sur les patrons, ont été largement discutés dans la littérature scientifique (Seffah, 2015). La section suivante donne un aperçu sur cet état de l'art. Il est important de noter que cet article discute uniquement de comment les patrons peuvent être documentés en intégrant les règles ergonomiques générales (adaptables sur toutes les plateformes) et spécialisés (accommodées sur une plateforme spécifique).

Les patrons comme : (1) outil de conception des interfaces (Wolff and Forbrig, 2010) (Märtin et al., 2013) (Seffah et al, 2007 et (2) comme modèle pour la génération du code dirigé par les patrons (Vanderdonckt and Simaro, 2010), ont été

largement décrits dans la littérature. Différentes collections de patrons de conception d'IHM sont ainsi disponibles (van Welie *et al*., 2003).

Cependant, lorsque l'on examine ces langages et leur applicabilité dans l'approche de conception-dirigée par les modèles et basée sur les patrons, on peut observer les faiblesses suivantes:

- Un manque de consistance entre les différents formats de description de patterns (différents taxonomies, niveau de détail et présence d'homonymes).
- Un manque de structure de documentation en particulier dans l'uniformité dans les attributs de description,
- L'absence de de suggestions d'implémentation du pattern pour différentes plateformes.
- Un manque de liens avec les approches de développement logiciel, comme par exemple l'approche par modèles qui est pourtant préconisée pour le développement des interfaces multiplateformes (Molina et al., 2002).
- Un manque d'outillage permettant l'automatisation de l'application des patrons et de la génération de code.

Différentes extensions de ces langages de patrons ont été défini pour répondre à différents contextes d'utilisation (Engel *et al.*, 2015, Seffah, 2015). La première extension a été formulée par le groupe PLML (Pattern Langage Markup Langage) (Fincher, 2006). Différentes variations de PLML ont émergé afin de répondre à différents problèmes (Wendler et al, 2013). En comparant toues les extensions disponibles dans la littérature scientifique, nous remarquons que l'aspect multiplateformes et les critères ergonomiques dans ces interfaces ne sont pas suffisamment pris en compte et dans un même lieu.

Le langage UIPLML (User Interface Pattern Markup Langage) que nous proposons est fondé sur le langage et le framework de développement des interfaces, UsiXML (Vanderdonckt, 2012). Il tire profit aussi du standard de W3C également utilisé dans le langage étendu PLML. Un aspect fondamental de UIPLM est un méta-model qui spécifie les règles ergonomiques, par exemple de (Scapin and Bastien, 1997). Ces règles ergonomiques ont été sélectionnées selon divers critères de l'utilisabilité et proviennent d'une sélection de références diversifiées, populaires et pertinentes. Les patrons de conception intégrés dans le langage UIPLML utilisent les règles ergonomiques décrites dans la base de données DESTINE (Figure 1). La base de donnée contient 256 enregistrements pour les systèmes multiplateformes, 42 enregistrements sur le contexte d'utilisation, 10 sur la culture et 51 sur l'accessibilité (voir table 1).

Type de pattern	Nombre d'Enregistrement	Objectif
Multi-plateformes patterns	237	Les patterns pour les smartphones, tablettes, ordinateur portables, PC, Pocket PC, PDA sont structurés en 8 catégories (taille de l'écran, page d'accueil, menu, contenu, actions, formulaires, contact)
Context-aware patterns	42	Les patterns sont classés selon la sensibilité du contexte dans les interfaces hommes machines
Culture-aware patterns	10	Les patterns ont classées selon des critères culturels (par exemple, les choix de couleurs, le langage, etc.)
Accessibility patterns	51	Les patterns sont classées selon le dégré d'accessibilité pour différentes plateformes

