From wikis to tube guitar amplifier simulations, the universal web platform...

Michel Buffa

▶ **To cite this version:**

Michel Buffa. From wikis to tube guitar amplifier simulations, the universal web platform.... Computer Science [cs]. Université Côte d'Azur, 2018. tel-01963301

HAL Id: tel-01963301

https://hal.univ-cotedazur.fr/tel-01963301

Submitted on 21 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Des wikis aux simulations d'amplificateurs de guitare à lampes, le Web plateforme universelle….

Mémoire d'Habilitation à Diriger les Recherches

Soutenue le Mercredi 12 Décembre 2018 par Michel Buffa

Devant un jury composé de :

- Président : Fabien Gandon
- Rapporteur : Alain Mille
- Rapporteur : Pascal Molli
- Rapporteur : Norbert Schnell
- Examinateur : Jean-Marc Fédou
- Examinateur : Peter Sander
- Invité : Yann Orlarey

Université Côte d'Azur

à Sophie et à mes enfants,
à mes parents,

Remerciements

Je tiens particulièrement à remercier :

- Ceux qui m'ont motivé à (enfin) préparer cette HDR : Jacques Misselis, Jean-Marc Fédou,
- Bien sûr mes collègues que j'ai côtoyés dans les diverses équipes de recherche qui m'ont accueilli : Peter Sander, Nhan Le Than, et Fabien Gandon,
- Mes collègues du département informatique et en particulier ceux de la Miage…

Sommaire

1. Introduction 12
2. Naissance du Web Social : les wikis 18
2.1 Du Web aux wikis : une histoire des outils collaboratifs (mes travaux sur la période 1999-2005) 21
2.2 Les prémices des outils collaboratifs 21
2.2.1 Les moyens de collaborer sur le Web 23
2.2.2 WebDav et BSCW, les premiers vrais outils collaboratifs basés sur le Web 24
2.2.3 Les systèmes auteurs 27
2.2.4 Le premier wiki, un outil révolutionnaire 28
2.2.5 On cache HTML 29
2.2.6 On crée un réseau de pages à l'aide de WikiWords 30
2.2.7 Un succès inattendu 31
2.3 L'ère du wiki 32
2.3.1 Des fonctionnalités classiques 32
2.3.2 Un logiciel relationnel ou social d'un nouveau genre 34
2.3.3 La question des droits des utilisateurs 35
2.3.4 Un outil adapté à la structure sociale dynamique 36
2.3.5 Des règles « sociales » pour un fonctionnement idéal 37
2.4 Wikis dans les intranets d'organisations 38
2.4.1 Introduction 38
2.4.2 Intranets : différences avec le Web public (ou bien est-ce la même chose ?) 39
2.4.3 Intranets étudiés 41
2.4.4 Structurer un wiki 56
2.4.5 Conclusion 58
3. Wikis sémantiques (mes travaux sur la période 2006-2010) 60
3.1 Introduction 64
3.2 Définition d'un wiki sémantique 64
3.3 Wikis sémantiques : définition et état de l'art en 2008 65
3.4 SweetWiki : un des premiers wikis sémantiques 70
3.5 Architecture de SweetWiki 71
3.5.1 Ontologies dans SweetWiki 71
3.5.2 Moteur de recherche sémantique et serveur Web 73
3.5.3 L'éditeur 74
3.5.4 Intégration des documents Office 78
3.6 Frameworks Web sémantique et marquage social (« *social tagging* ») 78
3.6.1 Support du marquage social dans le wiki 79

3.6.2 Intégration d'un éditeur d'ontologie dans le wiki 80
3.7 SweetWiki est un wiki d'application 82
3.8 Discussion 85
3.9 Conclusion 88
4. Wikis sémantiques d'application (mes travaux sur la période 2010-2014) 89
4.1 Intégration de la sémantique dans un Wiki programmable, captation d'activité en RDF pour intégration dans un réseau social : le wiki SweetDeki du projet ANR ISICIL 94
4.1.1 Introduction 94
4.1.2 Le framework ISICIL 95
4.1.3 Wikis sémantiques : rappels et état de l'art réactualisé en 2010 97
4.1.4 SweetDeki : le couteau suisse du réseau social ISICIL 98
4.1.5 Modalités d'évaluation 108
4.2 Comment faciliter la programmation et l'apprentissage d'applications pour le Web Sémantique ? WikiNext ! 109
4.2.1 RDF comme structure de données de base, JavaScript comme langage de programmation pour la couche applicative 110
4.2.2 Les principes fondamentaux de WikiNEXT 115
4.2.3 Mashup sémantique et utilisation de la base de connaissance globale de WikiNEXT 120
4.2.4 Usages et évaluation de WikiNEXT 124
4.2.5 Bilan des nouveautés apportées par WikiNEXT 125
4.3 Gestion des droits d'accès via l'ontologie AMO 126
4.3.1 Introduction 126
4.3.2 Positionnement 127
4.3.3 L'ontologie AMO 130
4.3.4 Gestion des droits d'accès dans SweetWiki 134
4.3.5 Bilan sur la gestion des droits d'accès sémantique intégrée au projet ISICIL . 137
4.4 Conclusion 138
5. Réseaux socio-sémantiques (mes travaux sur la période 2006-2012) 141
5.1 Analyse des réseaux sociaux : extension des opérateurs classiques en utilisant des frameworks du Web sémantique 143
5.2 Analyse sémantique des réseaux sociaux (SNA) 147
5.2.1 SemSNA : l'ontologie de l'analyse des réseaux sociaux. 148
5.2.2 Extraire des concepts de SNA avec SPARQL 149
5.3 Lier les données d'interaction en ligne au Web sémantique 154
5.3.1 SemSNI : Extension de SIOC pour modéliser les interactions des réseaux sociaux. 155
5.3.2 Génération de RDF à partir d'une base de données relationnelle avec Corese 157
5.4 Résultats 158
5.5 Conclusion 161

6. Enrichissement sémantique collaboratif et assisté de folksonomies (social tagging, 2008-2011) 163
6.1 Introduction 167
6.2 Un cycle de vie complet pour l'enrichissement sémantique des folksonomies. 169
6.2.1 Contexte 170
6.2.2 Travaux connexes sur l'enrichissement de folksonomies 171
6.2.3 Enrichissement sémantique de folksonomies 174
6.2.4 Extraction automatique de la sémantique émergente 177
6.2.5 Capture et exploitation des contributions individuelles 189
6.3 Conclusion et discussion 196
7. Identification de sous communautés en exploitant le réseau socio-sémantique et la folksonomie (2010-2012) 200
7.1 SemTagP : détection sémantique des communautés dans les folksonomies. 203
7.1.1 Introduction et travaux connexes 204
7.1.2 SemTagP : Propagation d'étiquettes sémantiques 206
7.1.3 Mise en œuvre de SemTagP avec des frameworks du Web sémantique 209
7.1.4 Expérimentations et résultats 213
7.1.5 Améliorations possibles 223
7.2 Conclusion 224
8. Accès aux connaissances sur le Web et robots de télé-présence, 2013-2016 226
8.1 Introduction 232
8.2 Travaux connexes 235
8.3 Architecture logicielle et matérielle de la solution AZKAR 236
8.4 Le Vocabulaire de l'ontologie AMO et le jeu de données RDF décrivant des scènes du Musée de la Grande Guerre 236
8.5 Expérimentations à la Cité des Sciences et de l'industrie 240
8.5.1 Quelques exemples d'outils utilisés pour la création du Dataset 240
8.5.2 Extraits de l'expérience utilisateur lors d'une visite 244
8.6 Conclusion 246
9. Web, Audio et sémantique pour l'interaction musicale (2017-) 248
9.1 Construire une base de connaissance de chansons commerciales populaires 252
9.2 Contexte et état de l'art 256
9.3 Questions et objectifs de recherche 257
9.3.1 Identification et évaluation des sources de données sémantiques et textuelles/culturelles/audio 257
9.3.2 Analyse des paroles de chansons et consolidation avec les données de l'analyse audio 261
9.3.3 Améliorer la recherche d'informations musicales en utilisant à la fois l'audio, les paroles et le Web sémantique. 262
9.3.4 Expérience utilisateur innovante 264

9.4 Premiers résultats 267
9.5 Architecture logicielle 269
9.6 Conclusion 270
10. Participation au standard WebAudio, simulation d'amplificateur à lampes pour guitare et autres effets audio (2015-) 272
10.1 Introduction 277
10.2 État de l'art de la MAO sur le Web 281
10.3 Modélisation des différents étages 285
10.3.1 Préamplificateur 285
10.3.2 Tonestack (partie des réglages basses-médiums-aigus) 293
10.3.3 Réverbération 296
10.3.4 Amplificateur de puissance 297
10.3.5 Simulation des haut-parleurs 301
10.4 Évaluations 302
10.4.1 Latence 302
10.4.2 Réponses en fréquence et formes d'onde 306
10.5 Quelques mots sur l'interface graphique et sur le support MIDI 308
10.6 Tests utilisateurs (*blind tests* et interviews) 311
10.7 Perspectives 314
10.8 Conclusion 314
11. Proposition pour un Standard ouvert pour Plugins WebAudio et applications (2018-) . 316
11.1 Introduction 319
11.2 Contexte 321
11.2.1 Le format WAP comparé aux recommandations de la proposition GMPI... 322
11.2.2 Les WAP ont des URIs ! 323
11.2.3 Convaincre les développeurs de rejoindre le standard WAP 323
11.2.4 Prise en charge de plusieurs environnements d'exécution (Web et applications natives) 325
11.3 Etat actuel de notre proposition 327
11.3.1 Une spécification 327
11.3.2 Outils en ligne, didacticiels et exemples 328
11.3.3 Pedalboard : un hôte plus complexe pour plugins WAP 336
11.4 Perspectives 338
11.5 Conclusion : des plugins audio universels, fonctionnant même dans des logiciels natifs ! 338
12. Conclusion générale 340
13. Bibliographie 342

1. Introduction

J'ai eu la chance d'avoir eu accès, dès l'âge de 12 ans, aux premiers ordinateurs personnels arrivés en France (Tandy TRS-80 en 1977, Apple II en 1980, Commodore Amiga en 1983 etc.), et d'avoir pu étudier très tôt leur architecture et les bases de l'algorithmique et de la programmation, grâce au club Microtel d'Aix-en-Provence. La règle, au sein de ce club, était déjà le partage des connaissances, sans distinction d'âge ou de métier. Des chercheurs passionnés donnaient bénévolement des cours d'Algorithmique, des vendeurs de matériel électronique expliquaient la conception des carte mères etc. Du haut de mes 15 ans, au bout de quelques mois, j'étais responsable des ateliers de programmation en langage Basic !

Au début des années 80, sur mon Apple II équipé d'une carte Modem, j'ai codé et animé un BBS (Bulletin Board System) pour le club Microtel, puis vers 1985 un serveur minitel, qui n'était accessible que pendant deux heures chaque soir pour ne pas monopoliser la ligne téléphonique familiale. C'est là ma première expérience avec une communauté en ligne !

Quelques années plus tard, au début de ma thèse, ce fut la découverte d'Internet, du mail, de Gopher, WAIS et ARCHIE, architecture et protocoles pionniers dans une tentative de structuration de serveurs à travers l'Internet. Puis bien sûr les débuts du Web…

J'ai en effet connu un monde où il était difficile d'imaginer ce que serait le Web, en témoigne un tutorial que j'ai écrit en 1993[1] lors de mon post-doctorat à Carnegie Mellon University, pour présenter ce nouveau média au laboratoire de Robotique. On y voit clairement les questions posées alors « *Le Web survivra-t-il à l'an 2000 ?* » et « *Problèmes sociaux liés au Web ?* ».

Au début de l'année 1993 je suis à l'INRIA et je termine ma thèse en Robotique et Vision par ordinateur. Personne ne connaît le Web. Pourtant, nous allons le voir, il a accompagné ma carrière de chercheur et s'est avéré devenir petit à petit une plateforme universelle et *magique*.

Qui aurait imaginé les applications décrites dans les chapitres de ce manuscrit en 1993 ? Fabriquer de la connaissance et la partager en quelques clics à l'aide de wikis - outils simples et puissants à l'origine du Web social, organiser cette connaissance avec de puissants formalismes cachés (de la sémantique, des ontologies !), construire un réseau social pour la veille technologique mais aussi - plus exotique - contrôler un robot mobile à distance pour visiter des musées ou brancher une guitare électrique dans son ordinateur et jouer du Rock'n'Roll, le son traité en temps réel dans une page Web ? Oui, qui aurait imaginé cela… ?

Je ne vais pas décrire dans ce manuscrit l'ensemble des travaux menés depuis ma thèse (soutenue en 1993). J'ai préféré me focaliser sur trois grands thèmes de recherche qui se sont suivis dans le temps, avec parfois des chevauchements. Ainsi je consacrerai une première partie aux « wikis et wikis sémantiques » (1999-2014), une seconde partie aux "Réseaux socio-

[1] La version traduite en français est encore disponible : http://users.polytech.unice.fr/~buffa/cours/WWW_tutorialFrench/outline.html, je l'ai utilisée lors de mes premières années d'enseignement à l'Université de Nice, en 1994-1996.

sémantiques" (2006-2012) et une troisième partie à "l'accès aux connaissances sur le Web et à la mise en place de nouveaux standards du Web pour l'audio" (2013-aujourd'hui).

Nous verrons en particulier que j'ai fait plusieurs contributions dans le domaine des applications musicales pour le Web, qui s'insèrent dans la dernière partie du manuscrit, car liées à la construction d'une base de connaissance de deux millions de chansons développée dans le cadre du projet ANR WASABI dont je suis coordinateur national.

Mais avant, encore un peu d'histoire pour comprendre mon parcours…

1993 : découverte du Web à l'INRIA, je donne mes premiers tutoriaux à CMU

J'ai passé ma thèse en 1993 mais je me souviens très bien de l'arrivée du Web à l'INRIA, au début de cette année : Frédéric Devernay, alors thésard dans mon équipe, avait installé un des premiers navigateur Web de l'histoire, le "navigateur du NCSA" -la première version du célèbre navigateur Mosaic- et nous avions pu observer, après un temps de chargement assez important, des images de Jupiter prises par une sonde de la NASA quelques jours plus tôt. C'était ma première rencontre avec le Web !

Après une thèse en robotique et vision par ordinateur avec Olivier Faugeras, dans l'équipe Robovis d'INRIA, en Septembre 1993, je pars en post doctorat dans le laboratoire de robotique de Carnegie Mellon University à Pittsburgh. Rapidement, on demande des volontaires pour réaliser le premier site web de l'Université et pour développer une application permettant d'interroger la base de données de la bibliothèque. Avec un autre étudiant, Marc Maimone, nous allons devenir les spécialistes "Web" de CMU et donneront en 1993 et 1994 plusieurs tutoriaux lors de séminaires de partage de connaissances. A l'époque on ne compte que quelques dizaines de site Web dans le monde.

Le Web omniprésent dans mes activités de recherche et d'enseignement

En 1994 je suis recruté comme Maître de Conférences à l'Université de Nice. J'enseigne très vite les technologies Web en Licence et Maîtrise d'informatique, mais également à l'Ecole Supérieure des Sciences Informatiques (ESSI), ancêtre de Polytech Nice. Lorsque le W3C installe son premier bureau dans les locaux de l'INRIA en Avril 1995 et embauche ses premiers stagiaires, ce sont mes élèves. Certains, comme Philippe Le Hegaret sont toujours au W3C et continuent de jouer un rôle important dans le consortium.

Aujourd'hui, je collabore officiellement[2] avec le W3C, l'organisme qui fait ces standards, cependant dès 1995 je connaissais déjà la plupart des personnes travaillant au bureau du W3C à Sophia-Antipolis. Je fais notamment appel à Daniel Dardailler et à Arnaud Lehors du W3C pour intervenir dans mes cours. Nous assistons, avec mes élèves, à des présentations données par Tim Berners Lee.

[2] Depuis 2012 je suis le « Advisory Commitee Representative » (AC Rep) de l'Université Côte d'Azur au W3C. En d'autrres termes, je représente mon Université au sein du W3C. Mais ma collaboration « informelle » date de 1995.

J'étais sûr de participer à un mouvement qui risquait de changer la société en profondeur.

Néanmoins, je n'ai pas fait de recherche scientifique dans le domaine du Web, les premières années qui ont suivi mon recrutement à l'Université de Nice. En 1994 je deviens Maître de Conférences et rejoins l'équipe de recherche Mainline du laboratoire I3S (UMR CNRS), dirigée par Peter Sander, spécialisée dans la vision par ordinateur. C'est naturellement que j'ai publié dans ce domaine pendant trois années, et co-encadré la thèse de Franck Diard sur de la synthèse d'images distribuée.

Cependant, en 1997, avec Peter Sander, nous changeons le nom de l'équipe et l'orientons vers la modélisation de connaissances sur le Web. Nous étions peut-être lassés de notre ancien domaine de recherche, mais sans aucun doute nous étions surtout convaincus de l'énorme potentiel du Web comme vecteur de partage du savoir, et pleinement conscients de sa croissance exponentielle.

La nouvelle équipe s'appelle KEWI (Knowledge Engineering and Web Intelligence) et nous participons successivement à trois projets européens autour du standard OpenMath, entre 1997 et 2003. Ces projets ont pour thème le e-learning via le Web, plus précisément pour des cours de mathématiques.

Ce standard OpenMath, que nous avons contribué à créer est une syntaxe XML qui permet de décrire non seulement la représentation graphique de formules mathématiques, mais également leur sémantique.

En 1997, ce sont les prémisses du Web Sémantique et l'article de référence "*The Semantic Web*" écrit par Tim Berners Lee, James Handler et Ora Lassila, ne paraîtra que quatre ans plus tard dans Scientific American.

Du Web aux wikis, des wikis au "Web social"

Inspiré par les Wikis, les premiers outils collaboratifs ayant transformé un Web majoritairement en lecture seule en un média dans lequel il était facile de publier, je co-encadre avec Jean-Marc Fédou la thèse de Laurent Dirat (1998-2001) sur JOME (Java Open Math Editor), un éditeur interactif de formules mathématiques écrit en Java (JavaScript et l'API du DOM n'existaient pas encore), permettant de manipuler interactivement des objets modélisés à l'aide du standard sémantique OpenMath.

Pour la création et l'édition du texte des pages de cours, nous concevons à l'époque un moteur de wiki très simple permettant d'éditer le contenu des pages. Parallèlement à ces outils, développés dans le cadre d'un travail de recherche, j'utilise également beaucoup les wikis comme outils du quotidien, pour organiser le travail de l'équipe et pour mes enseignements. J'installe les premiers wikis à l'Université de Nice.

Je dirige en même temps la MIAGE de Nice (1997-2002) et avec l'Université de Mannheim nous avons mis en place un projet d'année collaboratif, entre les étudiants du Master Miage de Nice et de Mannheim, outillé par des wikis. Ce projet sera répété pendant plusieurs années. Je publie plusieurs articles à ce sujet à l'époque. Ce travail est décrit dans le premier chapitre de ce manuscrit.

En 2005 je rejoins pour deux ans l'équipe Edelweiss à l'INRIA, dirigée par Rose Dieng, spécialisée en modélisation de connaissances. C'est l'époque où le navigateur Web Internet Explorer 5 propose pour la première fois des implémentations de la technologie Ajax et de l'API du DOM qui permettent de créer des applications beaucoup plus dynamiques, capables de rafraîchir le contenu d'une page Web sans la recharger.

C'est aussi l'apparition des premières implémentations des standards encore très jeunes du web sémantique comme RDF, OWL, SPARQL. Un des premiers moteurs de recherche sémantique, CORESE, est développé dans l'équipe Edelweiss de l'INRIA, qui m'accueille alors. Je m'intéresse aux wikis et l'équipe a pris le train du Web sémantique…

Je vais rapidement mélanger les deux mondes : avec Fabien Gandon je co-conçois et réalise un des premiers wiki sémantique, SweetWiki (2006-2010), exploitant l'état de l'art à la fois dans le domaine du Web (ce wiki est un des premiers outils WYSIWYG du Web) et du Web sémantique (ses propres concepts étaient modélisés par une « ontologie du wiki » et étaient implémentés via le moteur embarqué CORESE, il proposait un éditeur d'ontologies, etc.). Les wikis sont de formidables outils de création de connaissances non structurées et l'apport du Web sémantique permet d'ajouter de la structure, du typage dans les connaissances produites.

Ce wiki sémantique innovant a été utilisé dans plusieurs projets de recherche et par des entreprises pour gérer leur corpus documentaire (section 2.4), il a fait l'objet d'un dépôt à l'Agence de Protection des Programmes (APP).

Parallèlement, le Web devient social (chapitre 3). Les premiers réseaux sociaux comme Facebook ne sont pas encore apparus qu'avec Fabien Gandon, nous initions dans l'équipe le thème du « Web Sémantique et Social », en co-encadrant deux thèses (2009-2012) : une sur « l'analyse des réseaux sociaux », et une sur le phénomène du « tagging social », que nous rassemblons au sein du projet ANR ISICIL (réseau sociaux d'entreprise pour la veille technologique), dont je suis le responsable scientifique dans l'équipe KEWI du laboratoire I3S.

Par la suite je continue ce travail sur les wikis sémantiques en m'orientant vers des « wikis encore plus programmables » appelés « wikis d'application » (chapitre 4).

J'encadre seul la thèse de Pavel Arapov autours de WikiNEXT (2010-2014), un wiki utilisant le langage RDF comme structure de données de base et JavaScript comme langage de programmation pour la couche applicative. WikiNEXT mélange le meilleur des wikis sémantiques et des IDEs en ligne comme codepen.io ou jsbin.com, en permettant la programmation d'applications sémantiques directement dans les pages du wiki.

En 2013 après 8 années de collaboration, je rejoins officiellement l'équipe WIMMICS de l'INRIA, fusion de l'ancienne équipe Edelweiss de l'INRIA et de l'équipe KEWI d'I3S dont je faisais partie. WIMMICS est aujourd'hui dirigée par Fabien Gandon.

Implication dans les standards du Web : « standardiser pour démultiplier le transfert et les résultats de recherche »

En parallèle de ces travaux sur le Web social et les wikis, entre 2009 et 2014 je participe avec le W3C à la création du standard HTML5. Je présente plusieurs tutoriaux à des conférences internationales (WWW notamment, mais aussi des conférences grand public comme Paris Web ou Blend WebMix), j'écris pour le W3C un premier cours en ligne donnant une certification HTML5, sur la plateforme w3devcampus.com. Puis, entre 2015 et 2018, j'écris trois MOOCS hébergés sur la plateforme edX.org du MIT/Harvard, qui depuis 2015 ont réuni plus de 450.000 étudiants.

Un de ces MOOCs est sélectionné pour le prix du meilleur MOOCs de la plateforme en 2016, parmi plus de 1700.

Je participe aussi aux groupes de travail du W3C sur WebAudio, WebMidi et WebRTC (Web Real Time Communication, audio/vidéo/data en mode pair à pair), trois autres standards émergents du W3C.

Accès aux connaissances du Web, pour les humains et pour les programmes, applications à la Robotique et à la Musique

Je m'intéresse aussi à l'accès aux connaissances sur le Web… tout d'abord en modélisant des expositions de musées et en reliant les scènes et objets d'expositions à des ressources du Web de données, puis en créant une base de connaissances musicale de deux millions de chansons, enrichies notamment par des données collectées sur le Web (section 4).

Entre 2014 et 2017 je suis ainsi responsable scientifique du projet AZKAR financé par la BPI (chapitre 8), qui a pour partenaire le laboratoire I3S (WIMMICS), la société Robosoft (fabricant de robots), la société Another World (spin off d'Orange, spécialisée dans la visite à distance de biens culturels), le musée de la grande guerre de Meaux, la Cité des Sciences et de l'Industrie et l'association Approche qui s'occupe d'enfants handicapés. Le projet propose plusieurs scénari mettant en œuvre un robot mobile piloté à distance, notamment pour la visite de musées par des personnes ne pouvant se déplacer (écoles, maisons de retraites, prisons, personnes handicapées ou trop éloignées).

Dans notre proposition nous décidions de placer les technologies Web au cœur du projet, en particulier le Web Sémantique (pour la modélisation des expositions, scènes, objets, parcours, profils de visiteurs et documents reliés) et le standard émergeant WebRTC qui permet les échanges temps réel basés Web en audio, vidéo et data, en mode pair à pair. C'est cette technologie qui sera utilisée pour le contrôle à distance du robot.

Enfin, je participe au projet ANR WASABI (dont je suis à l'origine et aujourd'hui coordinateur national) qui vise à construire une base de données de deux millions de chansons populaires, enrichies par des métadonnées collectées sur le Web, mais aussi issues de l'analyse des paroles en langage naturel et issues de l'analyse audio des chansons.

Des applications clientes utilisant le standard WebAudio du W3C, reliées aux chansons, sont développées. Les cibles sont les compositeurs, écoles de musique, écoles d'ingénieurs du son, diffuseurs de musique… Je contribue à un champ de recherche émergent : "Web Audio" (tout ce qui concerne l'audio avancé sur le Web, notamment la Musique Assistée par Ordinateur), et contribue à ce standard du W3C. Ce travail est décrit dans les chapitres 9, 10 et 11.

J'ai joué un rôle important dans le montage de ces deux projets de recherche. Ils m'ont permis de conjuguer mes passions : le Web et la robotique (j'ai fait une thèse en robotique et vision par ordinateur en 1993), et le Web et la musique (je suis musicien amateur).

Je terminerai par cette citation : *"L'homme passionné par son travail fait ce pour quoi il est fait sans avoir le sentiment de travailler. Il a trouvé sa voie, c'est-à-dire le moyen de son épanouissement personnel. Il fait ce qu'il aime sans se forcer ni se raisonner à aimer ce qu'il fait. Celui qui travaille sans passion accomplit son labeur par obligation, volonté et devoir, alors que le passionné oublie qu'il doit travailler."* (Philippe Laurent)

J'ai oublié que je dois travailler...

2. Naissance du Web Social : les wikis

Contexte :

Entre le début des années 90 (la naissance du Web) et 2008, une profonde révolution s'est opérée, le Web a d'abord vu l'apparition d'outils "collaboratifs" puis il est devenu beaucoup plus "social", notamment grâce aux Blogs et au Wikis. Entre 1997 et 2006 je me suis intéressé aux applications pédagogiques des wikis, puis à leur intégration progressive dans les intranets des entreprises.

Je propose dans ce premier chapitre des rappels historiques sur l'histoire des outils collaboratifs jusqu'au Wikis. Il s'agit d'une synthèse de plusieurs articles écrits à l'époque, notamment pour la revue Interstices de l'INRIA. C'est à partir des expériences décrites que j'en viendrai au champ de recherche qui m'a occupé pendant presque dix années : « les wikis sémantiques et le Web Sémantique et Social », abordés dans les chapitres suivants.

Contributions

J'étais directeur de la Miage de Nice, et faisais partie de l'équipe Mainline (devenue KEWI) du laboratoire I3S qui participa consécutivement à plusieurs projets européens sur le e-learning : OpenMath (ESPRIT-EEU 1996-2000), Trial Solution (IST-EEU 2000-2003) et Open Math Thematic Network (IST-EEU 2000-2002), j'ai contribué à la création du langage standard OpenMath (syntaxe XML, http://www.openmath.org/). J'ai également utilisé des wikis pour organiser des projets d'enseignement innovants (article dans la revue ISDM, 2003 [52]), pour organiser le travail collaboratif dans le cadre des projets de recherche, mais également, nous nous sommes inspirés de l'approche proposée par les wikis pour le développement d'outils livrables de ces projets, tels que l'éditeur de formules mathématique JOME [56, 57], intégré à des pages Web, elles-mêmes éditables dans le navigateur (thèse de Laurent Dirat, conférences ICMT 1999, TICE 2000) [53].

En 2006, quand j'ai rejoint l'équipe Edelweiss de l'INRIA, dirigée par Rose Dieng, j'ai publié deux articles de synthèse : un sur l'utilisation des wikis dans les intranets d'entreprise et un sur l'histoire des outils collaboratifs de la naissance du Web jusqu'à l'apparition et à la popularisation des wikis (revue Interstices, 2008).

Collaborations, encadrement

J'ai utilisé des wikis à cette période à la fois dans des projets d'enseignement innovants impliquant des étudiants en Master Miage à Nice et des étudiants de l'Université de Mannheim (trois années d'affilée, collaboration avec le professeur Wolfgang Effelsberg de Mannheim), puis pendant une dizaine d'années j'ai organisé avec l'Université de Bordeaux le même type de projets (collaboration avec le professeur Jean-Guy Penaud, directeur de la Miage de Bordeaux). Dans le cadre du projet européen Open Math j'ai

encadré à 50% avec Jean-Marc Fédou la thèse de Laurent Dirat ("outils pour des mathématiques interactives et distribuées", 2001).

Publications principales ayant résulté de ces travaux :

Michel Buffa, *« Du Web aux Wikis, histoire des outils collaboratifs », revue Interstices,* Mai 2008. URL.

Michel Buffa "*Intranet Wikis*", workshop "Intrawebs" of WWW 2006, Edinburgh, Scotland.

Michel Buffa, Jean-Claude Grattarola, Peter Sander*, Une nouvelle forme de coopération, nationale et internationale, entre laboratoires de recherche et filières d'enseignement.* Numéro 10 de la revue « Information Science for Decision Making » (ISDM), numéro 10 d'Octobre 2003. URL.

Laurent Dirat, Michel Buffa, Jean-Marc Fédou, Peter Sander, *JOME, un composant logiciel pour le télé-enseignement des mathématiques via le Web, compatible OpenMath et MathML*. (PDF) International Conference TICE 2000, Troyes, France.

Michel Buffa, Laurent Dirat, Jean-Marc Fédou, Peter Sander, *JOME : Java OpenMath Editor*, International Conference ICTMT 4, Plymouth, 9-13 August 1999.

Résumé de cette section

Vous avez dit « wikis » ? Rendus célèbres par l'encyclopédie en ligne Wikipédia, ces outils collaboratifs ont peu à peu gagné le Web. Dans une première partie de ce chapitre, nous dresserons l'historique des outils collaboratifs de l'époque, précédant l'arrivée du Web à nos jours. Ainsi plongés dans le contexte du Web, nous aborderons ensuite les aspects techniques et ergonomiques communs à la grande majorité des wikis, ainsi que les aspects sociaux indissociables de ces outils depuis leur origine.

Les outils collaboratifs répondent aux besoins des utilisateurs de travailler en groupe, par exemple, faciliter l'écriture collaborative de documents. Nés de la croisée de technologies indépendantes (messagerie, forum, workflow...), ils ont permis de mutualiser les ressources. C'est ce qui explique pourquoi ils se sont progressivement généralisés auprès des utilisateurs. Les applications à but collaboratif sont à peu près aussi anciennes que l'Internet et indissociables de l'idée de communauté. Dès 1968, Joseph Carl Robnett Licklider et Robert W. Taylor, deux des fondateurs d'Arpanet — le prédécesseur d'Internet —, pressentent l'apparition des communautés qu'ils définissent alors comme des communautés de partage d'intérêt par opposition aux communautés de proximité géographique classiques.

Nous identifierons les aspects "sociaux" des wikis et montreront comment ils ont joué un rôle qui a influencé très fortement les outils de production de contenu web que nous connaissons aujourd'hui.

Nous illustrerons ensuite l'utilisation de wikis dans deux contextes : (1) des projets collaboratifs pour l'enseignement entre étudiants distants, et (2) l'utilisation progressive de wikis dans les entreprises.

Historiquement, lorsque les organisations professionnelles ont réalisé que les technologies du web pourraient également être utilisée en interne, le développement d'intranets collaboratifs a rapidement pris de l'ampleur. En 2002, plusieurs études montraient que 75% des serveurs Web s'exécutaient derrière des pare-feu d'entreprise.

Pendant des années, le World Wide Web a été de facto un média quasiment *en lecture seule,* un média où tout le monde pouvait naviguer et seulement un très peu pouvaient publier. De même pour les intranets (ou intra-webs) : le syndrome du webmaster a tué dans l'œuf, pendant longtemps, le potentiel de collaboration et de partage.

Lorsque sur le World Wide Web, les choses ont changé avec l'émergence des blogs et des wikis, le Web a commencé à se transformer en un média plus ouvert à la publication. Le navigateur pouvait être utilisé alors pour éditer et créer du contenu comme l'était le tout premier navigateur créé par Tim Berner Lee au début des années 90.

Ainsi, avec un peu de retard, les wikis et les blogs ont commencé à apparaître dans le monde des intranets d'entreprise. Cette partie de l'étude menée à l'époque, que nous résumons dans cette section du manuscrit, se concentre sur l'utilisation de wikis, dans le contexte des intranets, à la fois académiques et dans les organisations d'entreprises.

Naturellement, comme je m'intéressais aux standards du Web, je me suis intéressé au Web sémantique et à la modélisation de connaissances, sous le prisme des wikis, et je créerai avec Fabien Gandon un des premiers wikis sémantiques et lancerai dans l'équipe de recherche Edelweiss de l'INRIA, dirigée par Rose Dieng, le thème de recherche « Web Social » en 2005. Ceci fera l'objet des chapitres suivantes de ce manuscrit.

Ce chapitre de « vulgarisation scientifique » introduit ce manuscrit et a pour but de montrer l'expertise que j'ai pu acquérir avant de démarrer une véritable initiative scientifique sur les « wikis sémantiques » et plus généralement sur le « Web social ».

Êtes-vous prêt pour replonger dans l'histoire des outils collaboratifs ? Allez, on y va !

2.1 Du Web aux wikis : une histoire des outils collaboratifs (mes travaux sur la période 1999-2005)

Figure 1. Cette photo représente ce que voyait Ward Cunningham tous les matins en attendant son bus à l'aéroport international d'Honolulu. C'est de là que lui est venue l'idée de donner le nom de « Wiki » à son logiciel.
(Photo : Arik Baratz / Wikipédia.)

2.2 Les prémices des outils collaboratifs

L'histoire des outils collaboratifs commence avec la « libéralisation » de l'Internet vers la fin des années 80. Cet événement majeur annonce le début d'une nouvelle ère : le réseau n'est plus réservé aux seules universités, et des instances non gouvernementales — l'IAB (*Internet Architecture Board*) et l'ISOC (*Internet Society*) par exemple — sont créées pour mettre sur un pied d'égalité les universités et les entreprises. Cette ouverture permet l'émergence de « fournisseurs d'accès à Internet ». C'est la naissance de l'Internet « commercial ». Le début des années 90 a été marqué quant à lui par la création du Web (l'Internet multimédia) par Tim Berners Lee [167]. Vient ensuite *Mosaic*[3], le premier navigateur graphique créé en 1993 par Marc Andreessen. Capable d'afficher des images et des formulaires de saisie interactifs dans des pages, ce navigateur web est le catalyseur qui a rendu le World Wide Web si populaire. Il est en effet à l'origine de l'augmentation

[3] https://www.wired.com/2010/04/0422mosaic-web-browser/

exponentielle du nombre de serveurs Web. Depuis, d'autres outils sont apparus, mais ce sont ces éléments qui ont fait office de déclencheurs. D'ailleurs, comment marche le Web ?

Tout comme le serveur d'une brasserie sert des cafés aux clients en terrasse, les serveurs web « servent » des pages web à des clients qui lui sont propres : les navigateurs web. Les pages web sont en réalité soit des fichiers texte que le serveur envoie tels quels au client via le réseau, soit des pages « calculées » par programme. C'est le cas lorsque les pages web comportent des formulaires de saisie. Par exemple, une recherche sur Google produit des résultats différents en fonction des mots clés spécifiés. Lorsque la recherche est validée, le client envoie les mots clés au serveur Web et ce dernier interroge une base de données. Les résultats de la requête sont alors mis en forme et renvoyés par le serveur Web vers le navigateur qui peut les afficher : la page de résultats est ainsi « calculée » par Google et envoyée comme s'il s'agissait d'un fichier. On parle de pages « dynamiques » dans la mesure où le processus informatique qui effectue le traitement de la requête est réalisé en temps réel. La technologie permettant de réaliser de telles applications était présente dès le début du Web, avec la Common Gateway Interface (CGI), une norme définissant l'interfaçage d'applications externes avec des serveurs Web.

En 1996, l'arrivée de la technologie ASP de Microsoft a permis de développer plus facilement des applications web interactives, en mixant au sein du code HTML d'une page web des ordres écrits en langage de script, en particulier des requêtes SQL (*Structured Query Language*, le langage de requête des bases de données relationnelles). C'est le serveur Web qui se charge ensuite d'interpréter ces scripts. Cette solution évite d'exécuter un processus externe lors de chaque requête, tout comme le permet la technologie CGI.

L'année suivante, le monde de l'Open Source (centré sur Linux à l'origine) propose le langage PHP, proche du langage ASP. Mais la véritable révolution dans l'histoire des technologies du Web a débuté avec la large diffusion du « LAMP », c'est-à-dire l'association du système d'exploitation Linux, du serveur Web Apache, de la base de données MySql et du langage PHP. Ces outils ont joué un rôle fondamental dans l'essor des applications dynamiques sur le Web en composant une plate-forme commune. À l'époque, on a même parlé du LAMP en tant que « Windows du Web ». En parallèle, Sun Microsystems a proposé dans sa distribution J2EE du langage Java, des interfaces de programmation (APIs), et des outils (serveurs de Servlets) permettant de développer des applications web en Java et de soulager le serveur Web classique du travail d'interprétation des commandes.

Enfin, les années 2000 ont consacré l'utilisation des langages de script, en particulier de JavaScript comme langage permettant de réaliser des interfaces utilisateurs plus réactives, en déportant une partie du traitement dans les navigateurs web.

2.2.1 Les moyens de collaborer sur le Web

Il existe différentes manières de collaborer sur le Web.

Le premier type de collaboration est la discussion, synchrone ou asynchrone. Le courrier électronique ou *mail* en est l'exemple type : il existait bien avant la naissance du Web, à l'époque où l'Internet n'était pas encore multimédia. Les forums sont apparus peu après sous forme de Bulletin Board Systems (les fameux « BBS », ancêtres des serveurs modernes) puis sur le réseau Usenet.

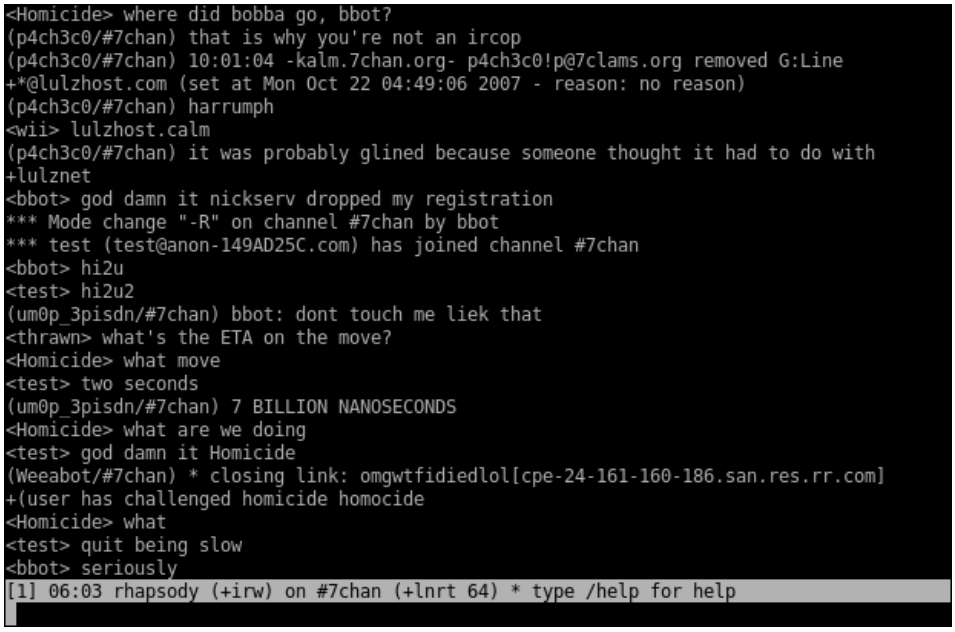


Figure 2. Capture d'écran d'une application IRC.

La messagerie instantanée quant à elle, est arrivée dès 1985 avec IRC (*Internet Relay Chat*, Figure 2), l'ancêtre encore populaire des *MSN*, *ICQ, Hangouts*, *Messenger etc.* d'aujourd'hui. Véritables défis technologiques à l'époque, les outils de *chat* comme IRC ont donné à des milliers de personnes distantes la possibilité de discuter en même temps dans des « salles de discussion virtuelles », les « canaux IRC ».

Ces différents outils permettent de collaborer, d'avoir des discussions synchrones (*chat*) ou asynchrones (forum, *mail*) et d'échanger des messages ou des fils de discussion. L'origine de l'essor communautaire de l'Internet réside dans la propagation des listes de diffusion (*mailing lists*), de Usenet et surtout des BBS. En effet, près de 10 millions d'utilisateurs fréquentaient les BBS en 1992. Ces outils ont joué un rôle très important, car ils ont façonné les modes d'interaction actuels.

Le second moyen de collaboration consiste à échanger des fichiers à l'aide de serveurs partagés. Pour effectuer ce type d'opérations, Ftp (*File Transfert Protocol*) est l'outil le plus populaire de cette époque précédant le Web. À l'aide de clients Ftp, il est possible de se connecter sur un serveur Ftp et de déposer un fichier que d'autres personnes peuvent récupérer, modifier puis redéposer. Ftp est encore très utilisé de nos jours.

Au fil du temps, ces outils se sont améliorés ou ont été remplacés par des outils mieux adaptés. Avec l'apparition du Web, ils sont devenus multimédias et les outils

de *chat* comme *Messenger* ou *Hanghouts* se sont enrichis de nouvelles possibilités en incluant des tableaux blancs (pour le dessin partagé), des fonctionnalités de communication audio ou vidéo, etc. La mise en ligne de pages HTML via des serveurs Web a remplacé en grande partie l'échange de documents textes ASCII auparavant échangés par Ftp ou par *mail*.

Toutefois, de nombreux défauts persistent. Avec le mail par exemple, chaque utilisateur doit trier et organiser ses messages pour pouvoir retrouver les informations pertinentes ou lire une conversation coupée en morceaux. Le problème se répète avec les forums : parfois, il faut sauter des messages pour retrouver le fil d'une discussion, tous les messages n'étant pas dignes d'intérêt. Le partage de fichiers par Ftp pose également de nombreux problèmes : que se passe-t-il lorsqu'on désire travailler à plusieurs sur le même fichier par exemple ? Comment ne pas se « marcher sur les pieds » ? Si on dépose un fichier sur un serveur alors qu'une précédente version du fichier est présente, que faire ? L'écraser ? Renommer le nouveau fichier ? Le problème de la gestion des versions se pose alors. La collaboration n'est donc pas toujours aisée…

Le modèle de publication de documents HTML tel qu'il était encore pratiqué jusqu'en 1997 a rencontré des problèmes du même ordre, les pages web étant la plupart du temps statiques, éditées hors ligne puis déposées sur un espace disque visible par le serveur Web via le protocole Ftp que nous venons de critiquer (pas de gestion de versions, etc.). En outre, la création de pages HTML nécessitait d'installer sur la machine client un éditeur spécialisé tel que le populaire *Dreamweaver* édité par la société Macromedia (devenue Adobe), ou des outils gratuits plus simples comme l'éditeur de pages HTML fourni par le navigateur *Netscape* (ancêtre de Mozilla Firefox) ou encore le logiciel Amaya de l'INRIA.

HTML ne cessant de devenir plus complexe jusqu'à sa version 4.0, les éditeurs HTML ont pris le pas pour devenir de véritables usines à gaz. Ce modèle de publication a néanmoins remarquablement fonctionné faute de mieux, et le nombre de serveurs Web et de pages HTML a augmenté de manière exponentielle.

Mais on ne peut pas encore parler de travail collaboratif, car il s'agit dans la très grande majorité des cas de publications de type 1 – *n* avec un producteur pour *n* consommateurs. Ce modèle est à rapprocher de celui, plus classique, de la publication des journaux et magazines, l'hypertexte et la mise en ligne en sus.

2.2.2 WebDav et BSCW, les premiers vrais outils collaboratifs basés sur le Web

Pour répondre au problème de la gestion de versions sur des serveurs partagés, de nouveaux protocoles comme WebDav (*Web Distributed Authoring and Versioning project*) sont apparus fin 1995. Cette tentative pour étendre le protocole HTTP, permet l'échange transparent de fichiers entre un client et un serveur, à la manière d'un simple système de fichiers. Un utilisateur dépose un fichier dans un répertoire distant de la même manière qu'il déplace un fichier sur son disque local. Le répertoire distant est alors partagé. En outre, WebDav propose un système pour gérer les versions : si on dépose

un fichier alors qu'un fichier du même nom est déjà présent dans le répertoire distant, ce dernier n'est pas effacé et on garde ainsi une trace de toutes les versions existantes.

Des systèmes comme BSCW (*Basic Support for Cooperative Work*) proposant des fonctionnalités bien plus évoluées, ont succédé à WebDav. Leur plus ? Fournir des interfaces utilisateurs sous la forme de pages web ainsi que la gestion des utilisateurs ou des groupes d'utilisateurs et la possibilité d'être prévenu (par une notification) lors de modifications sur certaines parties de l'espace partagé, etc. La Figure 3 présente un écran typique de BSCW tel qu'il apparaissait sur les navigateurs Web de notre ordinateur en 1997.

Figure 3. Écran typique de BSCW. L'icône NEW (2) indique que le fichier est apparu dans l'espace partagé depuis la dernière connexion.

L'apparition de WebDav et de BSCW marque une étape importante. On commence à disposer d'outils de manipulation de fichiers, de partage de documents (y compris de documents hypertextes) incluant la gestion de versions et des mécanismes de notification évolués. Les interfaces utilisateurs sous forme de pages web permettent de classer BSCW parmi les pionniers des outils collaboratifs « basés sur le Web ». Cependant, des éditeurs externes — éditeurs HTML, éditeurs de type *Microsoft Word* ou simple éditeur de texte, etc. — sont toujours nécessaires pour créer les documents. En effet, à cette époque, les limitations des technologies web ne permettent pas encore de construire des interfaces utilisateurs riches et réactives (l'API du DOM permettant cela n'apparaitra qu'en 2005 avec Internet Explorer de Microsoft).

Mais pourquoi éditer des documents dans son navigateur web alors que des outils puissants et reconnus existent ? Dans le monde des entreprises, à la même époque, des outils collaboratifs tels que Lotus Notes d'IBM connaissent un succès important en proposant des services — le plus souvent articulés autour de formats propriétaires — tels que le *mail*, l'échange de fichiers, le partage d'espace disque, le partage d'annuaires, etc. Ces outils proposent donc des interfaces utilisateurs très riches (peut-être même trop) et des fonctionnalités à profusion. Dédiés à des applications intranet, c'est-à-dire limitées au réseau interne de l'entreprise, ceux-ci offrent alors une alternative « industrielle et sérieuse » aux outils du Web, souvent jugés trop simples et peu sécurisés.

Un outil tel que le *mail* « standard » basé sur le protocole SMTP a mis dix ans avant d'être utilisé par les entreprises. Pourquoi ? Simplement parce que l'idée qu'un employé puisse envoyer des messages à l'extérieur de l'entreprise était difficile à accepter.

L'accès au Web également a longtemps été restreint ! Interdire l'accès à *Google* aujourd'hui, cela semble inimaginable dans le monde de l'entreprise et pourtant, il n'est pas rare au début des années 2000 de rencontrer des sociétés où l'accès au Web externe est encore sous contrôle.

Se tourner vers *Lotus Notes* ou vers d'autres outils équivalents semblait à l'époque plus judicieux, parce que ces logiciels lourds correspondaient à cette ancienne « culture de l'entreprise » où tout doit être régulé. Au cœur de *Lotus Notes* se trouve en effet un mécanisme très complet de gestion des droits et privilèges, incluant de puissants et complexes algorithmes qui permettent de dire ce que chacun a le droit de faire ou de ne pas faire. Il est possible par exemple de donner à certains utilisateurs des pouvoirs de délégation *(« tu as le droit de donner des accès à cet espace, mais uniquement aux personnes du groupe des managers que tu n'as pas le droit de modifier »*).

Pourtant, ces outils sont chers, lourds, et finalement peu collaboratifs.

Les premiers outils collaboratifs « basés sur le Web », proposent une autre vision de la collaboration :

- Aucune installation n'est nécessaire, un simple navigateur web suffit. Ceci est particulièrement adapté aux situations où les utilisateurs du système sont mobiles : intranet d'universités ou d'entreprises réparties sur plusieurs sites, télétravail depuis le domicile, etc.
- Gestion des versions : les documents que l'on voit sont régulièrement mis à jour et l'accès aux versions précédentes est possible, rien ne peut être perdu.
- Systèmes basés sur un langage hypertexte, permettant la création d'un corpus documentaire formé de pages reliées entre elles par des liens hypertextes.

Par ailleurs, le langage HTML s'est imposé naturellement pour la réalisation des documents hypertextes, malgré sa complexité qui le rendait difficilement utilisable par des non informaticiens. Le temps où l'on tapait du HTML dans un éditeur de texte pour composer une page web est révolu ! Le navigateur *Netscape* a ouvert la voie en proposant un éditeur HTML « noyé » dans son navigateur, mais il ne s'agissait pas réellement d'un

éditeur intégré dans une page web. Pour éditer une page HTML, il fallait toujours *Netscape*, un autre navigateur ne pouvant faire l'affaire !

Puis sont arrivés sur le marché *les systèmes auteurs* : les premiers éditeurs hypertextes embarqués dans des pages web qui ne nécessitaient pas une grande expertise en HTML.

2.2.3 Les systèmes auteurs

Prenons des étudiants sur un campus universitaire… Ils sont susceptibles d'utiliser n'importe quel ordinateur parmi ceux mis à leur disposition dans les salles informatiques. Ils ne vont pas travailler tout le temps dans la même salle ni sur la même machine. S'assurer que tous ces ordinateurs disposent des logiciels nécessaires mis à jour oblige les administrateurs systèmes à un lourd travail de maintenance. Les personnes doivent être formées à ces logiciels ainsi qu'à l'utilisation des outils de publication (Ftp ou autre). L'idée du « zéro-installation » a toujours été séduisante : nul besoin d'installer des logiciels, un simple navigateur web suffit ! Embarquons donc les applications dans le navigateur !

C'est cette motivation première qui est à l'origine de l'apparition des premiers systèmes auteurs en 1994. Ces derniers permettent de créer à l'aide de formulaires HTML du contenu destiné à être affiché dans des pages web. Par exemple, les plates-formes de télé-enseignement permettent dès lors aux enseignants de remplir des formulaires pour indiquer le descriptif des cours qu'ils comptent mettre en ligne. Ils peuvent y insérer des mots-clés caractérisant le cours, le type de documents composant le cours, etc. Parfois, on peut même entrer directement du code HTML dans une zone de saisie et ainsi mettre du texte en gras ou en italique, ajouter des liens hypertextes… Certains éditeurs vont même plus loin en proposant de l'édition HTML « interactive », comme l'outil *WebWriter* apparu en 1996.

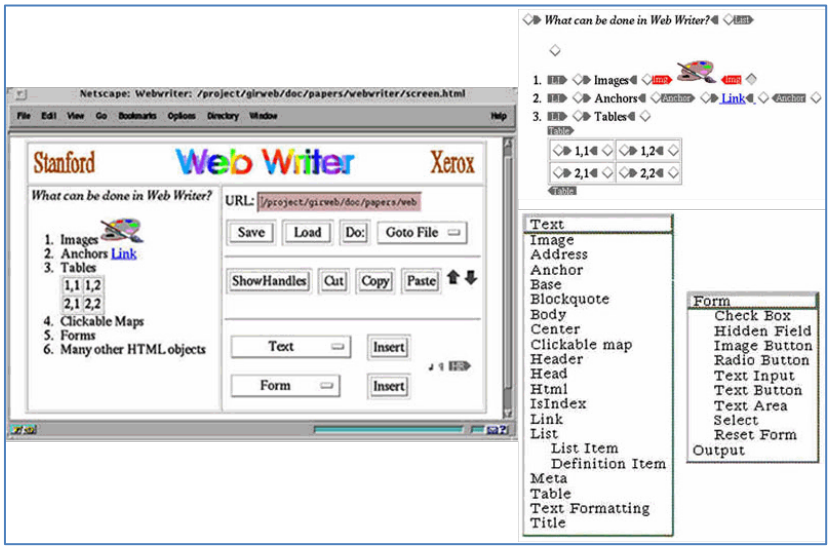


Figure 4. L'interface graphique en ligne de l'outil WebWriter (1996)

La Figure 4 donnent une idée de la difficulté d'utilisation d'un de ces éditeurs de page web disponible en 1996, *WebWriter* en l'occurrence.

À gauche, création d'une page de « patrons de présentation » (*template*) pour *WebWriter* (1996). Sur cet écran, on voit bien l'interface de création de *templates* qui permet de spécifier quelles parties du texte sont éditables. L'image en haut à droite illustre quant à elle le mode d'édition à proprement parler : le texte est entouré de points d'ancrages, et lorsqu'on clique avec le bouton droit sur un de ces points d'ancrage, un menu déroulant (avec deux niveaux de hiérarchie) propose d'insérer des balises HTML simplifiées (un sous-ensemble de HTML version 3.2 étant proposé). Ce menu est illustré sur l'image en bas à droite. Imaginez le travail fastidieux qui attendait celui qui voulait réaliser une page un peu longue !