Table 1. UIPLML Patterns dans la base de donnée de Destine Application

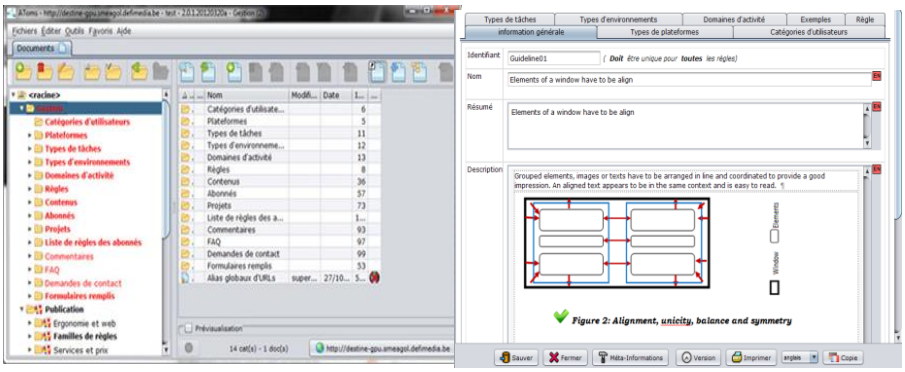


Figure 1. Editeur Correspondant à UIPLML Pattern

Adopter l'approche basée sur les patrons de conception permet d'assurer l'utilité et l'utilisabilité des produits pour les utilisateurs. En effet, les patterns décrivent les problèmes des utilisateurs avec une interface utilisateur au niveau conceptuel. Cela permet de les réutiliser comme briques de construction et de génération de code au niveau de l'implémentation. Leur force se trouve donc dans la description à haut niveau d'abstraction et l'illustration de leur adaptation qui permet rendre la tâche du développeur et désigner plus aisée.

La figure 2 présente l'outil proposé de génération de code à partir de patterns. La figure illustre aussi comment les développeurs peuvent concevoir des interfaces graphiques en utilisant des modèles de haut niveau (image centrale de la figure 2) accompagné d'une structure (image à droite[1] de la figure 2) dans un système multiplateforme.

[1] Une version électronique de l'arbre de décision est disponible à l'adresse suivante : http://usimad.alwaysdata.net/tree

Pour illustrer le fonctionnement de cet outil, nous avons utilisé le cas le Master/Details Pattern (Molina *et al.*, 2002). Ce patron de conception permet de présenter une liste d'éléments (le Master). La sélection d'un élément permet de détailler celui-ci. La description de ce pattern dans le langage UIPLML inclut la spécification sous forme d'arborescence des différents modèles de représentation possibles avec des « widgets » d'affichage correspondants aux multiplateformes. La spécification est donnée dans un fichier XML. Celui-ci permet surtout de faire un lien entre les patterns et les règles ergonomiques. Il permet aussi de combiner ainsi l'expérience des patrons de conception et l'utilisabilité dans l'implémentation des interfaces utilisateurs multiplateformes.

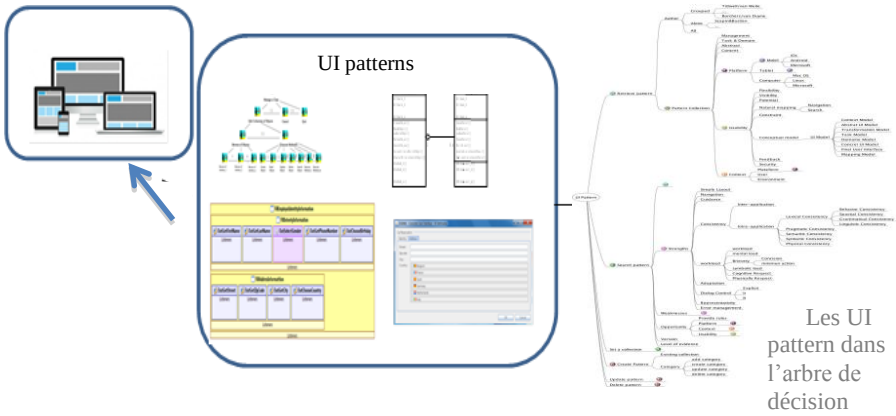


Figure 2. UIPLML pour la conception des multiplateformes

Le langage UIPLML proposé est loin d'être finalisé. Nous avons motivé dans ce papier les raisons et les principes de son développement. Nous avons aussi illustré brièvement l'usage de ce langage en utilisant le cas d'un patron particulier.

References

Engel J., Märtin C. and Forbrig P., (2015). A Concerted Model-driven and Pattern-based Framework for Developing User Interfaces of Interactive Ubiquitous Applications, *in Proc. of 1st Int. Workshop on Large-scale and Model-based Interactive Systems: Approaches and Challenges, LMIS'2015* (Duisburg, June 23, 2015), CEUR Workshop Proceedings, vol. 1380, CEUR-WS.org 2015, pp. 35-41.

Fincher S., (2006). *PLML: Pattern Language Markup Language*, University of Kent, February 2006, accessible at https://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html

Märtin C., Herdin C. and Engel J. (2013). Patterns and Models for Automated User Interface Construction - In Search of the Missing Links, in *Proc. of 15th Int. Conf. on Human-Computer Interaction: Human-Centred Design Approaches, Methods, Tools, and Environments HCI International'2013* (Las Vegas, July 21-26, 2013).

Molina PJ., Santiago M. and Pastor O., (2002). User interface conceptual patterns," in *Proc.*

of the 9th Int. Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS'2002 (Rostock, June 12-14, 2002), Lecture Notes in Comp. Science, vol. 2545, pp. 159-172, 2002.

Scapin D. and Bastien JMC. (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems, Behaviour & Information Technology, vol. 16, no. 4/5, pp. 220-231, 1997.

Seffah A., (2015). Patterns of HCI design and HCI design of patterns - Bridging HCI design and model-driven software engineering, *Human Computer Interaction Series*, Berlin: Springer.

Van Welie M. Van der Veer G., (2003). Pattern languages in interaction design: structure and organization, in *Proc. of IFIP TC13 Int. Conf. on Human-Computer Interaction INTERACT'2003* (Zurich, September 1-5, 2003), Zurich: IOS Press, pp. 527-534, 2003.

Vanderdonckt J. and Simarro F. (2010). Generative pattern-based design of user interfaces, in *Proc. of the 1st Int. Workshop on Pattern-Driven Engineering of Interactive Computing Systems PEICS'2010* (Berlin, June 20, 2010), New York: ACM Press, pp. 12-19, 2010.

. Wendler S., Ammon D., Philippow I. and Streitferdt D. (2013), A factor model capturing requirements for generative user interface patterns, in *Proc. of 5th Int. Conf. on Pervasive Patterns and Applications PATTERNS'2013* (Valencia, May 27-June 1, 2013), Wilmington: Int. Acad., Research, and Industry Association, pp. 34-43, 2013

Session commune avec RCIS : Requirement Engineering

Validation, accreditation or certification: a new kind of diagram to provide confidence

Validation, accréditation, certification : un nouveau diagramme pour établir la confiance

Thomas Polacsek

ONERA, Département Traitement de l'Information et Modèlisation
2, avenue Edouard Belin BP74025, 31055 TOULOUSE Cedex 4

Article accepté et présenté à la conférence internationale RCIS, co-localisée avec INFORSID 2016 à Grenoble. La version longue de l'article, en anglais, est disponible dans les actes de RCIS.

RÉSUMÉ. *Le but des arbres de l'argumentation est d'organiser et de visualiser, de manière synthétique, les principaux éléments prouvant la validité d'une propriété pour un produit. Un arbre d'argumentation ne représente pas un processus, il organise et donne à voir la rationalité sous-jacente à l'ensemble des documents de Vérification & Validation (V&V). En fait, un arbre d'argumentation répertorie et structure tous les éléments de preuve nécessaires dans un cycle de développement. Mais, la validation d'un arbre d'argumentation passe nécessairement par la validation et l'identification des éléments de preuve de chaque étape unitaire, de chaque étape intermédiaire, incluse dans l'arbre. Par conséquent, dans cet article, nous présenterons un modèle d'argumentation générique et ses dérivés qui permettent de structurer de façon logique tous les éléments de V&V pour chaque étape. Ce patron est issu de travaux provenant des sciences légales et des théories de l'argumentation et il est l'élément de base pour la construction des arbres d'argumentation.*

ABSTRACT. *The aim of argumentation tree diagram is to organize and visualize, in a synthetic way, all key elements proving the validity of a product's property. The argumentation tree does not represent the process, but gives the rational behind all Verification & Validation (V&V) documents. In fact, it lists and organizes necessary evidence in a development life cycle. But the validity of the final assessment requires the validation and the identification of the evidence of each intermediate step. So, in this article, we introduce a generic argumentation pattern and its derivations whose support a rational organization of all V&V evidence at each step. This pattern stems from legal science and argumentation theory legacies, and it is the basic building block for the argumentation tree construction.*