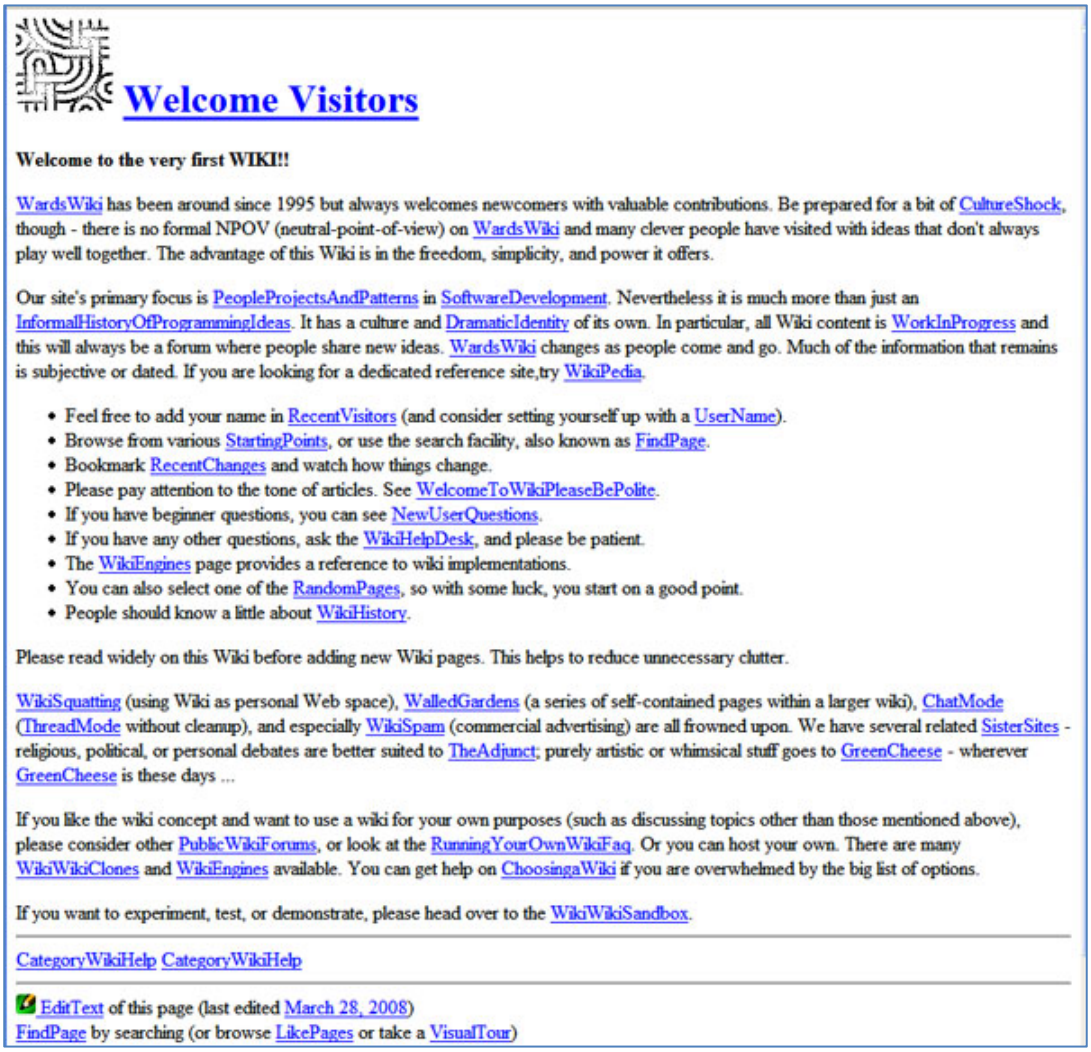
Ces outils ont peu marqué l'histoire du Web, leur insuccès s'expliquant par leur grande complexité. En effet, il semble que leurs auteurs se soient surtout enthousiasmés pour les aspects technologiques, au détriment de leur ergonomie.

En dépit des technologies web trop limitées ne permettant pas la création d'interfaces utilisateurs riches, d'autres outils mieux adaptés à ces limitations ont vu le jour et les gens ont enfin pu collaborer. Du reste, la technologie n'est qu'une partie du problème. Dans la vie réelle, nous sommes parfois amenés à collaborer sans l'aide d'un ordinateur et pour ce faire, nous griffonnons des dessins sur des feuilles de papier, sur des tableaux, nous utilisons des interfaces parfois difficilement compatibles (langages différents, cultures différentes…). De même, avec un petit effort de mémoire, il est amusant de se rappeler tout ce que l'on arrivait à faire avec les outils informatiques d'il y a 15 ans. À l'époque, malgré les défauts des outils disponibles (inhérents aux contraintes technologiques notamment), et même si cela demandait de gros efforts, les gens réussissaient déjà à travailler ensemble. Parmi les outils collaboratifs les mieux adaptés aux limitations du Web de l'époque, certains ont eu un impact plus grand que d'autres : les wikis !

2.2.4 Le premier wiki, un outil révolutionnaire

Une révolution démarre en 1995 lorsque l'informaticien Ward Cunningham crée le premier wiki, celui du Portland Pattern Repository. Toujours en activité, le site est désormais entré dans l'histoire de l'Internet et une pratique populaire consiste à l'appeler « le wiki de Ward » (*Ward's Wiki*).

Lassé par le rôle central du webmestre et la complexité du processus de création de documents HTML, influencé par Hypercard et certainement par la vision initiale du Web qu'avait eue Tim Berners Lee, Ward Cunningham invente à l'occasion de la création du portail Web du Portland Pattern Repository, le concept génial du wiki : un site web permettant la création, l'édition et la mise en place d'un réseau de pages à la volée, depuis un simple navigateur web, de manière très simple et rapide (« wiki » signifiant « vite » en hawaïen).



Welcome Visitors

Welcome to the very first WIKI!!

WardsWiki has been around since 1995 but always welcomes newcomers with valuable contributions. Be prepared for a bit of CultureShock, though - there is no formal NPOV (neutral-point-of-view) on WardsWiki and many clever people have visited with ideas that don't always play well together. The advantage of this Wiki is in the freedom, simplicity, and power it offers.

Our site's primary focus is PeopleProjectsAndPatterns in SoftwareDevelopment. Nevertheless it is much more than just an InformalHistoryOfProgrammingIdeas. It has a culture and DramaticIdentity of its own. In particular, all Wiki content is WorkInProgress and this will always be a forum where people share new ideas. WardsWiki changes as people come and go. Much of the information that remains is subjective or dated. If you are looking for a dedicated reference site,try WikiPedia.

- Feel free to add your name in RecentVisitors (and consider setting yourself up with a UserName).
- Browse from various StartingPoints, or use the search facility, also known as FindPage.
- Bookmark RecentChanges and watch how things change.
- Please pay attention to the tone of articles. See WelcomeToWikiPleaseBePolite.
- If you have beginner questions, you can see NewUserQuestions.
- If you have any other questions, ask the WikiHelpDesk, and please be patient.
- The WikiEngines page provides a reference to wiki implementations.
- You can also select one of the RandomPages, so with some luck, you start on a good point.
- People should know a little about WikiHistory.

Please read widely on this Wiki before adding new Wiki pages. This helps to reduce unnecessary clutter.

WikiSquatting (using Wiki as personal Web space), WalledGardens (a series of self-contained pages within a larger wiki), ChatMode (ThreadMode without cleanup), and especially WikiSpam (commercial advertising) are all frowned upon. We have several related SisterSites - religious, political, or personal debates are better suited to TheAdjunct; purely artistic or whimsical stuff goes to GreenCheese - wherever GreenCheese is these days ...

If you like the wiki concept and want to use a wiki for your own purposes (such as discussing topics other than those mentioned above), please consider other PublicWikiForums, or look at the RunningYourOwnWikiFaq. Or you can host your own. There are many WikiWikiClones and WikiEngines available. You can get help on ChoosingaWiki if you are overwhelmed by the big list of options.

If you want to experiment, test, or demonstrate, please head over to the WikiWikiSandbox.

CategoryWikiHelp CategoryWikiHelp

EditText of this page (last edited March 28, 2008)
FindPage by searching (or browse LikePages or take a VisualTour)

Figure 5. Page d'accueil du wiki de Cunningham telle qu'elle apparaît encore en 2008 : simple, austère, orientée clairement vers le contenu. Les premiers paragraphes donnent quelques clés sur le concept du wiki. Remarquons en particulier la syntaxe des liens : ce sont tous des WikiWords ! La page « brille » par ses WikiWords, emblématiques des wikis.

2.2.5 On cache HTML

Cunningham n'a fait que reprendre l'idée des systèmes auteurs, mais en la simplifiant. Chaque page du site web du Portland Pattern Repository comprend un lien *edit* permettant d'éditer le contenu de la page dans une zone de saisie d'un formulaire HTML (*TextArea* HTML), à la manière des systèmes auteurs.

Première révolution, on oublie le HTML ! L'informaticien choisit un langage d'édition de pages très simple, communément appelé un « dialecte WikiML » (*Wiki Markup Language*). Ce langage ressemble fortement au langage utilisé par les internautes pour formater les messages ASCII envoyés par *mail* ou échangés sur les forums Usenet.

À l'époque de Usenet, Internet n'est pas encore graphique et pour mettre un mot en évidence, on l'entoure d'astérisques `*comme ceci*` et pour simuler l'écriture italique, on l'entoure par des caractères soulignés, `_comme ceci_`. Le langage de ce premier

wiki, demeuré pratiquement inchangé jusqu'à aujourd'hui, propose des règles de formatage simples, très proches de la toute première version de HTML : gras, italique, souligné, plusieurs niveaux de titres, listes à puces, listes numérotées…

2.2.6 On crée un réseau de pages à l'aide de WikiWords

La véritable contribution de Ward Cunningham, c'est l'invention des WikiWords (appelés aussi « CamelWords » en anglais ou « ChatsMots » en français). La « casse ChatMot » est un format de syntaxe pour les phrases, dans lesquelles tous les espaces sont supprimés, les mots accolés et les premières lettres mises en capitales `UnPeuCommeCeci`. Ce nom animalier provient de l'assemblage des mots « Chat » et « Mot », formant ainsi deux bosses là où les lettres en capitales nous rappellent les deux bosses du chameau.

Le ChatMot est très usité pour les PagesNoms de wiki, parce qu'il permet de créer des liens automatiques vers ces pages. Pour créer de nouvelles pages web et les relier entre elles, il suffit en effet de taper dans une page deux mots avec l'initiale en capitales collés l'un à l'autre et de sauvegarder la page. Dans la page, le ChatMot est devenu un lien qui se termine par un « ? ». Il suffit alors de cliquer sur le lien pour créer une nouvelle page dont le nom est le ChatMot, et dont l'URL se termine par ce même ChatMot. La page nouvellement créée est automatiquement affichée en mode édition, il suffit de taper du texte et de la sauvegarder. Ainsi, de manière très simple, on construit un réseau de pages.

Voici un exemple simple. Si on tape dans une page du wiki : « Cette phrase contient un LienVersUneAutrePage » et que l'on sauvegarde la page, cela donne :

« Cette phrase contient un `LienVersUneAutrePage ?` »

Le ChatMot « `LienVersUneAutrePage ?` » devient un lien cliquable (le point d'interrogation indiquant qu'il pointe vers une page qui n'existe pas encore). Si on clique dessus, le wiki crée automatiquement une page vide reliée à la précédente (par le lien hypertexte `LienVersUneAutrePage`) et le lien perd son point d'interrogation à la fin, indiquant que la page existe maintenant. Les wikis modernes proposent un bouton *link* et autorisent la création de liens sans utiliser cette astuce, mais à l'époque, les éditeurs WYSIWYG pouvant fonctionner dans une page web n'existaient pas, et les ChatMots étaient très utilisés.

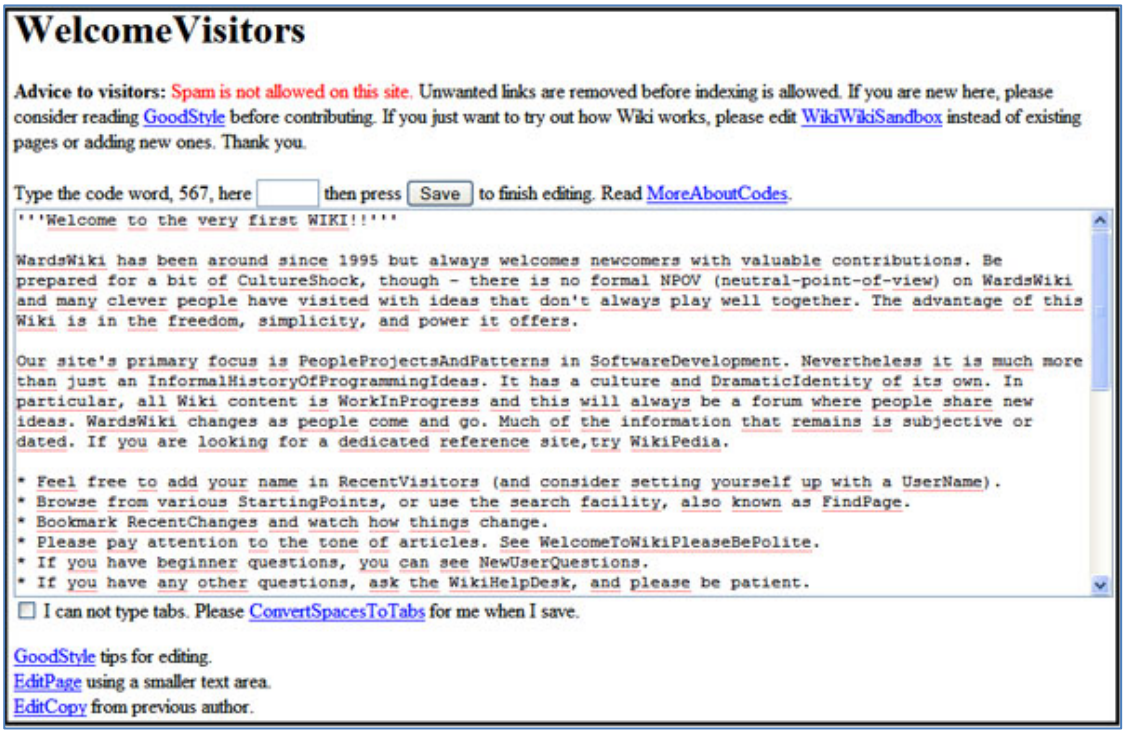


Figure 6. Version WikiML de la page d'accueil du wiki de Ward Cunningham en 2008.

Comment Ward Cunningham a-t-il eu l'idée de la syntaxe ChatMot ? La raison historique à cela provient du wiki original, le WikiWiki, qui était initialement le Wiki du Portland Pattern Repository.

N'oublions pas que Ward Cunningham appartenait à une communauté d'ingénieurs et de scientifiques qui s'intéressaient aux modèles de conception collaborative (*design patterns*) et aux langages de programmation. Le portail Web du Portland Pattern Repository qu'il a implémenté à l'aide de son moteur de wikis, était en effet destiné à publier et à discuter tous ces aspects. Il est naturel qu'il ait été hautement influencé par le langage de programmation orienté objet SmallTalk (dont le nom est lui-même un ChatMot). Dans ce langage, les noms de classes doivent également être en syntaxe ChatMot. Bien qu'aujourd'hui il soit possible de créer des liens dans des wikis de manière différente, l'usage de ChatsMots est une pratique toujours très populaire. C'est une fonctionnalité caractéristique des wikis, qui n'a jamais été vue ailleurs sur le Web.

2.2.7 Un succès inattendu

C2wiki, le wiki originel de Ward Cunningham, rencontra un tel succès qu'il est vite devenu le premier grand wiki communautaire, ses utilisateurs débordant rapidement du cercle des scientifiques et des passionnés de schémas de conception et de programmation orientés objet pour devenir des fidèles de la WikiWay, nom donné à la « CultureWiki » !

Et devinez le sujet de discussion favori des utilisateurs de ce wiki originel sur le site ? Les wikis et la WikiWay ! Le concept connut un tel triomphe, un tel impact, qu'il a réellement

contribué à modifier la vision de l'Internet moderne, le transformant en un média sur lequel on peut écrire et collaborer facilement.

Les plus curieux pourront lire le livre « *The WikiWay, quick collaboration on the web* » [60], ainsi que l'interview de W.Cunningham (en anglais) en quatre parties que Bill Venners a publié en 2003, dans laquelle Ward Cunningham revient sur la genèse du premier wiki.

2.3 L'ère du wiki

Définir ce qu'est un wiki aujourd'hui n'est pas si simple. Le concept initial a en effet été dérivé et implémenté dans plusieurs centaines de moteurs de wikis. Et d'autres types de logiciels collaboratifs se sont inspirés des wikis.

Certains « puristes », à l'époque, vous auraient dit que pour être un wiki, l'édition des documents doit être effectuée à l'aide d'un langage de type WikiML, ce qui exclurait automatiquement les outils équipés d'un éditeur WYSIWYG. Dans le cas présent, nous nous en tiendrons à la définition de Brian Lamb, la plupart des moteurs de wikis modernes se dotant aujourd'hui d'un éditeur WYSIWYG.

La définition communément admise est donc celle qui suit :

« Un wiki est un site web collectif dans lequel un grand nombre de participants sont autorisés à modifier les pages et à en créer de nouvelles à l'aide de leur navigateur web. »

Le moteur de wikis est quant à lui l'outil logiciel qui permet de maintenir le site.

Il est courant d'utiliser le mot wiki pour décrire indifféremment le moteur de wikis ou le site web créé à l'aide d'un moteur de wikis, ce qui prête parfois à confusion.

De très nombreuses variantes des moteurs de wikis existent aujourd'hui. Brian Lamb, dans son article très intéressant « *Wikis, Ready or Not* », insiste sur le fait que l'on retrouve des points communs à toutes ces variantes — création de pages et de liens à la volée —, mais aussi beaucoup de différences. Faute de standardisation, le dialecte WikiML n'est jamais le même d'une implémentation à l'autre bien qu'un standard, le WikiCreole, soit récemment apparu. En outre, les fonctionnalités varient en fonction de la cible visée par le logiciel wiki : les wikis communautaires tels UseMod ou MediaWiki (le logiciel utilisé par la célèbre encyclopédie en ligne Wikipédia), incluent par exemple des systèmes anti-spam, et certains wikis ciblés pour les intranets d'entreprise comme TWiki proposent de nombreuses extensions permettant d'utiliser le wiki comme une « plate-forme d'applications » capable de se connecter à des services externes du type LDAP, etc.

2.3.1 Des fonctionnalités classiques

On trouve des wikis écrits dans tous les langages : Perl, Java, C#, PHP, C, C++, Python, etc. Certains — TWiki, MoinMoin, JotSpot, XWiki — sont dédiés à des applications de type intranet, d'autres — c'est le cas de UseMod et de C2wiki — sont destinés à la

création de sites web collaboratifs publics. Le très populaire moteur de wikis, MediaWiki, se destine quant à lui à des sites web articulés autour d'une taxonomie.

Peu à peu, avec l'apparition du Web 2.0 perçu comme le renouveau du World Wide Web et le développement de nouvelles technologies (Ajax, etc.), les interfaces graphiques se sont modernisées (éditeurs WYSIWYG). Par ailleurs, une nouvelle branche, les « wikis sémantiques », est apparue avec le développement des technologies du Web sémantique.

Tous ces wikis utilisent les fonctionnalités principales du wiki de Ward Cunningham :

- L'édition des pages à la volée est possible depuis le navigateur en un seul clic. Le bouton *edit* est bien en vue pour inviter les utilisateurs à modifier le document présenté.
- L'édition permet un formatage limité. Elle est ainsi simplifiée et les utilisateurs peuvent produire plus facilement du contenu, sans se sentir tenus de passer du temps sur la présentation. La plupart des wikis supportent un langage de type WikiML ou bien proposent un éditeur WYSIWYG proche des fonctionnalités de formatage de texte de l'outil *WordPad* de *Microsoft*.
- La création des pages à la volée est également possible en tapant un nouveau WikiWord. Une fois la page sauvegardée, il suffit de cliquer sur le WikiWord pour créer la page du même nom.
- L'édition de la structure du site à la volée est induite par les deux fonctionnalités précédentes. On peut copier le contenu d'une page pour le répartir dans deux pages reliées à partir du navigateur web.
- Une section « bac à sable » où tout un chacun peut s'entraîner à modifier et à créer des pages afin de se familiariser avec l'outil ou avec le langage WikiML en général est mise en place dès l'installation du site.
- Le gestionnaire de version agit comme un « filet de sécurité » : ainsi, il est rassurant pour la communauté des utilisateurs de savoir que rien ne sera perdu sur le site, quoi qu'il arrive.
- Une page contenant des liens vers les pages modifiées récemment (*Recent Changes*) est disponible.
- Un moteur de recherche, le plus souvent basé sur des recherches par mot clé « à la *Google* », ou bien parfois basé sur des *tags* entrés par les utilisateurs (*social tagging*) est intégré.

Cet ensemble de fonctionnalités « de base » fait des wikis des outils à part. Simples et intuitifs, ils invitent les utilisateurs à devenir des auteurs. Mais ces caractéristiques techniques n'expliquent pas tout !

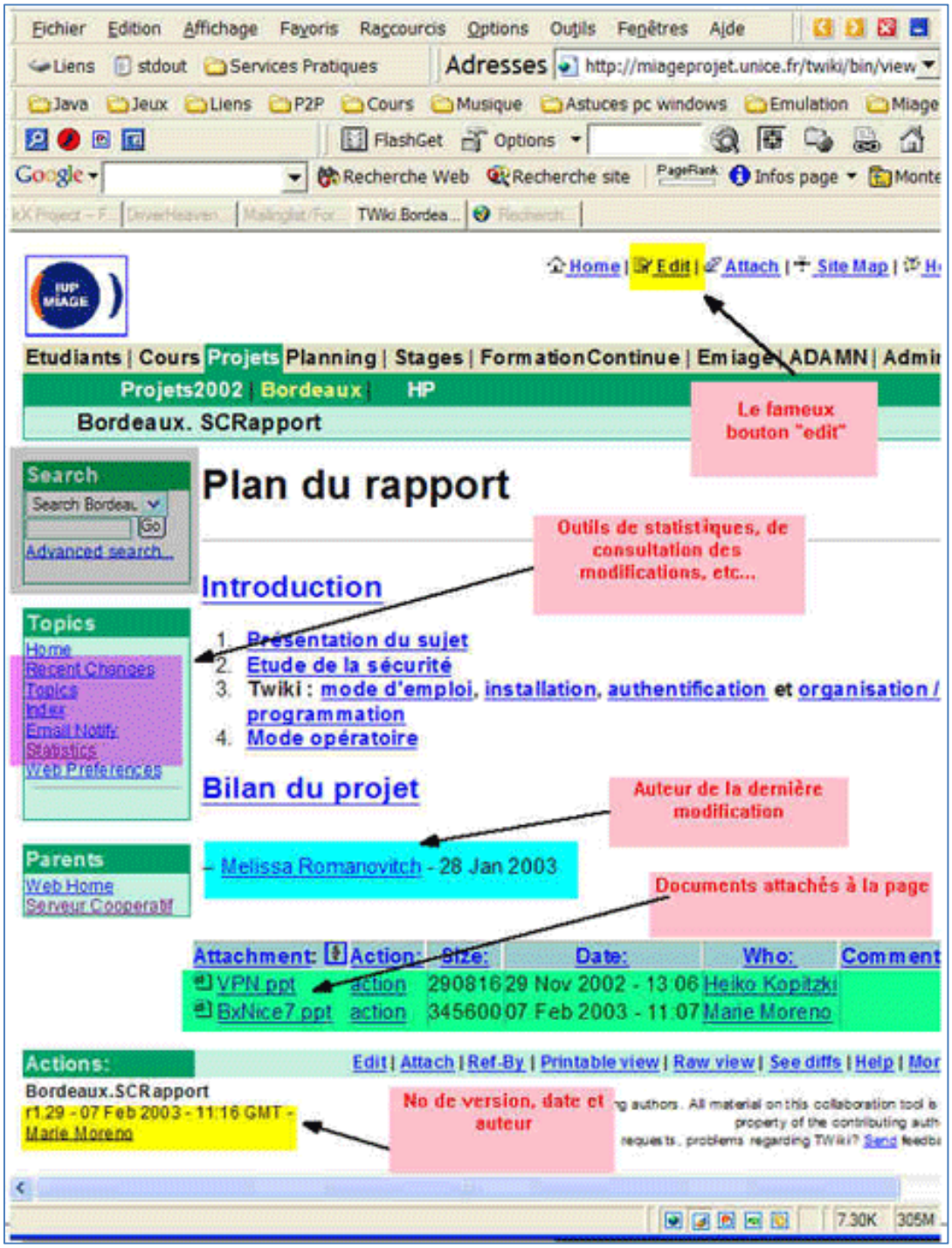


Figure 7. Le serveur wiki de l'IUP Miage de l'Université de Nice en 2001. Le logiciel wiki utilisé ici est TWiki, un des plus populaires à l'époque. Les principales caractéristiques mises en valeur dans cette image montrent l'intérêt d'un tel outil pour la collaboration : suivi des modifications, pages facilement éditables, notification, statistiques d'utilisation, etc.

2.3.2 Un logiciel relationnel ou social d'un nouveau genre

Pourquoi dit-on des wikis que ce sont des logiciels sociaux uniques en leur genre ?

Comme nous l'avons vu précédemment, l'histoire des wikis a commencé avec le wiki originel de Ward Cunningham, C2.com également appelé C2wiki. Ce premier wiki a

connu une évolution très rapide, le nombre de pages et d'utilisateurs croissant de manière exponentielle. On a pu observer le même phénomène avec la célèbre encyclopédie en ligne Wikipédia ou avec d'autres wikis communautaires.

Comment expliquer ces succès ? Tout d'abord, historiquement, personne ne s'attendait à une réaction aussi enthousiaste de la part des internautes. L'explication est autant d'ordre social que technique : participer, oui, mais pourvu qu'on me donne les moyens de collaborer et de communiquer simplement ! Il fallait donc non seulement des moyens techniques, mais aussi des moyens humains.

Figure 8. Page d'accueil de Wikia, un annuaire de wikis.