MOTS-CLÉS : *patron d'argumentation, vérification et validation, ingénierie des exigences.*

KEYWORDS: *argumentation pattern, verification et validation, requirements.*

Table Ronde et Ateliers

Ville digitalisée contributrice

Porteurs : Michel Léonard (ISS-CUI Genève/CINTCOM), Humbert Fiorino (Université Grenoble Alpes, Laboratoire d'Informatique de Grenoble), Chantal Soulé-Dupuy (Université Toulouse 1 Capitole & Institut de Recherche en Informatique de Toulouse), Antoine Burret (CINTCOM), Lionel Lourdin (CINTCOM), Anastasiya Yurchyshyna (CINTCOM).

Résumé :
Les poussées numériques sont telles qu'il devient de plus en plus fragile et de moins en moins responsable, de conduire une politique de développement durable d'une ville sans en tenir compte. Ce constat conduit à une question toute simple : oui, mais comment ?

Forte de toutes les connaissances, ingénieries, méthodes, plate-formes que la communauté INFORSID a établies depuis plus de 37 années dans le domaine des organisations et des entreprises, elle est bien placée pour relever ce défi.

Elle doit prendre en compte que le domaine est beaucoup plus vaste que celui d'une organisation ou d'une entreprise. Il s'agit de toute une ville ! Il ne s'agit pas seulement des aspects administratifs, mais il s'agit de créer les nouveaux environnements d'activités et d'échanges en tirant profit de toutes les potentialités offertes par les technologies numériques sans cesse émergentes. Il ne s'agit pas de rendre une ville intelligente : elle l'est déjà depuis des millénaires pour certaines d'entre elles. Il s'agit de la rendre digitalisée à l'aide d'un large mouvement d'innovation de services : certains sont construits sur des Big Data et/ou des Open Data, mais il y en a bien d'autres à créer. Tous doivent avoir une proposition de valeurs et une perspective de durabilité.

Comme 37 ans auparavant, où elle a dû se libérer des schémas cognitifs de la pensée informatique, notamment du génie logiciel, pour créer le domaine des « systèmes d'information », la communauté INFORSID doit maintenant se détacher d'une pensée purement numérique pour porter la question précédente au niveau « digital ». C'est ainsi qu'elle pourra supporter les nécessaires recherches interdisciplinaires, pour prendre en compte les nombreux enjeux éthiques, juridiques, économiques, politiques, médicaux, scientifiques, numériques, sociologiques, de culture, de marchés, d'employabilité, d'enseignement, de santé, de culture.

Dans un contexte démocratique, elle se doit d'apporter les éclairages scientifiques aux responsables politiques du développement des villes. Ils vont faire face à tout un essor digital engendré par de multitudes initiatives dans de nombreux domaines, qui vont rendre leur ville plus consistante. Il s'agit de rendre toutes ces initiatives durables, de limiter au maximum le gaspillage des énergies de développement et en conséquence du gaspillage financier, d'éviter au maximum de casser des initiatives pour des raisons obscures.

Ce sont des myriades de services à co-créer, co-développer dans des cadres de contribution entre des personnes s'engageant dans la création de services, des entreprises et associations nouvelles ou anciennes, des institutions publiques et les autorités politiques de la ville.

L'objectif de cet atelier est d'établir des bases consensuelles de ce nouvel essor de recherches.

SI pour l'aide à la décision et la diffusion d'alertes

Porteurs : Florence Sèdes (Université de Toulouse - IRIT), André Miralles (Irstea - UMR Tetis), Thérèse Libourel (Université de Montpellier / IRD - UMR EspaceDev), Thomas Polacsek (Onera)