2.3.3 La question des droits des utilisateurs

Outre la simplicité d'utilisation du wiki de Ward Cunningham, il est important de souligner que les utilisateurs n'étaient pas authentifiés : pas de *login* ni de mot de passe ! S'ouvre alors pour n'importe qui, la possibilité — révolutionnaire à l'époque — de créer et de modifier les pages d'un site web public, de changer la structure du site sans en demander la permission et de manière totalement anonyme ! Encore aujourd'hui, les trois principaux wikis communautaires Wikipédia, le wiki de Ward Cunningham et Meatball, qui a remplacé le wiki de Ward Cunningham comme site où on parle de la WikiWay, n'acceptent que des contributions anonymes.

Ward Cunningham a également eu une attitude assez remarquable au cours de l'évolution de son wiki. Il a en effet laissé les utilisateurs libres de se l'approprier. Que la discussion dévie et ne s'adresse plus à la communauté orientée objet ? Ce n'est pas grave ! Que des utilisateurs publient des critiques corrosives ou que des polémiques s'installent ? On

laisse faire ! Il ne s'est pas senti le droit moral de censurer ses utilisateurs. Après tout, si le wiki a été un tel phénomène, c'est bien parce qu'il était basé sur le respect d'autrui et sur l'idée que la parole de l'un vaut bien celle de l'autre.

Mais attention, bien que Ward Cunningham n'ait pas censuré ses utilisateurs, il n'a jamais cessé d'être présent en contribuant aux nombreuses discussions qui sont apparues sur les pages de son wiki. Il n'a cessé de *« jardiner son wiki* ». D'autres utilisateurs passionnés se sont transformés en « jardiniers », si bien qu'au bout de quelque temps, le wiki s'est en quelque sorte *auto-modéré*.

Lors de la conférence internationale sur les wikis, WikiSym 2005, Ward Cunningham a insisté sur les aspects sociaux : « *A wiki is a work sustained by a community* », « *One's words are a gift to the community…* », « *to collaborate on a work, one must trust…* », etc.

Il a toujours encouragé ses utilisateurs à ne pas signer leurs contributions, afin de limiter les problèmes d'ego. Il considère les documents du wiki comme de l'argile que chacun peut modeler à sa guise pour former une œuvre commune. Les pages étant créées par un auteur initial puis complétées, modifiées, remodelées par d'autres personnes, devenant « co-auteurs » à leur tour, il est de fait souvent impossible d'attribuer la paternité d'une page à une personne dans les wikis anonymes comme celui de Ward Cunningham ou Wikipédia.

Ward Cunningham est convaincu que la nature humaine pousse à la collaboration – l'homme étant un animal social – et qu'en donnant ne serait-ce qu'une petite partie d'eux-mêmes, les utilisateurs d'un wiki produiront forcément quelque chose de bien. Selon lui, c'est cela qui pousse les utilisateurs à contribuer au travail de réflexion des autres. À ses yeux, c'est la preuve que des choses merveilleuses arrivent lorsqu'on se met à faire confiance aux gens plus que de raison.

2.3.4 Un outil adapté à la structure sociale dynamique

Si l'on regarde de plus près les grandes expériences de wikis communautaires apparus entre 1998 et 2008, (depuis le wiki initial) — comme l'encyclopédie en ligne Wikipédia ou le site communautaire Meatball —, la métaphore des documents vus comme de l'argile modelable par tout un chacun semble en effet fonctionner. Les contributeurs des wikis jouent le jeu et acceptent sans rechigner le principe de voir leurs idées remodelées par d'autres. Ils parviennent aussi à utiliser les idées des autres, à les mélanger, sans en perdre la substance.

Il faut accepter que les choses évoluent, qu'une page web soit un objet vraiment dynamique dans le temps. Ceci est rendu possible par le concept de l'hypertexte : si une page devient trop longue, c'est qu'elle renferme trop d'idées, donc coupons-la en morceaux (en pages wikis), donnons des noms « parlants » à ces idées (les ChatsMots) et relions-les ensemble. La manière avec laquelle les informations d'un wiki sont maillées, « brille » par ses ChatsMots.

Ici, il est pertinent de faire remarquer que les ChatsMots sont *signifiants*, bien plus que des URLs peu lisibles. En effet, comparons par exemple l'URL :

« `http://www.canalplus.com/prog.asp?day=today` »

…avec sa traduction en ChatMot qui donnerait quelque chose comme `ProgrammeCanalPlusDuJour`.

Les utilisateurs des premiers wikis communautaires devaient faire l'effort sémantique de bien choisir les mots pour décrire les documents qu'ils désiraient créer, ce qui rendait les liens dans les pages « parlants », tout comme un programmeur doit « bien » choisir les noms des classes.

En jargon informatique, ce remodelage facile s'appelle la refactorisation (*refactoring*). Devenu incontournable aujourd'hui dans le monde du développement logiciel, le *refactoring* a été rendu populaire par les méthodes de Programmation Extrême (eXtreme Programming, ou XP en anglais) et de Programmation Agile.

Il est intéressant de noter que le projet C3 de *Chrysler* (projet de gestion de paie) est historiquement le point de départ de l'*eXtreme Programming*. Ward Cunningham et surtout Kent Beck sont à l'origine de cette méthode, qu'ils ont tous deux mise au point quand ils travaillaient chez *Chrysler*. D'autant que la refactorisation était présente dans la culture SmallTalk dès le début.

Ainsi, puisque le wiki est un outil de *refactoring*, la plupart des logiciels wikis publics « historiques » ne fournissaient pas de structure par défaut, ou bien des outils de structuration simples. Point de menus horizontaux et verticaux sur les bords de la page web, ni de bannière avec des onglets, ou encore de *workflow* de navigation à configurer avant de mettre le wiki en œuvre. Tout cela devient alors inutile, puisque ce sont les utilisateurs qui vont petit à petit structurer le wiki.

2.3.5 Des règles « sociales » pour un fonctionnement idéal

Certaines expériences de wikis publics ont été de retentissants échecs. On peut citer en exemple le wiki mis en ligne par le Los Angeles Times : il fut retiré au bout de deux jours, les pages étant rapidement vandalisées par des photos pornographiques. Ce cas fut discuté lors de la conférence WikiSym 2005. Quelles ont été les causes de cet échec ? Les responsables du L.A Times ont en fait commis de nombreuses erreurs : ils n'ont pas établi de communauté d'utilisateurs, ils n'ont pas « jardiné » leur wiki, ils sont arrivés à un moment où le Web était devenu une terre bien plus hostile qu'à l'époque où C2wiki ou Wikipédia se sont installés, si bien que leur wiki n'a pas eu le temps de s'auto-modérer.

On retiendra de ce cas d'échec la confirmation que l'outil ne peut pas fonctionner si l'on ne tient pas compte de son aspect social. D'une part, il faut une communauté d'utilisateurs et d'autre part, ces utilisateurs doivent avoir envie de collaborer, de s'investir personnellement dans cette collaboration. Si les utilisateurs se contentent d'utiliser le wiki « en touristes », le wiki n'aura aucune chance de fonctionner.

Une autre source d'échec purement conceptuelle et technique est le problème de la structuration des documents d'un wiki. L'organisation libre de la structure du wiki avec la plupart des moteurs de wikis fait qu'ils supportent difficilement la mise à l'échelle. Les wikis d'entreprises notamment montrent des exemples d'échecs dus à ce type de problèmes.

En guise de conclusion, on peut dire que les wikis sont aujourd'hui bien ancrés dans les pratiques culturelles. Peu à peu, ces outils collaboratifs ont envahi la toile et en se démocratisant, ils sont entrés dans une nouvelle phase de développement.

Lecture suggérée pour en savoir plus sur l'histoire des wikis : le livre de Cunningham et Leuf *« The Wiki Way: Quick collaboration on the web »* [60].

Dans les sections suivantes de ce chapitre, nous aborderons l'évolution des wikis, leur impact sur le monde de l'entreprise et sur l'émergence du Web 2.0. L'étude se terminera par une présentation des wikis issus du monde de la recherche, tirant parti à la fois des innovations du Web 2.0 et des technologies du Web sémantique.

2.4 Wikis dans les intranets d'organisations

Remarque préliminaire : dans cette section, je rappelle une étude que j'avais menée à propos de l'utilisation de wikis dans le monde de l'entreprise [55] en 2006. Il est important de se remettre dans le contexte historique. Souvenons-nous qu'à l'époque de nombreuses entreprises ne donnaient ni un accès au mail, ni au Web à leurs employés, et que les outils collaboratifs présentés dans la section précédente étaient souvent remplacés par des produits très contrôlés comme Lotus Notes [61]. Les technologies AJAX et l'API du DOM présentes dans les navigateurs, permettant la création de « web applications » dynamiques venaient à peine de sortir, etc.

La valeur de cette section est avant tout historique, elle demeure à ce jour une des rares études publiées sur l'utilisation de wikis dans l'entreprise et préfigure l'arrivée des Systèmes de Gestion de Contenus (« Content Management Systems » ou CMS en anglais), des outils de Gestion Electronique de Documents (GED), et l'intégration dans les intranets de « réseaux sociaux d'entreprise ». Ces outils contiennent dans leurs gènes de l'ADN de leur ancêtre : le wiki !

2.4.1 Introduction

Entre 2000 et 2006 nous avons mené deux expériences avec des wikis installés dans des intranets : nous avons installé un wiki situé au cœur de l'intranet du département informatique de l'Université de Nice, avec 400 utilisateurs réguliers [58]. Entre 2001 et 2006 nous avions également collaboré avec la société ILOG qui avait développé un intranet centré autour d'un wiki open source, ce qui était très innovant à l'époque [59]

(cette décision avait été prise à l'initiative de Colas Nahaboo[4], avec le support de Jean-François Abramatic[5]).

Par ailleurs, à l'époque, des entreprises comme Google, Motorola ou le New-York Times avaient publié la manière dont elles utilisaient des wikis dans leur organisation. Dans certains cas le concept du wiki avait été adopté et avait amélioré le partage des connaissances, dans d'autres cas cela avait été un échec. Pour quelles raisons ? Quelle est la bonne approche pour favoriser l'adoption d'un wiki dans un intranet d'entreprise ? Et au fond, un wiki est-il nécessaire ?

2.4.2 Intranets : différences avec le Web public (ou bien est-ce la même chose ?)

Nous avons vu dans la section 2.1 que le Web a longtemps été considéré comme un système où le nombre de producteurs de contenu était infiniment plus petit que le nombre de consommateurs. Nous avons également montré que publier sur le Web pré-wikis et pré-blogs nécessitait des connaissances techniques importantes [67].

Des chercheurs comme Chang [68] ont expliqué cette situation par la conception biaisée des navigateurs Web apparus depuis que NCSA Mosaic est devenu le navigateur standard de facto en 1994, c'est-à-dire très rapidement après la naissance du Web. Contrairement au navigateur original créé par Tim Berners Lee qui permettait lire *mais aussi de créer et publier* des documents à distance via HTTP, Mosaic et ses successeurs ont été conçus surtout pour *consulter* des documents distants.

La plupart des intranets de 2006 suivaient le même modèle : contenu produit par un très petit nombre d'employés à qui on avait confié le travail de publication de documents, et délégation des tâches techniques induites à un personnel spécialisé dans la maintenance du système informatique de l'entreprise [69].

« Ne laissez personne s'exprimer sans contrôle ou vous aurez l'anarchie dans l'entreprise très rapidement ! »

Chaque morceau d'information devait respecter les règles « *corporate* » de l'entreprise. La littérature recommandait à l'époque des procédures, des règles, des recettes à suivre afin d'obtenir un bon intranet « en dix leçons » [71, 72].

Au début du 21ème siècle, plusieurs publications ont montré que cette situation trouvait sa source dans la culture du contrôle et de la sécurité ancrée profondément dans l'ADN des entreprises [70]. Il existait un agrément tacite pour dire que le processus de publication était « trop complexe pour être laissé à des non spécialistes ! »

[4] Colas Nahaboo a fait sa thèse à l'INRIA et a travaillé pour Bull en R&D dans une équipe très liée au W3C, hébergée dans les locaux de l'INRIA. Ensuite il est recruté par ILOG Sophia-Antipolis à la fin des années 90. Il est alors responsable de la refonte de l'intranet de cette société.

[5] En 1992 l'INRIA met Jean-François Abramatic à disposition du World Wide Web Consortium (W3C) récemment créé et hébergé dans ses locaux, comme directeur du développement résidant à Boston. Il devient par la suite président du W3C de 1996 à 2001. Il continue à siéger au Comité consultatif du W3C qui compte 9 membres, et dont il avait été le fondateur. En 1999 il est recruté par la société ILOG comme responsable de la recherche et du développement.

Rapidement, les entreprises ont pris conscience que leur intranet était pauvre, statique, rarement mis à jour et peu utilisé. De nombreuses frictions ralentissaient le processus de publication et les personnes en charge du contenu sur l'intranet se plaignaient régulièrement de la complexité de la tâche qui leur incombait [58, 59].

2.4.2.1 Intranet : le point de vue de l'organisation

Pour une entreprise, l'intranet joue différents rôles : c'est un outil de communication ("*prévenons nos employés que quelque chose s'est passé*"), il peut contenir des services ("*allez à cette URL et vous trouverez le répertoire des personnes*"), etc. Cependant, en tant que système d'information, il peut grandement améliorer le partage des connaissances entre les membres d'une organisation et devenir *sa mémoire*. Idéalement, l'Intranet est un outil de gestion des connaissances (GC) qui peut stimuler la productivité, aider à prendre les bonnes décisions et aider l'organisation à survivre à la perte éventuelle de certains de ses membres ou équipes clés.

Bientôt, des chercheurs comme Steinmark [62, 63] ou Hahn et Subramani [66] ont déclaré que " le succès de la GC dépend des contributions de tous les membres de l'organisation, et pour que l'intranet serve d'environnement de partage des connaissances, *une forte participation est importante* ".

Pierre Levy décrit pour sa part l'intranet comme un cyberespace partagé qui peut être considéré comme un écosystème d'idées [61]. Son hypothèse suppose qu'il existe une forte corrélation entre l'intelligence collective d'une organisation (l'équilibre dynamique de toutes ses dimensions) et ses performances sur le marché économique.

Ainsi, la principale difficulté consiste à capturer et à rendre disponibles les connaissances sur l'intranet. Régulièrement, ces connaissances sont transportées et échangées sans appel à la technologie (discussions autour d'une table de réunion, à la machine à café, etc.) mais lorsque la technologie est impliquée, il devient possible de capturer, d'archiver et de transformer ces connaissances en informations consultables. Les bases de données, les documents comme les rapports, les courriels ou les pages Web représentent de la connaissance. Cependant, il reste deux défis à relever : (1) Comment capturer plus de connaissances ? En particulier les connaissances implicites et informelles qui ne se trouvent pas dans les documents d'entreprise des intranets de l'époque précédente (pré-2005) : brainstorming qui se déroule dans des réunions improvisées, décisions prises à la volée pour résoudre un problème inattendu, projets de documents jamais publiés qui peuvent contenir des informations pertinentes, listes de contacts, discussions informelles, *modus operandi* qui ne franchissent jamais les frontières d'une équipe particulière à l'intérieur de l'entreprise, etc. et (2) comment rendre ces connaissances largement disponibles si l'intranet est en lecture seule et si le courriel ne fonctionne pas, comme nous l'avons vu dans la section 2.2.1.

Pour réussir, les mots-clés sont : participation élevée, collaboration, implication de l'utilisateur, confiance, autonomie.

Le modèle de publication centralisée sur le Web devait être repensé.

2.4.2.2 Intranet : le point de vue du salarié

Pour les membres d'une entreprise, l'intranet est un outil d'archivage, de partage et de recherche d'informations. C'est aussi un *outil pour mieux travailler ensemble*.

J. Nielsen a décrit l'intranet idéal du point de vue de l'utilisateur dans [75] : « *partager en un seul endroit toute l'information nécessaire au projet ; tout le monde peut facilement trouver cette information et contribuer à l'information existante, tuons le webmaster !* »

Les échanges, les opinions, les discussions sont écrits en partie à l'aide d'ordinateurs. Le travail peut être effectué de manière asynchrone, même à distance (à la maison, sur la route). Les acteurs du projet disposent d'une mémoire visible du projet. Les actions de chacun peuvent être retracées en regardant l'historique des changements. La participation à cet espace de connaissances partagées doit améliorer les liens sociaux. Les outils doivent rester simples et pousser les lecteurs à devenir des écrivains.

Mais quels que soient les talents que l'on peut avoir, quelle que soit la tâche à laquelle on assigne un employé, il est nécessaire de s'adapter aux besoins et aux contraintes des autres. Quelle que soit l'autonomie que l'on peut obtenir, les buts et objectifs personnels doivent toujours - en principe - contribuer partiellement ou totalement aux buts et objectifs de l'organisation, et les responsables de l'organisation, ceux qui jugent le travail des autres, doivent être d'accord avec cela. Si l'organisation d'une entreprise est mauvaise, s'il y a trop de frictions sociales, les employés ne partageront pas leurs connaissances et la solution au meilleur problème de travail ne peut pas être améliorée ou résolue par un outil magique !

La description de Nielsen suppose que l'intranet est un média *inscriptible*. Entre 1998 et 2006, des outils sont apparus qui, d'une certaine manière, ont transformé le World Wide Web en un média inscriptible, résolvant ainsi le syndrome du webmaster : ce sont principalement les blogs et les wikis.

Ces outils sont rapidement devenus un phénomène, comme nous l'avons vu dans la section 2.3. Mais là où les blogs ont tendance à exprimer la voix d'une seule personne [76], les wikis sont plus centrés sur l'écriture et le partage de l'information.

Ce chapitre s'intéresse principalement aux wikis en tant qu'innovation technologique qui peut aider à améliorer la convivialité d'un intranet en stimulant la collaboration des utilisateurs et le partage des connaissances.

Nous allons essayer de comprendre pourquoi les wikis ont été un tel phénomène sur le World Wide Web. Nous verrons que la philosophie du concept wiki tel que proposé par son inventeur, Ward Cunningham, *ne peut être transposée telle quelle* aux intranets.

2.4.3 Intranets étudiés

Les nombreux articles sur les wikis qui ont été publiés sur le web ou dans la presse populaire illustrent l'intérêt croissant pour ce phénomène par le public mais aussi par les acteurs de l'industrie. Les wikis d'entreprises, par définition, s'exécutent derrière des pare-feu et sont plus difficiles à étudier. D. Steinmark, qui a écrit plusieurs articles sur ce sujet,

a remarqué qu'une recherche sur le mot-clé "wiki" en 2006 sur la bibliothèque numérique de l'ACM a retourné moins de 100 résultats, mais tous les articles ont été réalisés sur le Web public [64].

Entre 2000 et 2006, nous avons eu l'occasion de regarder de près, pendant une longue période de temps, deux grands intranets différents articulés autour d'un wiki.

Le premier a été installé par l'auteur dans l'intranet du Département Informatique de l'Université de Nice (France) en 2000 et est utilisé par tous les enseignants, étudiants, partenaires industriels, etc. En 2006 il contenait plus de 4000 pages et comptait plus de 400 utilisateurs réguliers. Nous avons utilisé ce wiki chaque année entre 2001 et 2006 pour des projets coopératifs d'étudiants au-delà des frontières nationales (voir [58] et [77]). C'était aussi un outil quotidien pour la plupart des élèves et des enseignants. En 2006, nous avons mené des interviews avec 17 étudiants en dernière année de maîtrise d'informatique et avec 5 enseignants (sur 23). Les interviews portaient sur l'utilisabilité du wiki, comment et pourquoi il a été utilisé, etc. Une synthèse des résultats est présentée dans les sections suivantes.

Le second wiki était au cœur de l'intranet de la société française ILOG (maintenant IBM qui l'a rachetée en 2009). Installé en 1999, il est en 2006 composé de 37 000 pages web et compte plus de 700 utilisateurs. Depuis 2000, nous avons mené avec ILOG plusieurs projets d'étudiants de master et encadré de multiples stages, tous en rapport avec le wiki installé dans l'intranet de cette entreprise. Tous les projets et stages avaient pour but d'améliorer la convivialité du wiki et nous avons finalement mis en place une collaboration officielle entre l'INRIA et ILOG sous la forme d'une action de recherche appelée "Usable Intranet"[6]. Pendant ce temps, nous avons eu accès à l'intranet d'ILOG, nous avons effectué des réunions régulières avec l'équipe en charge de cet intranet et deux étudiants en maîtrise spécialisés en ergonomie ont passé six mois de formation dans l'entreprise. Ils ont mené plusieurs entretiens avec plus de 20 employés de ILOG afin d'analyser l'utilisation de l'intranet.

D'autres sources ont été utilisées pour cette enquête. En 2006, Google a présenté son organisation interne : l'exposé de Shashi Seth (*lead product manager*) lors de la table ronde "corporate wikis"[7] qui a eu lieu lors de la conférence WikiSym 2005 ainsi que l'exposé de Douglas Merrill (senior director, information technology) lors de la conférence Vortex 2005 ont confirmé que Google, depuis ses débuts, utilise massivement des outils collaboratifs[8] [73, 74]. En particulier, son intranet s'articulait alors autour d'un grand wiki qui utilise le moteur TWiki.

Lors de cette table ronde, Peter Thoeny de WindRiver, créateur du logiciel TWiki[9], a parlé de différentes *success stories* dans de grandes organisations utilisant TWiki[10]. Thomas Weigert (Motorola Global Software Group) a décrit le wiki dans l'intranet de sa

[6] http://www-sop.inria.fr/acacia/Colors-2005-UsableIntranet-Page-Web.html
[7] Note : j'ai participé à cette table ronde.
[8] Inside Google, Rough Type Blog, http://www.roughtype.com/archives/2005/10/inside_google.php
[9] http://www.twiki.org/
[10] TWiki success stories : http://twiki.org/cgibin/view/Main/TWikiSuccessStories

société. Nous avons également eu l'occasion de parler avec Crawford Currie qui a été le pionnier des wikis chez Motorola[11] et qui a créé une petite société, Cdot.com qui vend des services autour des wikis pour les intranets.

Par ailleurs, trois entreprises qui gagnent leur vie en vendant des services intranet autour des wikis ont participé aux discussions et ont présenté leurs services et produits : SocialText est célèbre pour l'introduction d'un wiki chez Nokia, JotSpot a développé une solution autour d'un moteur wiki très innovant (racheté par Google, il est aujourd'hui l'ancêtre de Google Docs) et XWiki, une petite société française célèbre pour le logiciel éponyme XWiki.

2.4.3.1 Les wikis ont été introduits dans les intranets par la porte de derrière !

2.4.3.1.1 Comment les gourous wiki ont introduit les wikis dans les intranets.

Nous avons démarré l'intranet du Département d'Informatique de l'Université de Nice parce qu'à l'époque nous n'avions pas d'intranet ! En 1999, nous n'avions que des pages web personnelles et un site web public officiellement mort qui n'avait pas été mis à jour depuis des mois. Nous avons installé un wiki et l'avons d'abord utilisé, à titre de test, pour certains projets étudiants collaboratifs impliquant une coopération à distance entre pays. Nous avons eu des étudiants de Nice, en France, qui ont travaillé sur les mêmes projets avec des étudiants de Mannheim en Allemagne. Les résultats de cette expérience ont été publiés dans [58] et [77]. Les entretiens avec les acteurs de ce projet ont montré que le wiki était globalement très apprécié. Puis nous avons décidé de l'utiliser pour l'ensemble de l'intranet du département informatique. Nous avons commencé à remplir le wiki avec des cours, des FAQs, des informations générales sur la vie quotidienne dans le département. Nous avons également demandé à tous les étudiants de s'inscrire et ils ont été obligés d'utiliser le wiki pour publier et archiver tous leurs projets. J'étais le wiki gourou/évangéliste/jardinier jusqu'à la fin 2005. D'autres collègues m'ont remplacé dans ce rôle car le wiki était apprécié par la majorité de notre communauté et est devenu auto-modéré au fil des ans. Il conserve la mémoire du département informatique depuis 2000 jusqu'en 2012, date à laquelle il a été remplacé par un logiciel plus récent (un autre moteur de wiki).

Nous avions entendu parler du concept wiki par Colas Nahaboo, un ami qui travaillait pour la société française Bull et dont l'équipe était hébergée par l'INRIA. Il a installé un wiki là-bas et nous a conseiller de faire de même à l'Université de Nice. Il était le gourou du wiki local et aimait vraiment la façon dont les membres de son équipe pouvaient l'utiliser comme bloc-notes partagé. Fin 2000, toute son équipe a été rachetée par ILOG. Sa mission consistait à rénover et améliorer l'intranet de la division R&D qui paraissait pauvre par rapport à l'intranet de l'INRIA. Certains serveurs WebDav offraient de l'espace disque partagé pour l'échange de fichiers, certains modèles de documents étaient disponibles pour les formats HTML ou Word et chacun éditait les documents à l'aide de

[11] Les premières années d'un wiki chez Motorola sont décrites dans le "Wiki at work", chapitre du livre de Ward Cunningham "The Wiki Way". Crawford Currie est l'un des principaux contributeurs de TWiki.

son propre éditeur. Il y avait quelques serveurs web officiels mais à chaque fois qu'il y avait un problème, une seule personne pouvait aider / dépanner. Les gens n'aiment pas être dépendants, alors une galaxie de serveurs web est apparue : presque un serveur dans chaque ordinateur de chaque équipe ! Il n'y avait pas de cohérence dans l'intranet. Comme décrit dans [59], le wiki est rapidement devenu la pierre angulaire de l'intranet du ILOG.

Presque la même histoire s'est produite au même moment chez Motorola. Un petit nombre d'ingénieurs dont l'équipe pratiquait l'Extrême Programming a installé un wiki comme bloc-notes partagé, puis d'autres équipes ont remarqué cet outil amusant et ont commencé à l'utiliser. Les personnes qui ont installé le wiki ont commencé à créer des espaces de travail pour ces autres équipes et le wiki s'est développé. En 2000, sept équipes ont utilisé le wiki, distribué dans différents pays : L'Allemagne, l'Angleterre, la France, l'Australie, la Russie et les États-Unis. A cette époque, 60 personnes contribuaient au wiki. Une personne à Motorola, le gourou wiki local (Crawford Currie), a joué le rôle de l'évangéliste et du jardinier du wiki. En 2004, cinq sites wiki étaient en activité, utilisés par 17 équipes dans le monde entier. Les statistiques ont montré plus de 16000 pages consultées chaque jour, faisant du wiki l'outil le plus populaire de l'intranet, bien au-dessus de l'outil officiel de gestion de la base de connaissances qui existait auparavant.

Le New-York Times Digital a aussi une histoire wiki similaire : en 1999 une petite équipe avec des compétences techniques a décidé d'installer un wiki pour aider à maintenir la documentation d'un projet sur lequel elle travaillait (voir le chapitre "wikis at work" du livre de Ward Cunningham [60]). Steve Wainstead, était le gourou du wiki. Avant ce wiki, la documentation en ligne était écrite en HTML à l'aide d'un éditeur de code populaire sous Unix, et l'accès au serveur web n'avait été autorisé qu'à quelques personnes qui connaissaient les mots de passe. Le wiki s'est développé, surtout parce que Noreen Wu, une autre évangéliste wiki au NY Times Digital, l'a utilisé pendant les réunions pour rédiger les procès-verbaux. D'autres participants se sont montré curieux à propos de cet outil et Noreen en a fait la démonstration. Les statistiques de l'utilisation du wiki ont montré quelques pics d'utilisations suite à ces réunions. Et après quelques semaines, d'autres équipes ont demandé si elles pouvaient avoir leur propre espace sur le wiki. Managers, secrétaires, journalistes, tout le monde a commencé à utiliser ce wiki. Et c'est devenu le plus grand dépôt de documents du NY Times Digital. Puis des problèmes sont apparus, mais nous en parlerons dans la section 2.4.4.

Le cas de Google est un peu différent car le wiki était là depuis les premiers jours, quand Google était une petite startup faite de geeks. Et les geeks ont adoré le concept wiki !

2.4.3.1.2 Pas de gourou wiki = pas d'intranet wiki ? Les petites entreprises ont comblé le vide !

Nokia est un autre cas célèbre d'un wiki qui a été introduit à l'initiative de quelques individus. Mais cette fois, il n'y avait pas de gourou wiki local !

La division Insight and Foresight de Nokia est spécialisée dans la veille technologique. Les membres de cette division étaient confrontés à d'énormes problèmes avec le courrier qui était leur principal outil de collaboration à l'époque : trop de messages, pas de moyen

facile de trouver et de structurer l'information. L'activité de cette division nécessitait un accès rapide à son capital de connaissances. Ils ont entendu parler de SocialText, une petite startup qui proposait une solution alternative aux problèmes de messagerie basée sur les wikis, les blogs et d'autres outils collaboratifs optionnels comme les messageries instantanées (voir Figure 9).

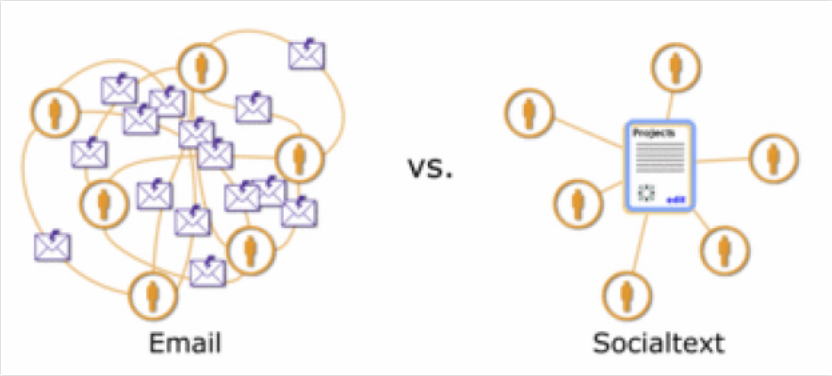


Figure 9. La société SocialText proposait de résoudre les problèmes liés au mail.

2.4.3.2 Profils d'utilisateurs dans les intranets étudiés

Les entretiens ont confirmé ce que les contacts quotidiens avec les utilisateurs d'ILOG et avec le département CS de l'Université de Nice ont montré de manière informelle : les utilisateurs sont des lecteurs qui parcourent les documents wiki et certains d'entre eux sont des auteurs/contributeurs réguliers. Les étudiants du département informatique sont un cas particulier car ils sont obligés d'utiliser le wiki pour leurs cours et leurs projets. De plus, pendant leur période de stage, ils ont besoin de mettre à jour leur journal personnel de stage (une page du wiki).

Ils contribuent donc tous régulièrement au corpus documentaire du wiki. Comme nous le verrons, les enseignants en informatique sont paradoxalement les plus difficiles à convertir au concept wiki.

Néanmoins, un noyau de profils communs peut être identifié parmi les utilisateurs du wiki que nous avons interviewés :

- **Techno-maniaques** : à chaque fois qu'une nouvelle technologie est disponible, ils sautent dessus. Étrangement, même dans les organisations où les utilisateurs connaissent l'informatique, cette catégorie est la plus réduite.
- **Utilisateurs techniques compétents** : ils utilisent le wiki de manière autonome et apprécient beaucoup cette nouvelle liberté. Chez ILOG, ils écrivent également de petites applications embarquées de type Excel en utilisant le langage de macros du moteur TWiki. Ils procèdent généralement par imitation ; quand ils voient des choses qui leur plaisent dans une autre page wiki, ils dupliquent (par copier-coller) le contenu dans leurs propres pages, et le modifient selon leurs besoins spécifiques. A l'Université de Nice, nous trouvons dans cette catégorie des enseignants qui utilisent le wiki depuis plusieurs années, qui ont commencé à l'aimer et qui ont finalement lu la documentation

en ligne pour apprendre l'utilisation de certaines macros afin de publier des formulaires, faire des tableaux éditables, des tableurs en ligne, etc. Ils ont également procédé de cette façon après avoir vu ce que d'autres ont fait. Les étudiants en génie informatique curieux tombent dans cette catégorie qui représentait environ 10 % des utilisateurs.

- **Suiveurs** : ces derniers n'ont pas aimé le wiki au début mais l'ont adopté après quelques mois quand ils ont vu que beaucoup de leurs pairs l'utilisaient. Nous l'avons constaté avec les enseignants sur l'intranet du département informatique (rappel : les étudiants étaient obligés d'utiliser le wiki pour leurs projets). Il a fallu beaucoup de temps pour faire admettre à certains enseignants que le wiki était finalement un outil utile. Chez ILOG, ce sont eux qui ont accepté de travailler avec les mêmes outils que ceux que d'autres utilisent pour faciliter leur intégration sociale. Et si un manager utilise le wiki, il y a de très bonnes chances qu'il puisse convaincre ses subordonnés de l'utiliser également ! Les adeptes représentent environ 80 % de la population.
- **Contributeurs non wiki.** Ce sont les managers de ILOG qui ne veulent pas changer leurs habitudes, les nouveaux venus, ou ceux qui rejettent le wiki pour différents prétextes comme l'absence d'un éditeur WYSIWYG. Ces derniers sont souvent des informaticiens qui ne peuvent pas accepter d'utiliser un outil *aussi simple*. Notre supposition est que les moteurs wiki sont des logiciels rudimentaires, avec des fonctionnalités limitées (interfaces graphiques réduites à un champ de formulaire HTML, etc.), conçus pour être utilisés à partir d'un navigateur web, sans qu'il soit nécessaire d'installer un logiciel client supplémentaire. Certaines personnes ne peuvent que voir les lacunes de ce compromis et se focaliser sur les aspects primitifs du wiki. Elles refusent simplement d'utiliser un logiciel aussi simpliste, ce qui va à l'encontre de tout ce qu'ils ont appris au cours de leurs études où les solutions complexes semblaient être les meilleures. Dans l'intranet du département informatique, ils sont pour la plupart des enseignants en informatique, et rechignent à utiliser le wiki pour ces raisons. Ils représentent environ 45% des enseignants du département ! Nous n'avons pas d'utilisateurs non wiki parmi les étudiants.

Les utilisateurs, après une petite enquête, remarquent néanmoins qu'il n'y a pas de solution alternative facile pour remplacer ce que le wiki propose et finissent finalement par l'adopter. A la question "*quelle meilleure solution alternative proposez-vous pour remplacer le wiki*", 100% des personnes du département informatique n'ont proposé aucune autre solution ou ont proposé un autre moteur wiki dont ils ont entendu parler sur le Web (Plone, dans ce cas particulier). Même ceux qui ont le plus critiqué le wiki, qui n'ont jamais contribué, ont admis qu'ils ne voient pas de meilleure solution jusqu'à présent, mais *il devrait y en avoir au moins une* !

2.4.3.3 Personne n'est anonyme !

Dans tous les intranets que nous avons étudiés, les utilisateurs n'étaient pas anonymes, contrairement aux grands wikis publics de l'époque 1995 - 2005 qui sont basés sur l'anonymat afin de faciliter les contributions (voir section 2.3).

Dans la section 2.4.2, nous avons expliqué comment l'état d'esprit de l'organisation a conduit à la féodalité de l'information. L'émergence des wikis et autres logiciels sociaux dans les intranets est un processus très lent et ces nouveaux moyens de publication de contenu web sont encore entourés de suspicion et de doutes ("*que se passe-t-il si quelqu'un publie de mauvaises choses sur l'entreprise ?*"). L'anonymat n'est même pas une éventualité comme le confirment les discussions avec les responsables de tous les intranets étudiés. Même pour l'intranet académique du Département d'Informatique de l'Université de Nice, nous avons dû identifier les contributeurs au wiki pour des raisons juridiques : chaque étudiant et chaque enseignant chercheur signe un contrat qui stipule que l'auteur d'une page web assume la responsabilité s'il publie un contenu qui ne suit pas les directives de l'Université. Dans un contexte organisationnel, les gens aiment aussi être identifiés. Là où l'anonymat peut réduire l'appréhension de l'évaluation dans les wikis publics ou sur le Web public, les participants à un intranet d'entreprise ont l'habitude d'être identifiés, de parler librement dans un espace ouvert, de partager leurs bureaux, de parler entre eux. Dans un tel environnement, l'anonymat n'est pas l'attribut auquel on pourrait penser. Steinmark a remarqué dans une expérience située au siège social de Volvo Information Technology que l'anonymat est une épée à double tranchants : il réduit l'appréhension mais a un effet négatif sur la participation si les employés ont le sentiment que leur contribution ne sera jamais reconnue ni récompensée [65].

Personne parmi les utilisateurs de wikis dans les intranets étudiés ne s'est plaint du manque d'anonymat. Tous les intranets décrits dans ce document utilisent un moteur wiki qui gère l'identification des utilisateurs et toutes les contributions au wiki sont tracées dans l'historique de chaque page.

2.4.3.4 Ce que nous trouvons dans les wikis d'intranets

L'intranet du département informatique de l'Université de Nice est organisé par type de contenu. Nous avons un espace de travail pour les cours, un espace de travail pour les projets, pour les clubs étudiants, pour discuter de la vie sociale du département, etc. La taille du corpus est d'environ 4000 pages réparties dans 25 espaces de travail différents.

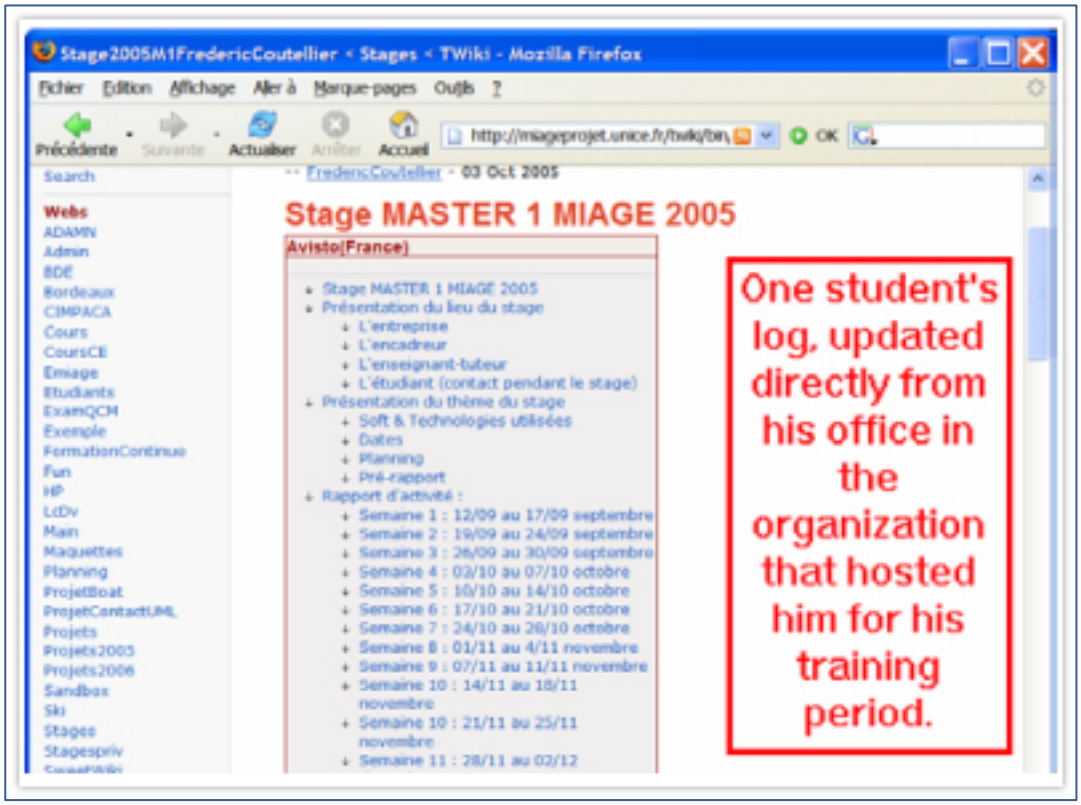


Figure 10. Journal de bord d'un étudiant en stage, mis à jour chaque semaine.

La plupart des documents sont des pages wiki standard, principalement du texte, parfois avec des documents et des archives attachés aux pages. Les enseignants mettent dans leurs pages wiki : notes de cours, exercices, parfois ils joignent des fichiers PowerPoint ou PDF, des fichiers ZIP, etc.

Les étudiants utilisent principalement le wiki pour mettre à jour et publier leur travail de projet (Figure 11) ou stage (Figure 10). Ils utilisent aussi beaucoup l'espace de travail "Etudiants" pour leurs propres discussions sur des sujets tels que l'organisation de week-ends ou un voyage à la montagne. Nous trouvons aussi des pages qui ne sont pas liées les unes aux autres ; le wiki est utilisé dans ce cas comme un bloc-notes en ligne improvisé.

Sur l'intranet d'ILOG, il n'y a pas de pages personnelles, pas d'espace de travail dédié aux hobbies ou au plaisir. Uniquement des documents utiles aux tâches quotidiennes. L'intranet est structuré en plusieurs espaces de travail, un pour chaque équipe au sein de la division R&D et il est actuellement composé de 37 000 pages. A l'intérieur de chaque espace de travail, les équipes sont libres de s'organiser. Nous trouvons principalement des pages utilisées pour la gestion de projet informelle classique : ordre du jour, procès-verbaux de réunions, tâches en cours, documentation produit à réviser, suivi des bogues logiciels, brainstorming, enquêtes sur divers sujets, blogs à l'intérieur du wiki, etc. La première chose que nous avons remarquée lorsque nous avons étudié cet intranet est l'omniprésence des tableaux, en particulier des tableaux de type Excel. Nous n'avons pas trouvé une telle chose sur l'intranet du département informatique. Ces tables sont en fait *de petites applications embarquées* écrites à l'aide des macro-commandes disponibles dans le TWIkiML, le langage de formatage utilisé par le moteur TWiki. Le logiciel TWiki

est ce que nous appelons "*un wiki d'application*" ou un "*wiki de composants*". On peut créer des formulaires, et il existe des outils simples pour archiver les données saisies dans ces formulaires sous forme de pages wiki.

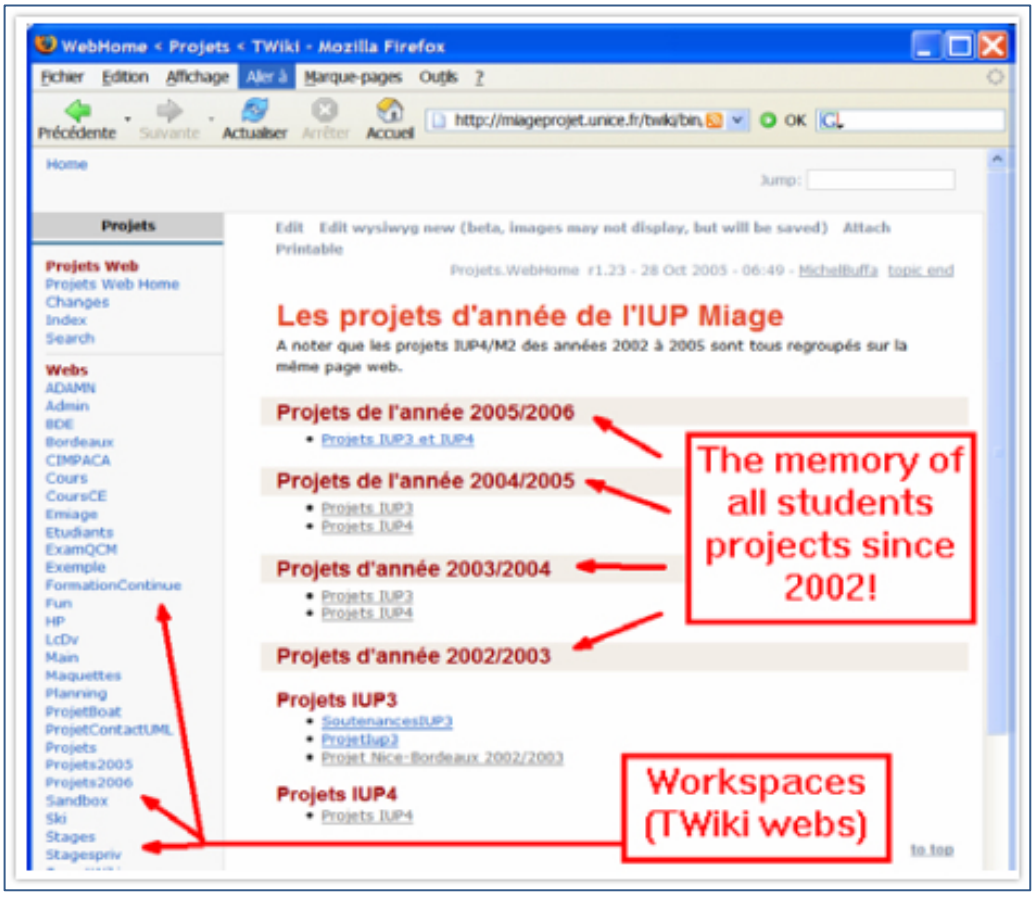


Figure 11. Tous les projets étudiants depuis 2002 se trouvent sur le wiki du département informatique de l'Université de Nice.

Une macro `%SEARCH%` très puissante (elle utilise des expressions régulières) peut à son tour être utilisée pour chercher dans le corpus des pages du wiki les données structurées capturées à partir de formulaires, et les présenter sous forme de tables dynamiques. ILOG a également installé des *plugins* pour TWIki pour créer des tableaux dont le contenu peut être édité directement sur la page web ou pour transformer ces tableaux en feuille de calcul (avec des calculs dynamiques comme dans Excel). La Figure 12montre un tableau éditable à partir d'une page wiki d'un ILOG : le contenu des cellules peut être modifié directement en tapant du texte ou en utilisant des listes déroulantes, des lignes ou des colonnes peuvent être ajoutées ou supprimées. La Figure 13 montre un tableau composé de données extraites d'autres pages wiki. Cliquer sur l'icône "éditer" à gauche de chaque ligne apporte une page wiki en mode édition : celle qui contient les données correspondantes. La Figure 14 montre une telle page ; nous remarquons quelques WikiML en bas tandis que le haut de l'éditeur est un formulaire pour éditer les données structurées.

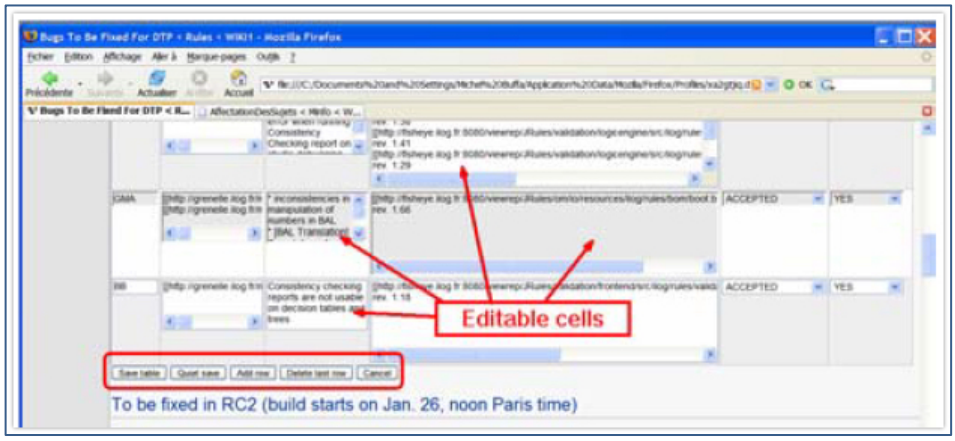


Figure 12. Un exemple de page contenant un tableau éditable. Intranet d'ILOG, 2006

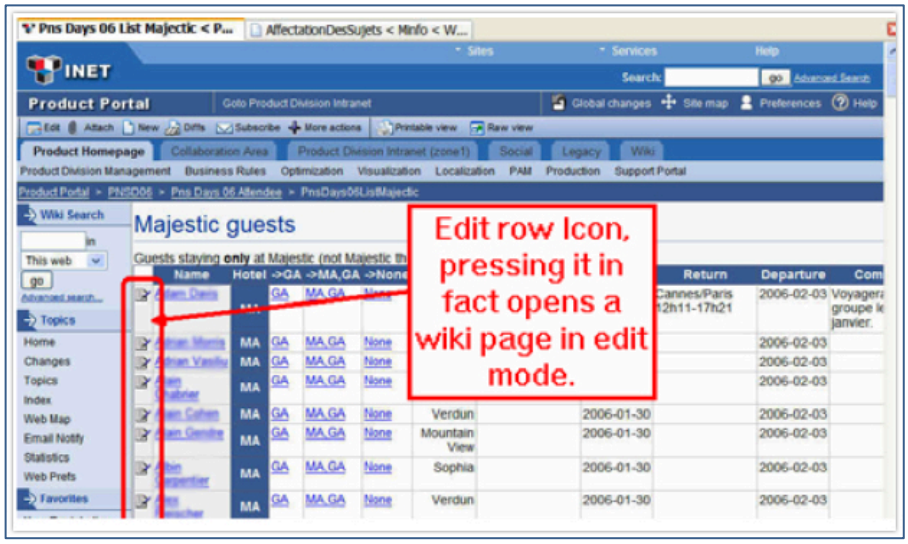


Figure 13. Une application embarquée dans le Wiki d'ILOG (2006). Les données présentées sont extraites à partir de données structurées situées dans d'autres pages du Wiki, à l'aide d'une macro %SEARCH% et du langage de macros TWikiML.

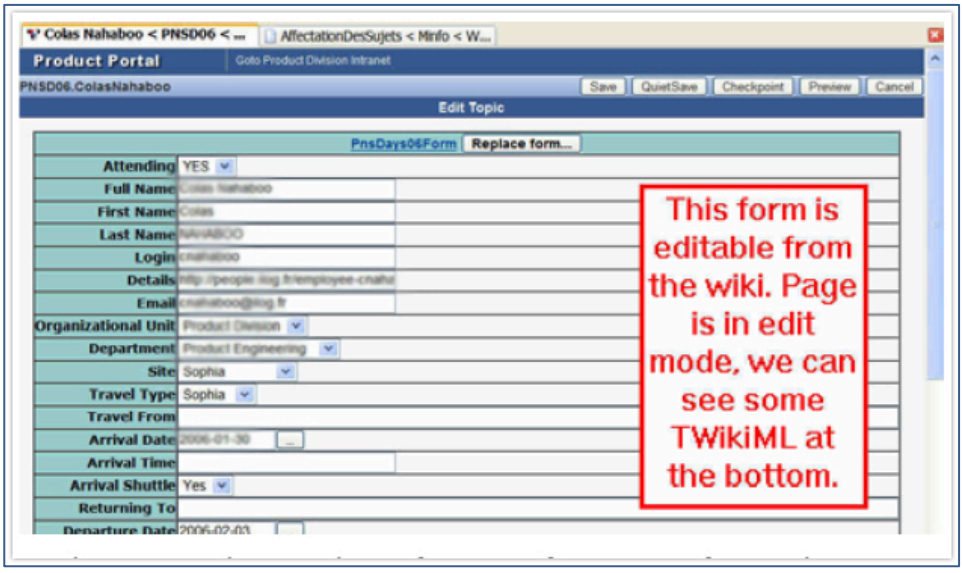


Figure 14. Vue éditable d'une rangée du tableau à partir de la Figure 13.

Les entretiens ont montré que les employés d'ILOG qui contribuent au wiki considèrent qu'il est facile à utiliser et l'utilisent spontanément pour des tâches très diverses. Il semble être le principal outil intranet, comparable en termes de fréquence d'utilisation, au courrier électronique.

La gestion de projet informelle sur le wiki diffère beaucoup d'une équipe à l'autre. Certaines équipes ne créent que des pages textuelles pures, d'autres utilisent beaucoup de micro-applications. Le chef d'équipe doit se sentir à l'aise avec l'outil : il ressort de l'analyse des logs d'utilisation qu'il est très souvent le principal contributeur dans l'espace de travail de son équipe. Parfois, il délègue cette tâche d'animation à une autre personne de son équipe, mais les interviews ont montré que c'était plus rare.

Comme il n'y a pas de règles qui indiquent comment faire les choses sur le wiki, chaque équipe le fait à sa façon.

Crawford Currie et Thomas Weigert (Motorola Global Software Group) ont confirmé lors de la table ronde "intranet wiki" qui a eu lieu lors de la conférence WikiSym 2005 que le wiki de Motorola est également utilisé pour la gestion informelle de projets, d'une manière très similaire à ce que fait ILOG, et inclut également de nombreuses applications embarquées. Voir [60] (chapitre "Wiki at work") pour plus de détails.

Un représentant de Google, Shashi Seth, lors de la même table ronde, a rapporté que l'énorme intranet de Google est basé sur un wiki, et que la production de documents intranet dans les formats Office est fortement déconseillée : il est acceptable d'utiliser le format Office pour les documents à auteur unique, mais les choses qui doivent être éditées par d'autres *doivent être* des documents Wiki. Google utilise un moteur TWiki - un point commun avec les autres intranets étudiés dans cette section -, et qu'il est utilisé pour recueillir autant de données que possible sur les changements rapides qui se produisent dans l'entreprise. Douglas Merill pointe vers la base de données des projets de Google (située sur le wiki) et la décrit comme une solution pour faire face aux changements rapides. Il décrit le système mis en place comme du *reporting*, et non comme un système de suivi de projets. Ce *reporting* permet aux "Googlers" de garder une trace de ce sur quoi eux-mêmes et d'autres dans l'entreprise travaillent. "*Nous créons des données publiques, pour que tous puissent voir et se mettre à jour sur les projets et maximiser les possibilités de pollinisation accidentelle entre les équipes*", a dit M. Merrill. "*Tout le monde, de l'ingénierie à la vente en passant par les gens qui balaient les planchers, peut lire et commenter tout ce qui se trouve dans la base de données du wiki.*" Presque toutes les activités passent par le prisme du wiki. Voir [74] pour plus de détails.