Résumé :
Face aux divers enjeux environnementaux et sociétaux actuels, mais aussi face aux besoins de prédiction et de gestion des catastrophes naturelles (inondations, avalanches, etc.), de la veille sanitaire ou de la surveillance de l'espace, des SI spécifiques sont de plus en plus souvent mis en place, comme outil de diffusion d'alertes, pour aider les acteurs dans leur prise de décision. Dans ce contexte, ces SI ont une fonction de capitalisation de l'information indispensable à l'analyse spatiale et temporelle des phénomènes observés afin de diffuser des indicateurs appropriés. Ces systèmes, faisant parfois intervenir directement l'humain (en tant que capteur), doivent donc posséder les propriétés fondamentales de vivacité, de réactivité et de qualité de l'information, car toute alerte doit absolument atteindre le bon destinataire au bon moment et au bon endroit avec le bon niveau d'information. De plus, dans le cadre de systèmes ouverts, viennent s'ajouter à cela les problématiques de véracité et de confiance dans les alertes. Cet atelier a pour but de croiser, d'échanger et de partager les réflexions, les méthodologies et les expériences autour de la conception et l'implémentation de tels systèmes. Pour ce faire, il sera intéressant de confronter des points de vue différents sur :

- les verrous essentiels en termes de modélisation (notamment aux niveaux des échelles de temps et d'espace, du choix des indicateurs) et de mises en œuvre innovantes (réseaux de capteurs, collaboration, interaction, etc.),
- les définitions et formalisation de ces systèmes, leurs rôles et fonctionnalités, leur périmètre, leur architecture, leur gouvernance et leur connexion avec des observatoires,
- la dynamique et la réactivité de ces systèmes ainsi que les problèmes soulevés par la confidentialité, le filtrage d'information, et les moyens de distinguer information et rumeur.

Mots-Clés : Système d'information, Décision, Indicateur, Alerte/Alarme

Articles acceptés :

Une approche sémantique autonomique pour la détection et le suivi des maladies
Sarah Slimani, Adel Alti, Sébastien Laborie, Philippe Roose (LIUPPA, Pau)

SI pour aide à la décision stratégique et alerte en pharmacovigilance.
Yannick Bardie (Laboratoire MRM Montpellier Recherche Management et Société Voute SAS, Montpellier)

Aide à la décision pour le recueil incrémental d'expertises médicales
Clément Duffau, Mireille Blay-Fornarino (i3S, Nice)

Le réseau SAGIR : apport des outils informatiques pour la vigilance vis-à-vis des maladies se développant dans la faune sauvage
Anouk Decors (1), Florence Baurier (2), Frédéric Dej (3), Karin Lemberger (4), Jean-Yves Chollet (1), Dominique Gauthier (5)
(1) ONCF, Dir. RE, Auffargis, (2) ADILVA, Labo. d'analyses du Cher, Bourges, (3) ONCF, Dir. SI, Birieux, (4) Vet Diagnostics, Lyon, (5) ADILVA, Labo. vétérinaire et d'hygiène alimentaire des Hautes-Alpes, Gap

Une perspective de la mobiquité au service de la gestion avant / pendant / après des séismes
Anne-Marie Lesas (LSIS, Marseille)

Toward Free Spam Social Networks : Detecting and Tracking Spammers in Twitter
Mahdi Washha (IRIT, Toulouse)

Using seal trajectories in biological early warning system for real-time zone tracking
Rouaa Wannous, Jamal Malki, Alain Bouju, Cécile Vincent, Cyril Faucher (L3i, La Rochelle)

Retour d'expérience sur l'analyse des exigences centrée sur les données pour des SI *-data
Christophe Ponsard, Annick Majchrowski, Stéphane Mouton (CETIC, Charleroi, Belgique)

Enseignement des SI

Porteurs : Gaëlle Blanco-Lainé (Université Grenoble Alpes), Sophie Dupuy-Chessa (Université Grenoble Alpes, LIG), Nadine Mandran (CNRS, Université Grenoble Alpes, LIG)

Résumé :
Le domaine des systèmes d'Information (SI) est par nature pluri-disciplinaire (management, gestion, informatique, génie industriel ?). Cette pluri-disciplinarité n'est généralement pas reflétée dans les enseignements qui restent segmentés par discipline. Or elle est souvent primordiale afin de concrétiser et de donner du sens à des concepts complexes et abstraits.

L'objectif de cet atelier est de partager sur les pratiques d'enseignement des différents aspects des SI. Il présente des expériences pédagogiques innovantes (pédagogie par le jeu, par la tangibilité ?) ou pluri-disciplinaires (pédagogie par projet, par l'action ?). L'atelier vise à :

— Élaborer une cartographie des pédagogies innovantes pour l'enseignement des différents aspects des SI.
— Mettre en avant les objectifs, avantages et limites de chaque modalité pédagogique.