2.4.3.5 Moteurs Wiki dédiés à l'utilisation de l'intranet

Les premiers « wikis d'application »

TWiki est le moteur wiki le plus ancien dédié à l'utilisation de l'intranet et le plus déployé en 2006. De nombreuses grandes entreprises l'ont utilisé pendant des années (et l'utilisent

encore en 2018[12]). Il s'agit d'un moteur open source qui a été initialement développé par Peter Thoeny pour l'intranet de la WindRiver Company pour laquelle il travaillait. Il comprenait des fonctionnalités clés comme l'authentification des utilisateurs, un système de gestion de versions, la possibilité de joindre des fichiers à chaque page wiki, un système de notification par courriel pour suivre les changements sur les documents wiki, etc. Open source, avec une architecture de plugin, il a été constamment amélioré par les contributions de ses utilisateurs, dont beaucoup travaillent pour les organisations qui l'ont utilisé dans leur intranet. Il s'agit d'un "wiki d'application" dans le sens où le langage TWikiML utilisé pour formater les documents inclut des macros très puissantes. Ces macros rendent l'écriture d'applications Excel-like facile, sans avoir besoin de mettre en place une base de données ni d'écrire plus que quelques lignes de code. La communauté des utilisateurs est l'une des plus importantes.

JotSpot, le moteur wiki développé par la société éponyme est un wiki "nouvelle génération", très influencé par TWiki, selon ses créateurs. JotSpot a été le premier moteur wiki à être livré avec un éditeur WYSIWYG de haute qualité, tout en proposant un éditeur WikiML basé sur le texte. La société JotSpot exploite une ferme de wikis pour l'hébergement externe d'intranets privés, ou peut déployer une des serveurs JotSpot (JotSpot Appliance) derrière le pare-feu d'une organisation. La majorité des utilisateurs de JotSpot n'utiliseront certainement que l'éditeur WYSIWYG, importeront et exporteront des documents à partir d'applications Office ou utiliseront le JotSpot Tracker, un outil innovant qui transforme n'importe quelle table/application Excel en un clone basé sur le Web. JotSpot propose également une "galerie d'applications" où en un seul clic on instancie une application complète comme une application de recrutement, un agenda, un blog, etc. Ces applications se présentent sous la forme d'un ensemble de pages wiki et peuvent être éditées, personnalisées, etc.

JotSpot supporte un langage de balisage (syntaxe XML) appelé JotScript. Les utilisateurs avancés peuvent écrire leurs propres applications en JotScript. Google a rapidement racheté JotSpot, très innovant pour son époque, qui a servi de base à la suite Google Docs que l'on connaît aujourd'hui.

XWiki, un moteur wiki open source qui jouait aussi dans la même ligue. La société XWiki exploitait une ferme de wikis pour l'hébergement externe et vendait des services pour aider les organisations à mettre en place XWiki à l'intérieur de leur intranet. Très ambitieux, le moteur XWiki supportait les web services, les applications embarquées, etc. mais en 2006 n'avait pas encore été accepté par les grandes entreprises.

D'autres wiksq d'applications existaient à l'époque, comme Confluence, MoinMoin, Daisy, mais nous avons préféré nous concentrer sur les plus déployés à l'époque.

Moteurs de wikis simples, dédiés à la production de documents textuels

L'écriture d'applications dans les pages wiki est certainement un besoin pour certains intranets où quelques utilisateurs ont des compétences techniques et où les applications

[12] http://twiki.org/cgi-bin/view/TWiki/WhatIsTWiki, section « Who is using TWiki ».

et les tableaux de type Excel sont un moyen commun de communiquer des informations et des connaissances, mais il existe de nombreux scénarios où un wiki simple basé sur le texte est suffisant.

Le moteur wiki de SocialText est très simple, avec des fonctionnalités limitées. Et c'est parfait pour les organisations non techniques qui recherchent un outil de brainstorming qui peut stimuler la créativité. Ross Mayfield de SocialText déclare : "*Eclipse est un environnement de programmation devinez le bouton ; nous ne voulons pas que notre wiki soit comme ça ! Gardons les choses simples et utilisables* ». L'enquête menée au sein du département informatique l'Université de Nice a montré qu'il n'y avait pas besoin d'un wiki d'application. Le climat organisationnel, les tâches qui doivent être effectuées guideront le choix d'un type de moteur wiki ou d'un autre. JotSpot ou TWiki sont cependant également très simples à utiliser, personne n'est obligé de lire la documentation ni d'écrire des applications complexes avec des macros cryptées.

2.4.3.6 Partage des connaissances avec le wiki

Mettre en place une culture du partage : le cas de Google.

En 2008, il existe une forte culture de partage chez Google depuis le tout début, dix ans auparavant. L'intranet de Google est alors constitué de nombreux outils différents mais il s'articule autour d'un grand wiki. Tous les employés sont poussés à partager librement et à apprendre les uns des autres, pas d'équipe isolée (pas de silo), et la communication ouverte est la règle. L'entreprise essaie d'embaucher des gens intelligents avec qui il est agréable de travailler, avec un esprit ouvert et adapté aux pratiques de partage. « *Tout le monde, de l'ingénierie à la vente en passant par les gens qui balaient les planchers, peut lire et commenter tout ce qui se trouve dans la base de données des documents produits sur le wiki* » (D. Merill, Google [74])

La perte de pouvoir induite par le partage des connaissances peut limiter considérablement la contribution des gens. Même avec des employés sélectionnés pour leur aptitude à partager l'information, dans le contexte concurrentiel d'une entreprise, de mauvais sentiments peuvent surgir le jour où quelqu'un obtient une promotion ou une récompense en utilisant les connaissances de quelqu'un d'autre. Afin d'éviter ce phénomène, Google a décidé de *promouvoir* la capacité de collaborer avec les autres. La gestion de la performance dans l'ensemble de l'entreprise est ouverte et transparente. Un système de gestion de la performance distribué et public automatise l'ensemble du processus, nécessite une discussion et fournit des données et un étalonnage. Il interagit avec l'intranet et détermine qui a travaillé avec un employé particulier sur un projet et envoie automatiquement un courriel pour obtenir une rétroaction. C'est le début d'un réseau social avec analyse des interactions. Nous y reviendrons…

Bien que Google soit très fier de son organisation interne, plusieurs observateurs ont remarqué que les récompenses dans le travail créatif et collaboratif échouent généralement sur le long terme. Cela peut parfois provoquer un effet inverse sur le rendement ; surtout si la récompense est utilisée pour inciter les gens à faire des choses qu'ils ne feraient pas autrement [65].

Utiliser la pédagogie, favoriser l'autonomie individuelle et l'autogestion : le cas de l'ILOG.

Il n'existe pas de culture officielle de partage à ILOG, mais les ingénieurs de la R&D ont l'habitude de partager le code source par le biais d'outils de gestion de version comme CVS. *Ne stressez pas les utilisateurs, expliquez-leur qu'ils ne peuvent rien casser*. Grâce au système de gestion des versions qui garde trace de chaque changement, il est toujours possible de revenir à une version précédente. En cas de problème, les utilisateurs peuvent demander de l'aide aux membres de WebCore, l'équipe en charge d'évangéliser le wiki, et on leur apprendra à chaque fois qu'ils peuvent simplement copier et coller des choses à partir d'autres pages. Ils apprennent donc peu à peu jusqu'à ce qu'ils soient autonomes. Aussi, *n'obligez jamais les gens à utiliser le wiki*. Après un certain temps, beaucoup se joignent aux troupes des utilisateurs du wiki car ils n'ont pas pu trouver une solution alternative valable.

Aujourd'hui, cinq ans après l'installation du wiki, en Mars 2006, l'analyse des logs d'utilisation montre que : 700 personnes différentes ont édité une page le mois dernier, il y a peu de pollinisation inter-équipes, les gens éditent surtout des pages dans l'espace de travail de leur équipe sur le wiki, il y a 37 000 pages sur le wiki, avec une augmentation linéaire dans la durée, entre 10% et 20% des utilisateurs contribuent régulièrement à la base documentaire, créant ou éditant des pages, on observe 700 éditions de pages par jour mais sur 50 pages (les mêmes pages sont éditées plusieurs fois, celles qui focalisent l'activité du moment).

Le pourcentage de contributeurs réguliers est faible par rapport au nombre total d'utilisateurs du wiki (lecteurs et écrivains) mais ce nombre augmente chaque mois. C'est un ratio très impressionnant comparé à ce que l'on trouve sur les principaux sites wiki publics comme WikiPedia où les WikiPedians - les contributeurs réguliers -, représentent moins de 0,1% de la communauté des utilisateurs.

Éduquer les gens à partager ! Cela fonctionne avec les étudiants. Le cas du département informatique de l'Université de Nice.

Au département informatique de l'université de Nice, nous avons des pics cycliques d'utilisation du wiki, en fonction de ce que les étudiants font à différents moments de l'année. Nous avons également enfreint la règle utilisée à ILOG : *nous les forçons à utiliser le wiki de manière collaborative* : ils font du travail de groupe sur leurs projets, et toute la gestion de projet doit être faite sur le wiki. Nous avons même mis en place une expérience unique en France sous la forme de projets collaboratifs impliquant des étudiants de différentes universités (même au-delà des frontières). Pendant six mois, des groupes d'étudiants des deux universités mettent à jour le wiki et l'utilisent comme un espace partageable. Les enseignants utilisent le wiki plus régulièrement, mais comme indiqué à la section 2.4.3.2, seuls quelques-uns d'entre eux créent des documents sur le wiki (65%) et la plupart sont des rédacteurs occasionnels. Les réponses au questionnaire ont seulement montré que 5 enseignants sur 23 sont des contributeurs réguliers, y compris le gourou wiki.

Nous avons déclaré plusieurs fois que les étudiants n'avaient pas d'autre choix que d'utiliser le wiki. C'est vrai, mais le questionnaire que nous avons mené avec des étudiants de maîtrise informatique, qui utilisent le wiki depuis trois ans maintenant, montre que tous apprécient le wiki à l'exception de quelques aspects (les pages ne sont pas jolies, la recherche parfois problématique).

2.4.3.7 Utiliser les règles globales d'accès, ne pas laisser les individus protéger leurs documents.

« Comment puis-je protéger mes documents ? » C'est l'une des principales préoccupations des personnes qui découvrent le concept wiki. Après des années d'outils centralisés et structurés pour produire des documents de qualité contrôlée, il est difficile de changer ses habitudes. C.Nahaboo de l'équipe WebCore d'ILOG : "*Heureusement, avec TWiki, les règles d'accès sont plutôt compliquées à utiliser, alors nous dissuadons doucement les gens de les utiliser !* » En effet, le fait de laisser les individus protéger les documents limitera considérablement le partage et la collaboration. De plus, il a été observé que cela amène les utilisateurs souvent à se tirer une balle dans le pied : ils vont surprotéger (en oubliant même comment accéder à leurs propres documents) ou au contraire oublier de protéger les données confidentielles, ou penser qu'ils ont mis les bonnes protections mais ne l'ont pas fait. La solution adoptée par l'équipe WebCore consiste en un système de protection global : il y a quatre zones différentes, chacune exécutant un wiki qui diffère de l'autre par son aspect et sa convivialité. Les utilisateurs passent de façon transparente de l'un à l'autre.

1. **Le wiki de R&D**. Accès global protégé par un pare-feu. Accès gratuit en lecture/écriture pour tous les membres de la division R&D. Pas d'accès pour les autres employés du BITG.
2. **Le Wiki mondial**, accès en lecture pour tous les employés d'ILOG. La plupart des pages ont un accès libre en lecture/écriture. Certaines équipes protègent une partie de leur espace de travail, comme l'équipe "qualité" protège les pages de brouillon mais donne accès en lecture/écriture une fois qu'elles sont d'une qualité suffisante. Ce comportement n'est pas courant.
3. **Le Wiki externe.** Accessible de l'extérieur de l'entreprise. Utilisé pour partager des données avec le monde externe. Certains espaces de travail sont publics (lecture seule), d'autres ont un accès restreint en lecture.
4. **Le Wiki pour les représentants des employés**, situé sur un serveur isolé. Accès restreint pour leur propre usage.

Les plus grands wikis sont les deux premiers de cette liste. Avec les protections globales, les utilisateurs se sentent plus en confiance : les documents de R&D n'ont pas besoin d'un haut niveau de qualité comme c'est le cas des documents situés sur le wiki global (accessible par tous les managers de l'entreprise, alors que le wiki R&D ne l'est pas).

Au département informatique de l'Université de Nice, nous avons deux wikis différents :

1. **L'intranet wiki**, principalement dédié à un usage interne et aux tâches quotidiennes. Celui-ci contient les cours, les projets de l'étudiant, etc. Ce wiki

est également accessible depuis l'extérieur. La plupart des espaces de travail ont un accès complet en lecture/écriture pour tous les utilisateurs enregistrés. Seuls deux espaces de travail ont un accès restreint en lecture : ceux qui peuvent contenir des rapports confidentiels sur les projets ou la période de formation de l'étudiant. C'est le cas lorsque nous avons des partenaires industriels.

2. **Un wiki "officiel"** utilisé pour présenter le département informatique au monde extérieur. Accès en lecture pour le monde entier mais accès en écriture uniquement pour les enseignants. Le *look and feel* de ce wiki est différent ; nous avons retiré beaucoup de choses de l'interface graphique du wiki standard pour qu'il ressemble au maximum à un site web de communication traditionnel.

Les rapports de Google semblent indiquer que leur intranet est en accès en lecture pour tous les employés, avec la possibilité pour n'importe qui de commenter n'importe quelle page du wiki [74].

2.4.4 Structurer un wiki

Le problème numéro un signalé par les personnes interrogées au ILOG et au département CS de l'Université est lié à la structure wiki ouverte qui rend la navigation, l'orientation et la recherche parfois difficiles. Ceci se produit lorsque le wiki atteint plusieurs milliers de pages.

Alors qu'à l'Université de Nice, nous utilisons toujours le moteur de recherche fourni avec le logiciel TWiki, ILOG utilise un moteur de recherche similaire à Google dans son interface utilisateur (aspSeek). De cette façon, toutes les ressources intranet sont indexées et peuvent être recherchées de la même façon. Curieusement, les logs d'utilisation ont montré une chute spectaculaire et après un court laps de temps, les gens ont cessé d'utiliser la fonctionnalité de recherche proposée. Les interviews et les enquêtes ont prouvé que l'hypothèse selon laquelle tout le monde sait comment utiliser Google était fausse. Sur Internet, quand on ne trouve pas quelque chose, c'est acceptable, peut-être que cela n'existe pas. Sur un intranet, les utilisateurs savent parfois que ce qu'ils cherchent est là, et ne comprennent vraiment pas pourquoi ils ne peuvent pas le trouver en utilisant le moteur de recherche. Après de nombreux tests d'utilisabilité, l'interface utilisateur pour le moteur de recherche a été améliorée, mais les gens se plaignent toujours de la difficulté à trouver des choses sur le wiki.

Les entretiens, les journaux de bord et les questionnaires de l'Université de Nice ont confirmé la même chose. La recherche devient de moins en moins utile au fur et à mesure que la base de documents wiki grandit.

La structure et l'organisation était un tel problème au NY Times Digital qu'ils ont dû abandonner le wiki. Ils ont utilisé le moteur wiki développé par Ward Cunningham qui n'incluait pas le concept d'espaces de travail que l'on retrouve dans TWiki ou XWiki par

exemple[13]. Quand le wiki est devenu énorme, avec des milliers de pages, plusieurs problèmes sont apparus…

Quand on crée une page, il faut choisir un nom pour cette page. Les utilisateurs, après deux ans, devaient parfois essayer des douzaines de noms différents avant de trouver un nom qui n'a pas encore été utilisé. L'idée originale avec la collision des WikiNoms était que si vous découvrez qu'il y a une page qui existe déjà avec le même nom ; vous la "rejoindrez" parce qu'elle est supposée être le meilleur endroit pour dire ce que vous avez à dire. Mais ça n'a pas marché au NY Digital : les gens voulaient créer leur propre page !

Ils ont donc inventé de drôles de WikiNoms qui n'avaient plus de sens en rapport avec leur contenu. La navigation et la recherche étaient si difficiles qu'il était presque impossible de trouver un document sans connaître son URL, ou du moins sans l'avoir marqué d'un signet/favori (cela a également été signalé dans certaines interviews chez ILOG). Tout le monde s'est rendu compte que le wiki devenait une masse de pages de plus en plus inaccessibles mais la communauté des utilisateurs n'était pas prête à fournir le travail nécessaire pour refactoriser et organiser tout cela. Le processus de rédaction et de publication dans un journal national est très structuré, et les employés du NY Times Digital n'ont pu obtenir cette auto-structuration des documents dans le wiki par ses utilisateurs. Ce qui semblait être un bel outil prometteur, largement adopté initialement, s'est avéré être une solution défectueuse pour aider le processus de publication. Le wiki n'a pas été complètement abandonné mais il est maintenant utilisé comme bloc-notes de partage, avec tous les travaux sérieux et structurés effectués avec d'autres outils.

En effet, les wikis de cette première époque (1998-2006) ont été conçus pour être structurés par les utilisateurs eux-mêmes. Les utilisateurs diffèrent les uns des autres, chaque individu a sa propre façon de classifier et d'organiser les données, et cela peut changer avec le temps. Une structure hiérarchique comme celle proposée par les espaces de travail est certes une bonne chose d'un point de vue technique mais elle ne donne qu'un " premier coup d'œil " sur la carte mentale d'un intranet wiki. La navigation horizontale (suivant les liens dans la page elle-même) est celle que la plupart des gens utilisent : des tests d'utilisabilité ont montré que la plupart des gens à ILOG ne connaissent pas les noms des différents espaces de travail. Un comportement commun que nous avons remarqué est que les utilisateurs ont commencé à ajouter des mots-clés de catégorie sur les pages wiki : *des tags*. Ces mots-clés sont des WikiNoms qui mènent à des pages qui proposent des hyperliens vers toutes les pages appartenant à la même catégorie. Cette classification naïve aide mais ne passe pas à l'échelle en raison de la complexité, de la richesse et du charme de la nature humaine ! Ce qui fonctionne pour l'un peut ne pas fonctionner pour un autre. Il doit y avoir un moyen d'organiser le chaos !

Dans le cadre d'un projet de recherche mené en collaboration avec ILOG, notre groupe proposa alors d'essayer d'utiliser les *folksonomies* et le marquage social comme un meilleur moyen de catégoriser les documents wiki [78]. L'étiquetage social est un concept

[13] Les espaces de travail de ces wikis permettent de donner un minimum de structure au wiki, ce sont des sortes de « super catégories » dans lesquelles les pages du wiki sont rangées

existant utilisé dans le Web public par des sites populaires tels que del.icio.us et flickr.com. Il est également largement utilisé par les *bloggers* (les utilisateurs de blogs). Vous pouvez annoter vos entrées de blog, les photos que vous avez postées sur flickr en ajoutant des associations de mots-clés à votre travail formant une *quasi-classification à la volée*. Ces balises sont utilisées par les robots web de technorati.com ; et le lien vers votre ressource balisée est ajouté aux autres entrées qui partagent la même balise. Non seulement cela, mais aussi les éléments de del.icio.us et flickr sont également montrés dans la page. L'excitation principale de cette façon de tagger est sa capacité à *socialiser une classification*.

Les gens peuvent toujours utiliser les balises qu'ils estiment représenter le contenu de leur écriture, mais ils peuvent découvrir que cette balise n'a jamais été utilisée auparavant. Il y a donc plus de chances qu'ils choisissent une autre étiquette qui les place dans une classification sociale active que celle qui ne le fait pas. Et il est important d'afficher ces informations au fur et à mesure qu'ils tapent les balises (un mécanisme de complétion automatique devrait aider à éviter les divergences). Voir un nombre gratifiant d'entrées pour cette balise fournit une rétroaction positive. Si l'on veut vraiment choisir une nouvelle balise, pas de problème ! Mais cette balise sera proposée comme un choix lorsqu'une autre personne entrera une balise qui commence par les mêmes lettres, et peut-être que cette personne la choisira à son tour. De cette façon, les utilisateurs, en tant qu'individus, peuvent catégoriser leur écriture comme bon leur semble et donc commencer une taxonomie de base pour utiliser ce qui est en train de devenir un terme populaire.

De nombreuses ressources, et pas seulement les pages wiki, peuvent être balisées dans un intranet : fichiers bureautiques dans un référentiel partagé, listes de diffusion, etc. Et si toutes les ressources sont étiquetées de la même manière, une recherche pour le mot-clé " gestion de logiciel " atteindra aussi bien les pages wiki que les autres ressources. Des choses intéressantes se produisent via notre proposition de lier toutes ces balises au sein d'une folksonomie où les balises sont organisées dans une hiérarchie et reliées les unes aux autres en utilisant des relations comme une relation de synonymie par exemple.

Un outil de supervision peut être proposé à certains administrateurs qui montreront la fréquence d'utilisation des balises, mettront en évidence les nouvelles balises qui ne sont toujours pas liées à une catégorie, etc. Les étiquettes peuvent être liées si elles apparaissent fréquemment ensemble, etc. Nous croyons que l'étiquetage social minimise les coûts et maximise la participation des utilisateurs, tout en aidant à bâtir une meilleure folksonomie et à améliorer les intranets des utilisateurs. La folksonomie est le concept qui relie les étiquettes entre elles. Elles peuvent être décrites à l'aide de la technologie du web sémantique et outillées par des moteurs de recherche dédiés.

2.5 Conclusion

Nous avons présenté une synthèse de deux expériences que nous avons eues sur plusieurs années avec les wikis dans des intranets (département informatique de l'Université de Nice et ILOG). Ces recherches ont été menées dans deux organismes français. Les

facteurs contextuels, y compris les normes et la culture, sont susceptibles de différer de manière significative entre les organisations et même au sein des organisations et des pays, et il faut se demander dans quelle mesure ces résultats peuvent être généralisés. C'est la raison pour laquelle nous avons essayé de corréler nos mesures avec d'autres rapports publiés sur d'autres intranets significatifs articulés autour d'un wiki (Google, Motorola, le NY Times Digital, Nokia).

Nous avons montré que l'utilisation d'outils collaboratifs comme les wikis aide au partage des connaissances et à la créativité. Cependant, nous avons fait remarquer que la solution au problème de *mieux travailler ensemble* ne peut se réduire à un outil ou à un ensemble d'outils : plusieurs conditions préalables sont nécessaires à l'apparition de la magie de groupe.

Il est difficile de mettre en place un wiki et de faire en sorte que les gens l'utilisent. L'aide d'un gourou local ou d'entreprises privées spécialisées est nécessaire, des compétences techniques pour l'installation et la maintenance du logiciel également, mais il est surtout très important de prendre en compte les aspects sociaux de la participation des utilisateurs. Nous avons essayé de comprendre pourquoi certains d'entre eux sont réticents à utiliser le wiki et pourquoi d'autres l'adoptent sans difficulté. Nous avons également examiné l'utilisation du wiki et avons remarqué qu'il est principalement utilisé pour *la gestion de projet informelle* en plus d'être *un bloc-notes partagé sur le Web*. Dans des entreprises comme ILOG ou Motorola, des personnes ont développé de nombreuses petites applications de type Excel, alors que cette fonctionnalité n'a jamais été utilisée par d'autres.

Les wikis ont rencontré quelques problèmes de passage à l'échelle dans nos expériences, rendant la navigation et les tâches de recherche problématiques. Nous avons proposé une piste basée sur l'utilisation de folksonomies : utiliser le marquage social comme moyen de classer les documents wiki dans une approche orientée utilisateur qui correspond à la culture wiki.

Nous avons démarré en 2006 la formalisation (conception, modèles, ontologies, idées directrices) de SweetWiki, un moteur wiki sémantique expérimental qui supporte ces folksonomies. Il fait l'objet de la section suivante de ce manuscrit.

3. Wikis sémantiques (mes travaux sur la période 2006-2010)

Contexte

En 2006, nous sommes en plein "Web 2.0", dont les interactions des utilisateurs et les réseaux sociaux forment la pierre angulaire. Les applications de ce "Web 2.0" s'exécutent généralement dans un navigateur Web, proposent du contenu dynamique avec des interfaces utilisateur riches -nouveauté pour l'époque-, offrent des moyens d'ajouter ou de modifier facilement le contenu du site Web auquel elles appartiennent et possèdent déjà les aspects fondamentaux des réseaux sociaux tels que nous les connaissons aujourd'hui. Les applications bien connues qui ont contribué à la diffusion du "Web 2.0" sont les blogs, les wikis et les sites de partage d'images / vidéos. Ces services pionniers (YouTube, Flicker, Blogger, et des sites d'hébergement de wikis aujourd'hui disparus) ont considérablement augmenté le partage et la participation parmi les utilisateurs du Web. Il est devenu possible de construire des connaissances en utilisant des outils qui peuvent aider à analyser le comportement des utilisateurs en coulisses : *ce qu'ils font, ce qu'ils savent, ce qu'ils veulent, ce qui les intéresse*. Ces outils qui permettent de partager ces connaissances à travers un réseau, et qui peuvent raisonner sur ces connaissances, conduiront théoriquement les utilisateurs à mieux utiliser les connaissances disponibles, c'est-à-dire, devenir des utilisateurs "plus intelligents" (on sait aujourd'hui qu'une des principales utilisations de cette collecte de traces a surtout servi à la publicité et aux systèmes de recommandations). Wikipédia, un exemple extrêmement réussi de technologie web, a aidé le partage des connaissances en laissant les individus créer et modifier librement son contenu. Mais Wikipédia était conçu à l'époque *par et pour les humains* - le logiciel ne pouvait pas comprendre et raisonner sur le contenu qu'il hébergeait. Parallèlement, le « Web Sémantique », un ensemble de technologies favorisant le partage des connaissances sur le Web entre différentes applications, a commencé à attirer l'attention. Les chercheurs ont commencé, vers 2005, à travailler sur le concept de « wiki sémantique », mélangeant les avantages du wiki présentés dans les sections précédentes de ce manuscrit, et les technologies du web sémantique.

Contributions

Entre 2006 et 2010 j'ai publié un état de l'art des wikis sémantiques, et j'ai et co-conçu avec Fabien Gandon et implémenté SweetWiki, un wiki sémantique réconciliant deux tendances émergentes à l'époque : un Web sémantiquement augmenté et un réseau d'applications sociales où chaque utilisateur est un fournisseur actif ainsi qu'un consommateur d'informations. L'article « SweetWiki : a semantic wiki » paru dans le « Journal of Web Semantics » est devenu une référence et fait l'objet de très nombreuses citations.