L'atelier présentera les différentes modalités pédagogiques envisageables en se basant sur les expériences des participants (par exemple, les approches agiles ou par le jeu). Dans un deuxième temps, une synthèse co-construite sera élaborée afin de mettre en avant des pistes de recommandations en terme d'usage de ces modalités.

Mots-Clés : Enseignement, pédagogie, expérience, pluri-disciplinarité

Sécurité des systèmes d'information : technologies et personnes

Porteurs : Pierre-Emmanuel Arduin (Université Paris-Dauphine), Káthia Marçal de Oliveira (Université de Valenciennes)

Résumé :
Alors qu'une carte perforée pouvait être dérobée pendant son trajet entre une perforatrice et une tabulatrice, une information d'aujourd'hui, dématérialisée, est soumise à d'innombrables menaces quant à sa sécurité. Les artefacts technologiques sont partout, l'informatique est ubiquitaire, si bien que l'individu n'est plus un simple utilisateur du système d'information, mais il en est un composant à part entière. Comment assurer alors la sécurité d'un système d'information en prenant en compte des technologies omniprésentes, universelles et discrètes, aussi bien que des personnes au comportement parfois irrationnel ?

Un système d'information peut être vu comme un ensemble de ressources numériques et humaines organisées afin de traiter, diffuser et stocker des informations. La sécurité d'un système d'information peut être abordée d'un point de vue technologique, certes, mais aussi d'un point de vue humain. La question des connaissances portées par les personnes, aussi bien que la confiance qu'elles accordent au système et que le système leur accorde sont cruciales pour assurer la sécurité du système d'information dans les organisations.

Dans cet atelier, les participants présenteront comment ils considèrent dans leurs recherches, leur activité professionnelle et leur vie quotidienne, que la sécurité des systèmes d'information peut être appréhendée. Chercheurs, utilisateurs et professionnels de l'industrie sont ainsi invités à venir discuter les thématiques suivantes (liste non exhaustive) :

— modèles de sécurité des systèmes d'information,
— solutions technologiques et / ou managériales pour la sécurité des systèmes d'information,
— management des connaissances dans les organisations,
— interaction Homme-Machine pour la confiance et la sécurité,
— modèles d'évaluation et construction d'indicateurs de confiance et de sécurité,
— failles de sécurité des systèmes d'information (failles physiques, failles systèmes, failles applicatives, failles humaines),
— retours d'expériences et études de cas en management de la sécurité des systèmes d'information.

L'objectif de cet atelier est d'être un espace d'échange entre chercheurs et professionnels de plusieurs domaines afin d'alimenter une réflexion commune sur la sécurité des systèmes d'information, d'un point de vue non seulement technologique, mais aussi humain.

Mots-Clés : Système d'information, sécurité, ingénierie sociale, confiance, management des connaissances, évaluation.

Articles acceptés :

Évaluation de la confiance dans un environnement multisources
Benjamin Coste (1), Cyril Ray (1), Gouenou Coatrieux (2)
(1) Ecole navale - CC 600, Brest, France
(2) Institut Mines-Télécom - Télécom-Bretagne, Technopole Brest-Iroise, Brest, France

Vers une ingénierie conjointe de la sécurité, de l'utilisabilité et de la résilience dans les systèmes socio-techniques
Wilson Goudalo (1, 2), Christophe Kolski (1), Frédéric Vanderhaegen (1)
(1) LAMIH-UMR CNRS 8201, Université de Valenciennes, Valenciennes, France
(2) Research and Innovation Department, Advanced Business Engineering, Lagny, France

Service Contracts : Beyond Trust in Service Oriented Architectures
Gloria Elena Jaramillo Rojas, Philippe Aniorte, Manuel Munier
Université de Pau et des Pays de l ?Adour, Mont de Marsan, France

Towards an ontology about trust and security of information systems : joining technology and human perspectives
Káthia Marçal de Oliveira (1), Bako Rajaonah (1), Pierre-Emmanuel Arduin (2), Inês Saad (3)
(1) LAMIH-UMR CNRS 8201, Université de Valenciennes, Valenciennes, France

(2) Université Paris-Dauphine, PSL Research University, Laboratoire DRM UMR CNRS 7088, Paris, France
(3) Université d'Amiens, Amiens, France

Comité de Programme :
Philippe Aniorté, Université de Pau et des Pays de l'Adour
Pierre-Emmanuel Arduin, PSL, Université Paris-Dauphine, DRM UMR CNRS 7088
Linda Atif, PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243
Henri Basson, Université de Lille Nord de France, Côte d'Opale
Cédric Campo-Paysaa, ON-X Groupe, Puteaux
Houcine Ezzedine, Université de Valenciennes, LAMIH UMR CNRS 8201
Michel Grundstein, PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243
Doudja Kabeche, AgroParisTech, DRM UMR CNRS 7088
Christophe Kolski, Université de Valenciennes, LAMIH UMR CNRS 8201
Elsa Negre, PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243
Káthia Oliveira, Université de Valenciennes, LAMIH UMR CNRS 8201
Dorian Petit, Université de Valenciennes, LAMIH UMR CNRS 8201
Philippe Ramadour, Aix-Marseille Université, LSIS
Camille Rosenthal-Sabroux, PSL, Université Paris-Dauphine, LAMSADE UMR CNRS 7243
Mustapha Sali, PSL, Université Paris-Dauphine, DRM UMR CNRS 7088
Ines Saad, ESC Amiens, Laboratoire MIS, Université de Picardie Jules Verne
Thierno Tounkara, Institut Mines Telecom, Télécom Ecole de Management, Évry

L'innovation par les SI dans l'écosystème

Porteurs : Camille Salinesi, Abdelkader Achi du CRI (Paris Sorbonne) et Nathalie Valles (IRIT Toulouse)

Résumé :
L'ère du numérique se caractérise par le besoin d'innover, la collaboration et le partage des nouveaux savoirs (Gartner, 2014). Au sein des organisations le rôle des SI se cherche entre centre de coûts et centre de valeurs. Une tendance forte de ces dernières années est l'utilisation des SI pour co-créer de la valeur avec les parties prenantes et innover. On assiste ainsi à la prolifération de nouveaux espaces de création de connaissances et d'innovation : crowdsourcing, communautés, médias sociaux, etc. et de nouveaux modèles d'innovation collaboratifs, inclusifs, ouverts à tous les acteurs de l'écosystème. Outre les cas de succès flagrants de digital natives tels que Google et Amazon, nous ne disposons pas encore de données chiffrées, d'études systématiques, de modèles, de bonnes pratiques, ou de référentiels qui permettraient aux organisations d'intégrer dans leur boîte à outils méthodologique les méthodes, techniques et outils leur permettant de mener une démarche systématique d'innovation qui s'appuie de manière efficace sur les SI. Il est indéniable que certaines entreprises arrivent à actionner les bons leviers pour relever les défis de l'innovation par les SI, mais les expériences malheureuses montrent des degrés de maturités très divers, et l'on peut s'interroger sur la réelle utilisation de savoirs scientifiques attestés et d'outils d'ingénieurs éprouvés dans les démarches d'innovation.

L'objet de cette table ronde est de s'interroger et de mettre en exergue une réflexion commune entre industriels et chercheurs sur les bonnes pratiques, facteurs de succès et freins, processus ou dispositifs à mettre en œuvre dans une démarche d'innovation par les SI.

Résumé

Ce document contient les actes du trente-quatrième congrès INFORSID (INFormatique des ORganisations et Systèmes d'Information de Décision) qui s'est déroulé à Grenoble du 31 mai au 3 juin 2016. 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). Les membres du CoP ont organisé une méta-évaluation d'un pool d'articles qui leur ont été affectés. La méta-évaluation a consisté à organiser les discussions entre les membres du CP relecteurs de chaque article afin de résoudre les conflits d'évaluation et d'aboutir, dans la mesure du possible, à un consensus. Les membres du CoP ont rédigé, à la fin du cycle de discussions, une brève évaluation de synthèse pour chacun des articles de leur pool d'articles. Les articles acceptés à RCIS et qui l'étaient aussi à INFORSID (les deux évaluations étaient complètement séparées) ont été retenus en une version résumée. Seuls les membres du CoP ont participé à la réunion de sélection finale.