SweetWiki faisait un usage intensif des concepts et des langages du Web Sémantique, et a démontré comment l'utilisation de tels paradigmes pouvait améliorer la navigation, la recherche et la facilité d'utilisation. Il a été longtemps utilisé par des entreprises comme système de documentation riche. (Conférences WikiSym 2006, ISWC 2008, Journal ISWC 2008). Il a fait l'objet d'un dépôt à l'Agence de Protection des Programmes (APP), j'en suis le principal contributeur.

Collaborations, encadrement

Le logiciel SweetWiki était un des livrables du projet européen PALETTE, qui visait à faciliter et accroître l'apprentissage individuel et organisationnel dans les communautés de pratique. Par ailleurs il a été utilisé dans le cadre du projet ePrep (rayonnement et développement des classes préparatoires aux grandes écoles, collaboration avec Nathalie Van de Wiele), et intégré dans le système d'information de la société Robosoft. J'ai co-encadré le stage de Master Recherche de Guillaume Ereteo, qui a débouché sur une thèse sur l'analyse sémantique des réseaux sociaux.

J'ai été le « guru wiki » lors de l'implantation de SweetWiki dans le réseau ePrep (avec Adil El Ghali) et lors de son implantation dans la société Robosoft. J'ai supervisé la création des ontologies de domaine pour ces deux initiatives.

Publications principales ayant résulté de ces travaux

Adil El Ghali, Amira Tifous, Michel Buffa, Alain Giboin, Fabien Gandon, and Rose Dieng-Kuntz. *"Using Sweetwiki in Communities of Practice"*. "Web-Based Learning Solutions for Communities of Practice", book chapter. IGI Global, 2009.

Michel Buffa, Fabien Gandon, Guillaume Ereteo, Peter Sander and Catherine Faron, *SweetWiki: A semantic wiki*, Special Issue of the Journal of Web Semantics on Semantic Web and Web 2.0, Volume 6, Issue 1, February 2008 , Edited by Mark Greaves and Peter Mika, Elsevier, Pages 84-97 PDF: http://smtp.websemanticsjournal.org/index.php/ps/article/viewFile/138/136

Michel Buffa, Guillaume Erétéo, Fabien Gandon, *A Wiki on the Semantic Web*, book chapter in "Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications", Editors: Jorg Rech, Bjorn Decker, Eric Ras, Information Science Reference, ISBN: 9781599048772, p115-137, 2008

El Ghali, A., Tifous, A., Buffa, M., Giboin, A., Dieng-Kuntz, R. (2007). *Using a Semantic Wiki in Communities of Practice*, In Proceedings of the 2nd International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice TEL-CoPs'07, EC-TEL'07, Sissi, Lassithi - Crete Greece, 18 September, 2007, pages 22-31.

Michel Buffa, Fabien Gandon, SweetWiki : *Semantic Web Enabled Technologies in Wiki*, WikiSym, IEEE International Symposium on Wikis, August 21-23, 2006, Odense, Denmark. Accept rate 31%. PDF : http://www.wikisym.org/ws2006/proceedings/p135.pdf

Michel Buffa, Gael Crova, Fabien Gandon, Claire Lecompte, Jeremy Passeron, SweetWiki : *Semantic WEb Enabled Technologies in Wiki*, 3rd Annual European Semantic Web Conference (ESWC), pp. 74-88, Budva Montenegro, 11-14 June 2006

Résumé de cette section

La "révolution wiki" a commencé en 1995 lorsque Ward Cunningham a écrit le premier wiki pour le Portland Pattern Repository. Fatigué du modèle centralisé de publication

web, en cours à l'époque, de la complexité du langage HTML pour les non-initiés (il n'existait pas d'éditeurs de pages HTML WYSIWYG à l'époque), et influencé par Hypercard et par la vision initiale du Web, il a créé un site web où les gens pouvaient créer, modifier, refactoriser et lier des pages à partir de leur navigateur Web, d'une manière très simple - un clic. Au lieu de HTML il a proposé un langage de balisage épuré (WikiML) inspiré de la manière dont les gens formaient des messages texte quand Internet n'était pas encore multimédia. La plus grande contribution de Cunningham de notre point de vue a été est l'invention de WikiWords comme un moyen de créer des hyperliens, même pointant vers des pages non encore créées. Un WikiWord, c'est une expression dont chaque mot commence par une lettre majuscule et dont on a enlevé les espaces : `CeciEstUnWikiWord`. Entrez au clavier un WikiWord (par exemple `NewPage`) et il sera enregistré en tant que lien vers une page dont l'URL se termine par ce WikiWord. Si la page n'existe pas, en cliquant sur le lien, cela créera la page à la volée, vide de contenu. Le mot "wiki" signifie "rapide" dans la langue hawaïenne et évoque ce nouveau moyen rapide et simple pour créer et éditer des sites Web. Les personnes non-techniques pouvaient enfin publier du contenu sans difficulté et les wikis ont commencé à croître de façon exponentielle. De nos jours, peut-être le plus célèbre exemple d'un wiki public est l'encyclopédie Wikipédia.

Cependant, des études ont montré que l'acceptation de cette politique documentaire ouverte, de ces outils collaboratifs peu structurés, n'est pas automatique pour la plupart des intranets ou sites communautaires. Il y a différentes raisons pour expliquer ce rejet, même partiel, y compris des raisons sociales. Par exemple, la culture d'entreprise peut ne pas être adaptée, ou bien le rejet des wikis peut venir de problèmes d'utilisabilité, par exemple, le wiki n'est pas assez structuré, il est difficile de naviguer et de trouver des informations pertinentes lorsque le nombre de pages wiki augmente, le langage de balisage wiki utilisé par la plupart des moteurs wiki rendent les gens réticents à contribuer, etc. (voir [24], [25] et [26]). La plupart des moteurs de wiki, le logiciel derrière les sites wiki, ont été conçus dans le milieu des années quatre-vingt-dix en exploitant les technologies web de l'époque, à savoir, principalement HTML, HTTP, CGI et URI. Inévitablement, les wikis ont développé des langages de balisage, variantes de WikiML, et il n'y a pas eu de WikiML standard malgré les efforts de normalisation (projet CREOLE et projet WIF -Wiki Interchange Format [27])

Notre idée d'un wiki sémantique était celle d'un moteur wiki qui utilise les technologies du Web sémantique pour intégrer des connaissances formalisées, du contenu structuré et des (hyper)liens typés, dans les pages wiki. La connaissance formalisée est représentée par des frameworks du web sémantique et donc accessible et réutilisable par les applications web. Dans le wiki, cette connaissance peut être utilisée pour proposer des fonctionnalités améliorées telles qu'un meilleur moteur de recherche, de meilleures suggestions vers de nouveaux documents ou personnes reliés au document courant. Pour simplifier, un wiki sémantique apporte les moyens pour capturer et exploiter des connaissances de manière très simple et souvent invisible, et rend simple par la nature même du wiki, les mise à jour de contenu dynamique, les notifications, etc.

Les wikis sémantiques de l'époque (2006-2010) étaient soit construits sur des moteurs de wiki existants comme MediaWiki, et proposaient à ces moteurs des extensions pour le web sémantique, ou ont été créés à partir de zéro avec les technologies web sémantiques au cœur de leur développement.

On trouve des wikis sémantiques spécialisés dans l'édition coopérative d'ontologies, mais d'autres utilisent les ontologies comme référence pour l'annotation du contenu du wiki, et certains font les deux. Certains wikis utilisent des éditeurs spécialisés pour le contenu sémantique, et certains utilisent des annotations à même le texte pour ajouter de la sémantique. Certains wikis sémantiques intègrent un moteur de raisonnement, peuvent exporter les annotations ou les ontologies définies dans le wiki en RDF ou RDFS / OWL, et laissent les utilisateurs/administrateurs lier ces données sémantiques à un moteur de raisonnement externe. En d'autres termes, la communauté des wikis sémantiques à l'époque explorait les multiples points de jonction entre le web 2.0, les spécificités des wikis et les capacités des frameworks du web sémantique.

Dans ce chapitre nous donnerons un état de l'art des wikis sémantiques de la vague 2005-2014. Par la suite nous détaillerons mes contributions en examinant les caractéristiques de SweetWiki [34, 35], dont certaines étaient originales à l'époque.

Enfin, nous reviendrons sur certaines caractéristiques de SweetWiki. En effet, ce dernier contenait les prémisses de ce qu'on appellera plus tard les "wikis d'applications" et qui font l'objet des prochains chapitres de ce manuscrit. Sa grande originalité était d'être conçu autour d'une "*ontologie du wiki*" lui-même. Les concepts du wiki, et ses fonctionnalités de base étaient décrites en RDF/RDFs et rendues opérationnelles notamment à l'aide de requêtes SPARQL. En ayant codé de zéro un wiki conçu avec les technologies du Web Sémantique au cœur, il était naturel qu'il utilise ces technologies pour ajouter et exploiter de la connaissance au sein des documents qu'il produisait. SweetWiki, novateur sur de multiples aspects, a fait l'objet d'un dépôt à l'APP. Par ailleurs, SweetWiki exploitait en profondeur des standards du W3C encore émergents à l'époque (documents dynamiques, éditeurs WYSIWYG etc.)

SweetWiki intégrait en 2005-2006 :

- Un éditeur WYSIWYG, pour l'édition de documents et pour l'annotation,
- Un système de tags suggérés avec des mécanismes d'auto-complétion,
- Un éditeur d'ontologie permettant notamment d'éditer et de hiérarchiser la folksonomie composée par les tags entrés librement par les utilisateurs,
- Le support d'ontologies externes (on pouvait en rajouter interactivement, elles devenaient disponibles pour l'annotation),
- Un moteur de recherche sémantique (CORESE),
- Un éditeur de requêtes SPARQL qui permettait d'enrichir les pages dynamiquement avec des résultats de requêtes,
- Une gestion des centres d'intérêts des utilisateurs.

Enfin, un état de l'art sur la période 2014-2018 complètera l'étude des wikis sémantiques.

3.1 Introduction

Tout le monde s'accorde à dire que les interactions des utilisateurs et les réseaux sociaux sont parmi les pierres angulaires du "Web 2.0". Les applications Web 2.0 s'exécutent généralement dans un navigateur Web, proposent un contenu dynamique avec des interfaces utilisateur riches, offrent des moyens d'ajouter ou d'éditer facilement le contenu du site Web auquel elles appartiennent et présentent des aspects de réseau social. Les applications bien connues qui ont contribué à la diffusion du Web 2.0 sont les blogs, les wikis et les sites de partage d'images et de vidéos ; ces applications ont considérablement augmenté le partage et la participation parmi les utilisateurs du Web. En outre, il est possible de construire des connaissances à l'aide d'outils qui peuvent aider à analyser le comportement des utilisateurs en coulisses : ce qu'ils font, ce qu'ils savent, ce qu'ils veulent. Les outils qui aident à partager ces connaissances à travers un réseau, et qui peuvent raisonner en fonction de ces connaissances, mèneront à des utilisateurs qui pourront mieux utiliser les connaissances disponibles. Wikipédia, un exemple très réussi de technologie Web, a par exemple contribué au partage des connaissances en permettant aux individus de créer et de modifier librement son contenu. Mais Wikipédia est conçu pour les gens - les logiciels ne peuvent pas directement comprendre et raisonner sur le son contenu. Parallèlement, le "web sémantique", un ensemble de technologies qui facilitent le partage des connaissances sur le web entre différentes applications, commençait à gagner en popularité et rapidement, en 2005, des chercheurs ont commencé à travailler sur le concept d'un "wiki sémantique", mélangeant les avantages du wiki et les technologies du web sémantique. Dans cette partie du manuscrit, nous rappellerons un état de l'art des wikis sémantiques en 2008, et nous présenterons SweetWiki, l'exemple d'une application conciliant deux tendances du web émergent à l'époque : un web sémantiquement augmenté et un web d'applications sociales où chaque utilisateur est un fournisseur actif ainsi qu'un consommateur d'informations. SweetWiki faisait un usage intensif des concepts et des langages du web sémantique, et a démontré comment l'utilisation de tels paradigmes pouvait améliorer la navigation, la recherche et la facilité d'utilisation. Il a été un des wikis sémantiques majeurs à l'époque de sa publication.

3.2 Définition d'un wiki sémantique

Un wiki sémantique est « un wiki qui qui utilise les technologies du Web sémantique pour créer / éditer / utiliser des connaissances formalisées et qui sont représentées à l'aide de frameworks du web sémantique et donc accessibles et réutilisables par des applications Web ».

Au sein du wiki, ces connaissances peuvent être utilisées pour proposer des fonctionnalités améliorées telles qu'un meilleur outil de recherche, un outil pour suggérer de nouveaux liens, pour aider à l'identification des réseaux de personnes partageant les mêmes intérêts, pour générer du contenu dynamique en embarquant des requêtes dans les pages, pour la vérification et la notification, etc.

Les wikis sémantiques existants en 2006 étaient soit construits sur des moteurs wiki existants et proposaient des extensions pour le web sémantique, soit avaient été créés à partir de zéro avec les technologies du web sémantique à l'esprit.

Certains wikis étaient dédiés à l'édition coopérative d'ontologies, d'autres utilisaient les ontologies comme référence pour annoter le contenu des pages du wiki, une petite partie faisaient les deux. Certains wikis utilisaient des éditeurs spécialisés pour le contenu sémantique, et certains autres utilisaient des balises WikiML spéciales pour ajouter une annotation à même le texte. Certains wikis sémantiques intégraient un moteur de raisonnement, certains pouvaient exporter les annotations ou les ontologies définies dans le wiki en RDF ou en RDFS/OWL et laissaient les utilisateurs établir un lien vers un moteur de raisonnement externe. En d'autres termes, la communauté wiki sémantique explorait les multiples points de jonction entre les aspects « wiki » du web 2.0 et les capacités des frameworks web sémantiques de l'époque. La section suivante examine et compare les différentes contributions dans ce domaine.

3.3 Wikis sémantiques : définition et état de l'art en 2008

De nombreux wikis sémantiques étaient en cours de développement, et nous nous sommes concentrés sur ceux liés à des activités de recherche scientifiques publiées. Nous n'avons pas considéré ceux, comme OmegaWiki, qui sont moins pertinents pour la recherche académique dans le domaine du web sémantique. Alors que de nombreux moteurs wiki présentés dans cette section sont des prototypes fonctionnels, certains, comme par exemple Semantic Media Wiki, ont déjà été déployés dans des applications à grande échelle. Ceux-ci, basés sur des moteurs wiki existants bien connus comme JspWiki ou MediaWiki, s'appuient sur la stabilité, la performance et la robustesse de ces moteurs. De plus, l'état de l'art présenté ici ne peut être qu'un instantané des contributions et des versions disponibles au moment de la rédaction, en 2008, il sera complété dans le chapitre suivant, dédié aux « wikis sémantiques d'application ».

En 2008, en regardant l'état de l'art, on pouvait distinguer entre les approches considérant "*l'utilisation des wikis pour les ontologies*" et les approches considérant "*l'utilisation des ontologies pour les wikis*" (peu de moteurs fusionnant les deux approches).

La plupart des projets en cours sur les wikis sémantiques entrent dans la première catégorie, c'est-à-dire qu'ils considèrent les pages wiki comme des concepts et les liens saisis (dans le contenu de la page) comme des propriétés d'objet ou des propriétés de données. Dans ce modèle, appelé « *wikitologie* » [79], le wiki devient le front-end du système de maintenance ontologique.

L'un des premiers wikis à tomber dans cette catégorie est Platypus [80] qui impose d'éditer séparément les métadonnées de chaque page wiki dans une "page de métadonnées" associée. Il supporte l'édition d'ontologie mais sans vérification de cohérence entre les annotations et l'ontologie. Il n'est pas fourni avec un moteur de raisonnement et ne prend en charge que les requêtes de base. Les métadonnées sémantiques sont utilisées pour améliorer la navigation, mais les utilisateurs doivent passer de l'édition de texte normal à

l'édition d'annotations sémantiques, car ces activités sont effectuées à l'aide de deux éditeurs de texte distincts. D'autres wikis comme SHAWN [81] offrent des fonctionnalités similaires. Certains wikis dans cette catégorie adressent les défauts de Platypus en permettant l'insertion d'annotations sémantiques directement dans le texte de la page, généralement sous forme de liens typés.

Rise [82] appartient également à la première catégorie : l'ontologie utilisée par la communauté est éditée via le Wiki lui-même et un ensemble de conventions de nommage est utilisé pour générer automatiquement l'ontologie réelle à partir du contenu du Wiki. Un langage propriétaire est utilisé pour décrire les métadonnées et l'exportation RDF est possible. Les informations sémantiques sont utilisées pour la navigation et les contrôles de cohérence. L'ontologie est reconstruite chaque nuit pour tenir compte des pages wiki ajoutées et modifiées.

Rhizome [83] supporte une version modifiée de WikiML (ZML) qui utilise des conventions de formatage spéciales pour rendre les propriétés sémantiques directement explicites dans le contenu de la page. Les pages sont sauvegardées en RDF et un autre éditeur peut être utilisé pour modifier directement le RDF. Les auteurs de Rhizome admettent que cette fonctionnalité est dangereuse car on peut compromettre le bon fonctionnement du wiki en sauvegardant du RDF incorrect. Pour atténuer les dangers inhérents à ce niveau d'ouverture, Rhizome fournit une gestion des autorisations et une validation à grain fin en plus de l'utilisation de contextes. La manière dont les métadonnées améliorent le comportement du wiki n'est pas très explicite dans les articles décrivant ce wiki : Rhizome ne fournit pas de recherche avancée et ne propose pas de mécanismes d'aide basé sur les métadonnées pour naviguer sur le wiki, en tout cas dans sa version de 2006. RDF-Wiki[14] est similaire à Rhizome en ce qu'il permet les annotations RDF pour le traitement externe.

Semantic Media Wiki [84] est basé sur MediaWiki. Contrairement à Rise, les liens typés peuvent également être utilisés pour spécifier les attributs de la page. Par exemple, le texte suivant :

```
San Diego is a [[is a::city]] located in the south-western
corner of [[is located in::California]]
```

…établit les faits « San Diego est une ville » et « San Diego est situé en Californie ». Tandis que le texte :

```
Its coordinates are [[coordinates:=32°42'54"N, 117°09'45"W]]
```

…définit un attribut nommé "coordinates". Ces données sont utilisées pour la navigation par facettes. Semantic Media Wiki traduit ces métadonnées en RDF et la prise en charge du moteur de raisonnement KAON2 existe en tant que preuve de concept. D'autres extensions sémantiques de MediaWiki sont disponibles [85] et permettent d'exploiter les métadonnées sous des formes variées (génération de cartes, formulaires de saisie, etc.)

[14] RDF Wiki: http://infomesh.net/2001/05/sw/#rdfwiki

Makna [86] est basé sur JSPWiki et fournit des extensions sémantiques sous forme de liens typés. Il est livré avec le moteur de raisonnement JENA qui permet la prise en charge de requêtes SPARQL. Son éditeur de texte propose des formulaires HTML supplémentaires (avec auto-complétion) pour interroger le moteur sémantique et rechercher des concepts/propriétés/relations. Ceci est utile dans le cas d'une grande ontologie.

WikSar [87] permet aux utilisateurs d'entrer des annotations sémantiques à partir de l'éditeur de texte wiki en utilisant des WikiWords. Par exemple : si une page nommée "PrinceHamlet" contient une ligne :

```
FigureBy : WilliamShakespeare
```

…elle peut être considérée comme une déclaration RDF indiquant que la figure du Prince Hamlet a été créée par William Shakespeare. En énumérant les primitives utilisées dans toutes ces déclarations intégrées, une ontologie peut être extraite du contenu du Wiki. L'éditeur est basé sur le texte et ne propose aucune aide d'aucune sorte à l'utilisateur ni aucun contrôle de cohérence. Au fur et à mesure que les pages sont sauvegardées, les métadonnées sont utilisées pour proposer une navigation à facettes. WikSar prend en charge les requêtes en RDQL et SPARQL, et les requêtes peuvent être intégrées dans des pages wiki ou dans des pages template (des modèles réutilisables). Une caractéristique distinctive de WikSar est l'outil de visualisation graphique et de navigation qui peut être utilisé pour explorer le wiki à travers son réseau de métadonnées.

Les liens typés sont puissants, mais il faut se souvenir de chaque concept, relation et propriété avant de le saisir, ce qui n'est pas pratique. Kuhn [168] va plus loin avec AceWiki[15]. En permettant d'ajouter et modifier des phrases écrites en utilisant le langage ACE (Attempt to Controlled English[16]), grâce à l'utilisation d'un éditeur interactif. L'éditeur est conscient de l'ontologie de fond et fournit des conseils à l'utilisateur en ne proposant que des compléments valides. De plus, l'éditeur peut être utilisé pour étendre l'ontologie en créant de nouveaux concepts, rôles et individus. AceWiki est donc aussi, de facto, un éditeur d'ontologie.

La deuxième famille d'approches se concentre sur "*l'utilisation des ontologies pour les wikis*". IkeWiki [88][89] prend en charge l'édition WYSIWYG du contenu des pages et des métadonnées, ainsi que le balisage des pages. L'éditeur est livré avec des fonctionnalités dynamiques comme l'auto-complétion sur les métadonnées. Il nécessite le chargement d'une ontologie existante. Un certain soutien pour l'édition d'ontologie est fourni. Il utilise Jena comme triple store et comme moteur SPARQL, et les métadonnées sont utilisées pour la navigation, la recherche et le rendu des pages. Les annotations peuvent être visualisées dans un cadre à côté ou en bas de la page wiki (un cartouche rassemblant les métadonnées, similaire aux *infotables* que l'on trouve sur les pages Wikipédia). SWIM [169], un wiki sémantique pour la gestion des connaissances

[15] AceWiki: http://gopubmed.biotec.tu-dresden.de/AceWiki/
[16] Attempt to Controlled English (ACE)

mathématiques est également basé sur une version plus ancienne du moteur d'IkeWiki, mais utilise OMDoc pour décrire les pages et leurs annotations.

SweetWiki fait également partie de cette deuxième catégorie. Sa caractéristique la plus originale est l'utilisation d'un "modèle d'objet wiki", une ontologie du wiki lui-même, qui décrit le modèle de données du système afin que le wiki puisse être interrogé dans ses propres documents. Par exemple, des requêtes telles que "*montrer les pages et les vidéos qui parlent de ce sujet*" sont rendues possibles.

Il soutient les concepts de marquage social et de folksonomie[17] en plus des ontologies externes. SweetWiki et WikSar partagent de nombreuses fonctionnalités comme la construction d'ontologies légères à la volée, les requêtes intégrées dans les pages wiki, l'édition des métadonnées et du contenu des pages dans le même éditeur. SweetWiki ajoute un moteur de raisonnement et un éditeur WYSIWYG extensible pour le contenu et les métadonnées, (comme IkeWiki ou Makna). L'éditeur de SweetWiki est amélioré par l'utilisation de la technologie AJAX (émergente à l'époque) : l'annotation des pages conduit à une gratification instantanée au fur et à mesure que l'utilisateur entre du texte ou des tags : (a) les utilisateurs peuvent voir un affichage instantané des liens facettés que l'annotation ajoutera à la page et (b) un mécanisme d'auto-complétion suggère les concepts existants dans l'ontologie, les catégories connexes et le nombre de pages partageant cette annotation. Ce comportement réactif est une incitation à réutiliser les balises existantes et montre le gain obtenu instantanément. En outre, SweetWiki est livré avec des outils complets et conviviaux de supervision et d'édition de l'ontologie. Cependant, SweetWiki n'est pas un wiki dédié à la gestion collaborative d'ontologies comme peuvent l'être OntoWiki [90] ou POWL [91]

En regardant le résumé proposé dans le tableau 1 page suivante, nous pouvons remarquer que la plupart des moteurs wiki sémantiques sont open source, que seuls quelques-uns d'entre eux utilisent des interfaces utilisateur riches ou intègrent un moteur de raisonnement.

Dans les sections suivantes, nous présenterons l'architecture de SweetWiki et détaillerons certaines de ses principales caractéristiques.

[17] Une folksonomie peut être considérée comme un vocabulaire commun qui est à la fois originaire et familier à ses principaux utilisateurs. Dans SweetWiki il est encodé comme une ontologie qui se développe en tant qu'utilisateurs tag pages, images, vidéos, etc. Il peut être édité à l'aide de l'éditeur intégré.

	Platypus	Shawn	IkeWiki	Rise	Rhizome	SeMediaWiki	Makna	WikSar	AceWiki	SweetWiki	SWiM	OntoWiki	POWL
reuses an existing engine	no	no	no	no	no	MediaWiki	JspWiki	no	no	no	IkeWiki	MediaWiki	no
wiki object model	no	no	no	no	no	no	no	no	no	yes	yes	no	no
annotations in content	no	yes	no	yes	yes or edited separately	yes	yes	yes	yes	yes	no	NA	NA
separated editor for annotations	yes	no	For typed links	no	yes	no	no but external editor for statements	no	no	no	no	NA	NA
assisted annotations	yes	no	yes	no	no	no	yes	no	yes	yes	yes	no	no
social tagging	no	no	yes	no	no	no	no	no	no	yes	yes	no	no
pages are classes	no	all	some	all	all	all	all	all	all	no	some	all	all
ontology edition features	basic RDF editing, text based, relations edited via the wiki medatada page	typed links	yes, some support	typed links	using ZML	typed links	typed links + statements	typed links	yes, in ACE	embedded ontology editor	yes, some support	yes	yes
representations languages	RDF/S & OWL	proprietary	OWL	proprietary, RDF export possible	RDF/S + ZML	RDFS	RDF/S, OWL	RDFS	ACE	RDFa, RDF/S, OWL lite	RDF/S, OWL	RDFS	RDFS/OWL
loading saving ontologies	yes	RDF export	yes	ontology is exported nightly	yes	export RDF	yes	export RDFS	no	yes	yes	export RDFS	RDFS/OWL export
queries	no	no	SPARQL	basic	no	WikiML extension	no	wikiML, RDQL, SPARQL	no	SPARQL	SPARQL	no	RDQL
reasoning engine	no	no	Jena	no	no	external (Kaon)	Jena	no	no	Corese	Jena	no	no
consistency check	no	basic	no	no	no	no	no	no	no	partial	no	no	no
versioning for pages	yes	no	yes	no	yes	yes	yes	no	no	yes	yes	yes	yes
versioning for metadata	no	no	no	no	through contexts	within page	within page	no	no	within page	no	yes	yes
how metadata are exploited	navigate	navigation	navigate, render, search	for exporting an ontology	navigate, render, search	navigation, search	navigate	navigate, render	defining concepts	all wiki features render navigate,	navigate, render	to build an ontology	to build an ontology
WYSIWYG editor	no	no	yes	no	no	no	no	no	yes	yes	no	no	HTML only
Persistence	text	text	DB	text + DB	DB	DB	text + DB	text	text	XHTML + RDFa	DB	DB	DB
Discussion pages	no	no	yes	no	yes	yes	no	no	no	no	no	yes	no
Security/access rights	no	no	yes	yes	yes	yes	yes	no	no	yes	yes	yes	yes
Notification	no	no	rss	no	no	yes	yes	no	no	not yet	rss	yes	no
AJAX-based GUI	no	no	yes	no	no	no	yes	no	yes	yes	no	no	no
programming language	java	perl	java	java	python	PHP	java	perl	java	java	java	PHP	PHP
Licence	GPL	N/A	GPL	commercial	GPL	GPL	GPL	from author	N/A	LGPL like	GPL	GPL	GPL

Figure 15. Tableau comparatif des principaux Wikis sémantiques en 2008. (extrait de notre article « Semantic Wikis », Journal of Web Semantics 2008)

3.4 SweetWiki : un des premiers wikis sémantiques

L'idée de SweetWiki est de revisiter la logique de conception des wikis (qui date de 1995), en tenant compte de la richesse des nouvelles normes disponibles pour le web onze ans plus tard afin de combler certaines des lacunes identifiées par l'expérience. SweetWiki s'appuie sur les standards du web pour le format de page wiki (XHTML), pour les macros incluses dans les pages (balises JSPX/XML), pour les annotations sémantiques (RDFa, RDF), pour les ontologies qu'il manipule (OWL Lite), etc. Il améliore l'accès à l'information grâce à une navigation par facettes, à des outils de recherche et de sensibilisation améliorés, à l'identification des réseaux de connaissances, etc. Il fournit également un éditeur WYSIWYG unique pour l'édition de métadonnées et de contenu, avec des outils d'annotation assistée (auto-complétion, vérificateurs pour les requêtes ou annotations intégrées). Il est livré avec un éditeur d'ontologie et un moteur de raisonnement. Un autre point intéressant est que SweetWiki permet aux métadonnées d'être extraites et exploitées par d'autres applications.

En annotant sémantiquement les ressources du wiki et en réifiant le modèle d'objet wiki lui-même, SweetWiki fournit des capacités de raisonnement et d'interrogation. Tous les modèles sont définis dans des schémas OWL capturant les concepts des wikis (WikiWord, page wiki, lien avant et arrière, auteur, etc.) et les concepts manipulés par les utilisateurs (folksonomie des utilisateurs, ontologies externes). Ces ontologies sont exploitées par un moteur de recherche sémantique intégré (CORESE [92]) nous permettant de prendre en charge le cycle de vie du wiki, par exemple en refactorant les pages (renommage, déplacement, restructuration du contenu), nous permettant de proposer de nouvelles fonctionnalités telles que : la recherche sémantique, la veille et la notification basées sur le profil utilisateur, et aussi supporter des extensions, comme par exemple le support de nouveaux médias ou l'intégration de formats de documents en importation (Microsoft Office etc.).

Dans SweetWiki, nous avons accordé une attention particulière à préserver l'essence d'un wiki, en particulier la simplicité et la dimension sociale. Ainsi, SweetWiki supporte toutes les fonctionnalités communes du wiki telles que les liens de page faciles à l'aide de WikiWords, la gestion de versions sur les pages, etc., mais il innove aussi en intégrant un éditeur WYSIWYG étendu pour supporter les fonctionnalités de balisage social, la possibilité d'intégrer des requêtes SPARQL dans les pages sans quitter l'éditeur, de les valider etc., masquant l'implémentation des annotations basée sur OWL lite. Les utilisateurs peuvent librement entrer des balises/tags et un mécanisme d'auto-complétion suggère des balises existantes en émettant des requêtes pour identifier les concepts existants avec des étiquettes compatibles. Ainsi, le balisage est à la fois facile et motivant (affichage en temps réel du nombre de pages associées) et les concepts sont collectés dans des folksonomies. Les pages Wiki sont stockées directement en XHTML ou au format JSPX, intégrant des annotations sémantiques (RDFa et GRDDL) prêtes à être réutilisées par d'autres logiciels.

Dans la section suivante nous détaillons l'architecture de SweetWiki, dans la section 3.5 nous nous concentrerons sur la façon dont nous avons mis en œuvre le support du

marquage social dans SweetWiki, y compris les améliorations de l'éditeur WYSIWYG ainsi qu'un éditeur d'ontologie web intégré qui peut être utilisé pour éditer, créer et gérer les ontologies, notamment la folksonomie construite sur les tags. Dans les section 3.6 et 3.7, nous montrons comment le framework ayant servi à la construction de SweetWiki le transforme en un des premiers *"wiki d'application"* : un wiki qui permet aux utilisateurs d'écrire de petites applications dans les pages wiki. Nous concluons à la section 3.8 par des améliorations et un positionnement futurs.

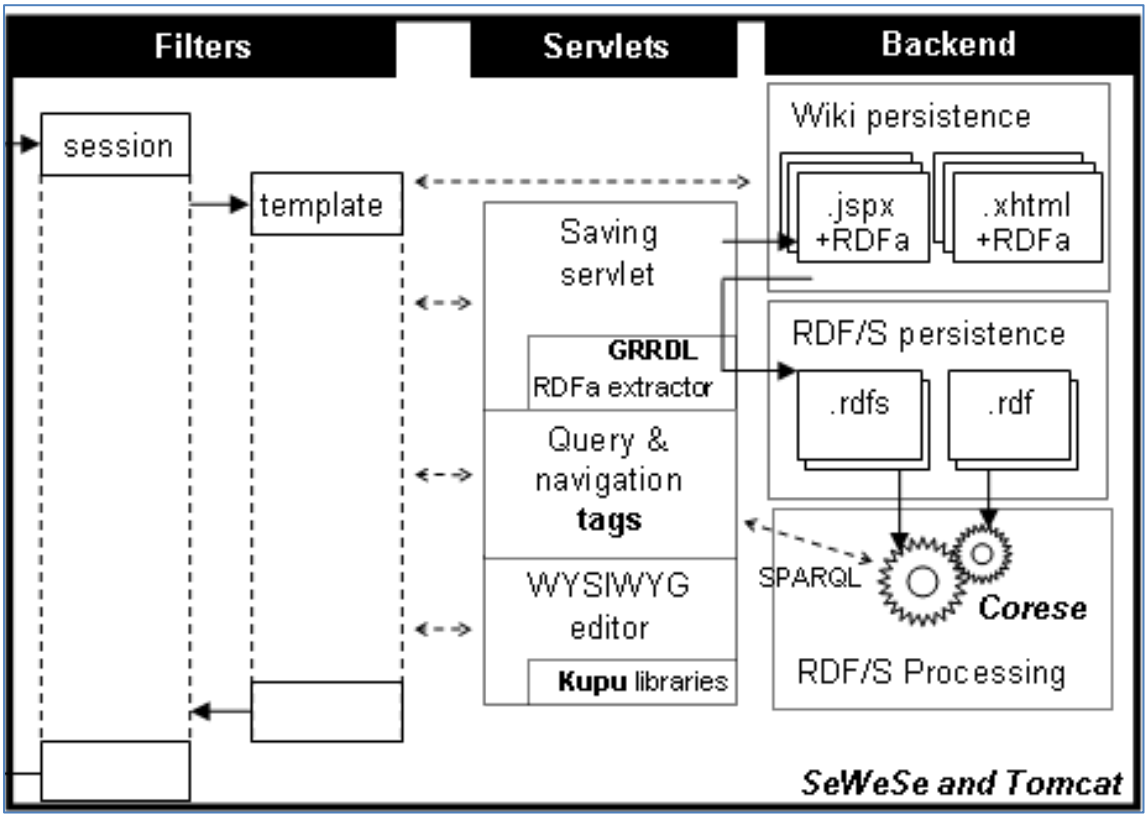


Figure 16. Architecture de SweetWiki dans SEWESE.

3.5 Architecture de SweetWiki

Dans cette section, nous détaillons les différents composants de SweetWiki : ontologies, moteur de recherche sémantique, éditeur WYSIWYG et support du balisage.

3.5.1 Ontologies dans SweetWiki

Les wikis sont des sites Web où les pages sont organisées autour de WikiWords, parfois en utilisant d'autres constructions telles que WikiWebs ou Workspaces[18]. Pour aller au-delà de cette structure informelle d'hyperliens, nous proposons de s'appuyer sur des fonctionnalités de balisage sémantique et de restructuration. Pour rendre explicite, manipuler et exploiter une telle structure, nous avons conçu SweetWiki autour d'une "

[18] Certains moteurs de Wikis de l'époque permettaient d'organiser *sommairement* leur contenu en regroupant des pages sous des catégories générales, en général avec un seul niveau de hiérarchie, appelés WikiWeb avec le moteur TWiki, ou WikiSpaces avec le moteur MediaWiki.

wiki ontologie " qui décrit les concepts wiki eux-mêmes. De plus, SweetWiki est également capable de tirer profit d'ontologies externes et des folksonomies développées par ses utilisateurs à travers le balisage.

3.5.1.1 Le modèle d'objet Wiki

SweetWiki s'appuie sur ce que nous avons appelé "*The Wiki Object Model*", en d'autres termes : *une ontologie de la structure wiki*. Dans de nombreux moteurs wiki, y compris plusieurs "wikis sémantiques", les concepts wiki sont cachés dans les implémentations *ad hoc*. Suivant les préceptes du « challenge de l'ontologie » de Mizoguchi and al. [93], nous les avons rendus explicites. Dans notre cas, nous avons défini en OWL Lite tous les concepts, propriétés et relations que nous utilisons dans le wiki lui-même. Des concepts tels que "document", "page", "tag", "lien", "lien vers l'arrière", "contributeur", "version", "fichier attaché", "image attachée", etc. sont décrits dans cette ontologie. Les métadonnées correspondantes sont intégrées dans les pages elles-mêmes.

Il y a plusieurs avantages à utiliser un tel modèle d'objet Wiki déclaratif. Nous pouvons raisonner à ce sujet, par exemple, utiliser des règles pour compléter les données, déclarer des relations symétriques pour maintenir la dualité entre les liens avant et arrière (si *« page A pointe vers page B »*, alors « *page B est pointée par page A* », ceci rend plus simple la programmation de certaines fonctions de recherche et de refactoring, etc. Nous pouvons ainsi plus facilement étendre et réorganiser le contenu du wiki (renommer, déplacer, supprimer des pages), et nous pouvons construire au-dessus, par exemple, faciliter l'interopérabilité entre plusieurs wikis. Nous pouvons générer des widgets pour aider à la navigation : lister les pages orphelines, ou pointant sur une autre page, ou afficher les pages partageant un même tag, etc. Nous verrons que l'outil SeWeSe – un éditeur d'ontologie interactif intégré à SweetWiki, permet également de hiérarchiser en classes et sous classes la folksonomie (entre autres) et faciliter la recherche « globale » de données dans le wiki (ex : rechercher toutes les pages « *qui parlent de robots* ». Bien que SweetWiki soit écrit en langage Java, les langages du web sémantique sont présents dans la quasi-totalité des fonctions principales du wiki. Le fait de se reposer sur RDF/OWL lite pour la description des métadonnées et des concepts du wiki lui-même illustre l'intérêt du langage SPARQL qui peut ainsi être utilisé pour interroger toutes ces métadonnées. L'ontologie de la structure wiki est maintenue par les concepteurs du wiki, alors que la folksonomie est construite à la volée et restructurée par les utilisateurs. Des ontologies externes (ex : ontologie sur les robots) peuvent aussi être intégrées, éditées et utilisées pour des annotations.

3.5.1.2 L'ontologie du domaine (folksonomie des utilisateurs)

SweetWiki supporte le marquage social (« *social tagging* »). Dans la version actuelle, les pages et les documents attachés (images, vidéos, fichiers attachés) peuvent être balisés à partir de l'éditeur. Les balises (« *tags* ») saisies par les utilisateurs forment une folksonomie. Ce point sera détaillé dans la section 3.6.

Afin de faciliter la navigation tout en conservant la simplicité habituelle, nous avons implémenté le mécanisme habituel d'utilisation de tags/mots-clés avec une ontologie de

domaine partagée par l'ensemble du wiki. Une page balisée avec les tags « robots » et « histoire » aura comme sujet les robots et l'histoire… Comme les tags sont libres, ils peuvent exprimer une panoplie assez large de significations, nous y reviendrons dans le chapitre 6, mais dans tous les cas, lorsqu'un élément de contenu du wiki est balisé avec un tag donné, on peut considérer que le tag est un des « sujets dont parle la page ».

En rendant cette ontologie des tags explicite (nous avons utilisé RDFS), nous pouvons à nouveau raisonner dessus, par exemple, trouver des tags sémantiquement proches, ou faire des requêtes complexes, par exemple, trouver des pages, images, vidéos, utilisateurs en fonction des tags avec lesquels ils ont été marqués. Les utilisateurs peuvent modifier interactivement cette ontologie, par exemple fusionner des tags (ce sont des concepts dans l'ontologie) équivalents, déclarer des liens hiérarchiques (subsomption), importer des ontologies de domaine existantes, etc.

L'ontologie du domaine est enrichie directement par les utilisateurs et peut être restructurée par des bénévoles de la communauté pour améliorer les capacités de navigation et d'interrogation. SweetWiki est livré avec un éditeur d'ontologie qui peut être utilisé par n'importe quel utilisateur afin d'organiser les balises dans la folksonomie (voir plus de détails dans la section 3.6.2).

3.5.1.3 Support de ressources externes

D'autres ontologies peuvent être ajoutées au moment de l'exécution par des utilisateurs privilégiés et deviennent immédiatement accessibles aux utilisateurs pour les requêtes SPARQL, le balisage/tagging et l'intégration. Si le wiki est utilisé dans un domaine, par exemple, la biologie, pour laquelle certaines ontologies sont déjà disponibles dans RDFS ou OWL Lite, ces ontologies peuvent être chargées dans le serveur web sémantique sous-jacent de SweetWiki. C'est aussi une excellente façon d'amorcer le pool de tags et cela facilite l'interopérabilité en favorisant la réutilisation d'ontologies existantes. En outre, nous prévoyons qu'une requête pourrait être incorporée dans une page mais dirigée vers d'autres serveurs SPARQL, permettant ainsi aux utilisateurs d'inclure les résultats de sites externes exécutant des instances de SweetWiki.

3.5.2 Moteur de recherche sémantique et serveur Web

Comme le montre la Figure 16, l'implémentation de SweetWiki repose sur le moteur de recherche sémantique CORESE [92][94] pour les requêtes et le raisonnement et sur SEWESE [95], une librairie de balises JSPs et un ensemble d'APIs de haut niveau pour implémenter des éléments d'interfaces utilisateur exploitant des ontologies. SEWESE implémente également (toujours sous forme de balises JSPs) un ensemble de fonctionnalités génériques (gestion de la sécurité, éditeurs d'ontologies, cycle de vie, etc.)

Il existe de nombreux autres documents et ressources en ligne décrivant le moteur CORESE sous-jacent ; nous n'entrerons pas dans ces détails ici. Cependant, une description concise pourrait être : un moteur de recherche sémantique complet implémentant la syntaxe SPARQL.

Le serveur Web de SweetWiki s'appuie sur une architecture d'application web standard écrite en Java avec les versions de 2006-2010 : des filtres interceptent les requêtes http et gèrent la session (Servlet Filters), par exemple, pour gérer les autorisations, les profils utilisateurs, etc. et les templates de présentation du wiki sont en XHTML ou JSPX standards. Une servlet gère la sauvegarde des pages et un ensemble de balises JSP fournit des fonctionnalités de haut niveau (par exemple, soumettre une requête SPARQL et formater le résultat avec une feuille de style XSLT). Des bibliothèques JavaScript sont utilisées, notamment pour l'implémentation de l'éditeur WYSIWYG Kupu, pour les mécanismes d'auto-complétion lors de la saisie de tags/balises, pour les requêtes Ajax etc. à l'époque l'utilisation de JavaScript pour implémenter ces fonctionnalités, rendant le Wiki très « dynamique » était un élément innovant, Ajax et l'API du DOM n'étant apparues que fin 2005 dans les navigateurs. Les premières versions de SweetWiki datent de fin 2005.

3.5.3 L'éditeur

Ainsi, SweetWiki utilise un éditeur WYSIWYG basé sur l'éditeur Open Source Kupu[19]. Cet éditeur est utilisé à la fois pour l'édition du contenu HTML en WYSIWYG, mais également pour la saisie des métadonnées et pour embarquer des requêtes SPARQL dans les pages. L'éditeur Kupu original a été considérablement modifié afin de supporter les extensions sémantiques et l'édition des métadonnées :

- Les requêtes SPARQL peuvent être intégrées dans une page en cours d'édition et sont traduites en temps réel en balises JSP à partir de la bibliothèque de balises SEWESE. Depuis l'éditeur, les requêtes SPARQL peuvent être testées et validées avant d'être insérées dans la page, comme illustré par les Figure 17 et Figure 18.
- Il est facile de créer des liens vers d'autres pages internes (pages du wiki) en utilisant les assistants AJAX qui exécutent des requêtes SPARQL pour obtenir la liste des documents existants et permettre l'auto-complétion.
- Le balisage/tagging d'une page ou d'éléments d'une page, comme des images, des vidéos ou des fichiers joints, repose également sur les appels AJAX à partir de l'éditeur. Par exemple, lors de la saisie d'une balise/tag, une requête SPARQL est émise et un mécanisme de complétion automatique propose la liste des balises existantes proches syntaxiquement ou hiérarchiquement.

[19] https://en.wikipedia.org/wiki/Kupu

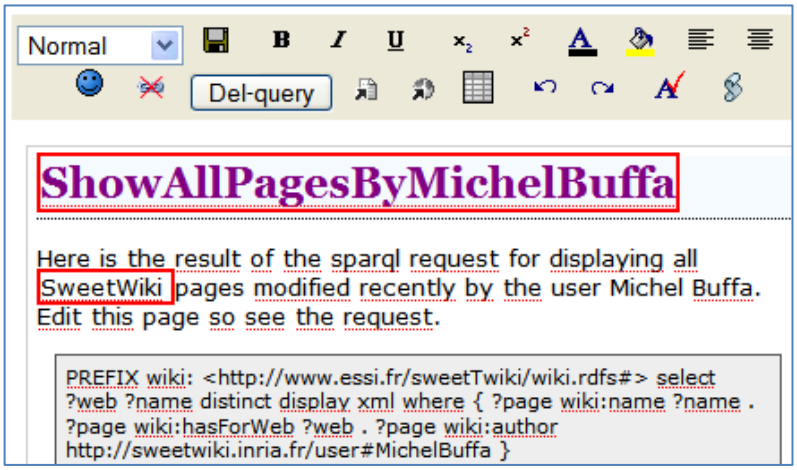


Figure 17. Exemple d'une requête SPARQL au sein d'une page de SweetWiki.

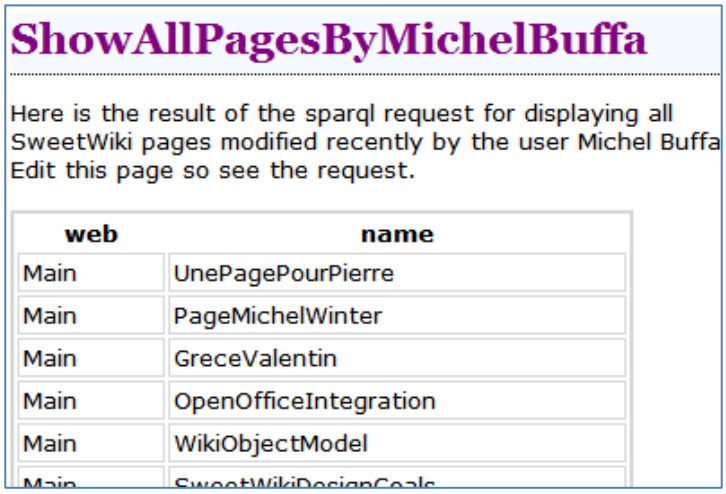


Figure 18. Page montrant les résultats de la requête de la figure précédente.

Certaines métadonnées sont incluses dynamiquement par l'éditeur dans le DOM de la page et d'autres sont ajoutées au moment de la sauvegarde (par exemple la date).

Nous utilisons la syntaxe RDFa[20] pour intégrer ces métadonnées dans le XHTML de la page. RDFa propose une solution pour augmenter le balisage existant avec des métadonnées, en utilisant des types de classes et de propriétés définis dans les schémas RDF, combinés avec le contenu existant de la langue hôte.

La Figure 19 montre le code RDFa correspondant à une image intégrée dans une page SweetWiki, étiquetée avec le mot-clé "holidays". L'espace de noms WIKI correspond à l'ontologie du wiki décrite dans la section 3.5.1.1.

[20] http://www.w3.org/TR/xhtml-rdfa-primer/

Annotation RDFa

```
<span about="/wiki/pub/102_9718.jpg">
    <img class="image-inline" alt="102_9718.jpg"
         src="/wiki/pub/102_9718.jpg" />
    <link href="WIKI:Image" rel="rdf:type">
    <link href="http://www.inria.fr/acacia/users-
    ontology#holidays"
          rel="WIKI:hasForKeyWord">
</span>
```

Figure 19. Exemple de code RDFa intégré dans une page SweetWiki

Contrairement aux annotations RDF externes, cette approche *reste dans l'esprit wiki* où tout est fait dans la page elle-même : le contenu (code HTML visible et métadonnées invisibles) peut être copié et collé en un seul endroit (la page wiki) même en utilisant un éditeur WYSIWYG. Avec RDFa, nous avons à la fois les données de page et les métadonnées servies dans le même fichier autonome (XHTML) et les pages peuvent être parcourues par des applications externes ou sauvegardées par les utilisateurs en utilisant leur navigateur sans perte d'information.

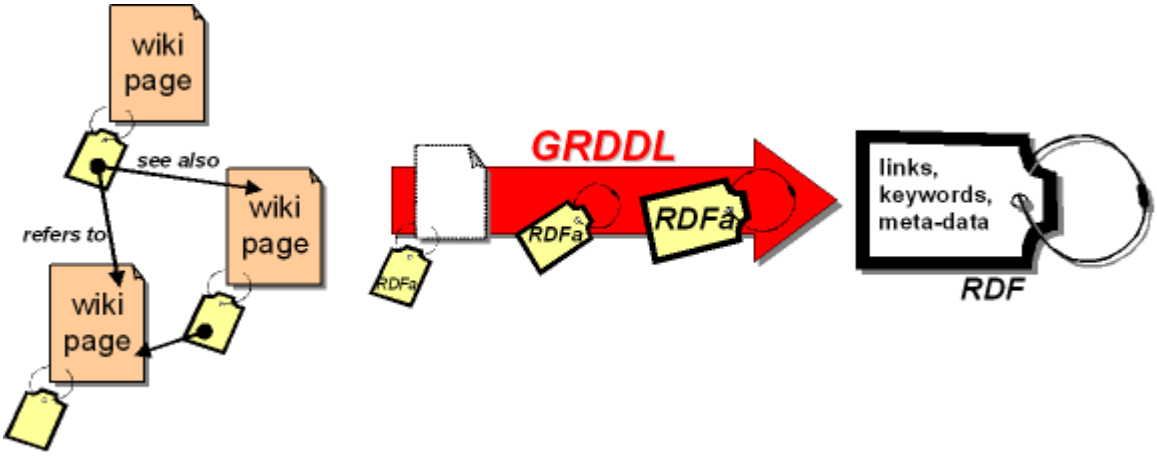


Figure 20. GRDDL permet d'extraire les annotations RDF des pages du wiki pour ensuite les utiliser pour structurer et naviguer dans le wiki.

GRDDL[21] est un mécanisme permettant d'obtenir des données RDF à partir de documents XML et XHTML en utilisant des algorithmes de transformation explicitement associés, typiquement représentés dans des feuilles XSLT [96]. Nous l'utilisons pour extraire les métadonnées des pages wiki et ce scénario a inspiré l'un des cas d'utilisation motivants proposés pour GRDDL (Figure 20).

[21] Gleaning Resource Descriptions from Dialects of Languages (GRDDL) http://www.w3.org/2004/01/rdxh/spec

Par conséquent, lors de l'enregistrement d'une page, plusieurs opérations se produisent :

1. Les métadonnées sont extraites de la page et traduites en RDF/XML à l'aide d'une transformation GRDDL (feuille de style XSLT).
2. Le RDF est chargé dans le moteur de recherche sémantique CORESE,
3. Une autre feuille de style est appliquée pour transformer le contenu dynamique de la page en balises JSP (par exemple, les requêtes SPARQL).

Les pages sont servies telles quelles par le serveur web Tomcat ; aucun traitement supplémentaire n'est nécessaire car les pages sont des pages JSP ou XHTML standards. Par exemple, si une page contient une requête intégrée comme dans l'exemple précédent, la balise JSP sera exécutée, la requête sera envoyée au moteur de recherche sémantique et les résultats seront insérés dynamiquement sous la forme d'un tableau XHTML, comme le montre la Figure 18.

L'éditeur ne peut éditer que du XHTML pur alors que certaines pages wiki sont au format JSPX. En fait, seules les parties dynamiques des pages, comme les requêtes SPARQL embarquées sont encodées en utilisant des balises non XHTML, le reste de la page est du XHTML standard. Afin de rendre la page modifiable à l'aide de l'éditeur, nous appliquons une feuille de style qui transforme ces balises en "vue XHTML" (typiquement en utilisant des balises `<span>...</span>` XHTML). Il s'agit exactement de l'opération inverse de celle effectuée à l'étape 3, lorsque la page a été sauvegardée. Un exemple typique est une requête SPARQL qui a été saisie dans l'éditeur WYSIWYG. Elle est codée dans le DOM à l'aide de certains attributs `<span>` XHTML qui sont ensuite transformés en balises JSP fournies par la boîte à outils SEWESE.

Prenons pour exemple la requête suivante :

Requête embarquée dans une balise JSP

```
<span property="sparql:query"
     xslt="/xsl/query_result_as_table.xsl">
     PREFIX wiki:
     &lt;http://www.essi.fr/sweetTwiki/wiki.rdfs#&gt;

     select ?web ?name distinct display xml
     where {
         ?page wiki:name ?name .
         ?page wiki:hasForWeb ?web .
         ?page wiki:author
                   http://sweetwiki.inria.fr/user#MichelBuffa
     }
     </span>
```

Elle se traduit en une balise JSPX par la librairie de balises SEWESE :

Requête embarquée dans une balise JSP

```
<sew:query query="PREFIX wiki:
     &lt;http://www.essi.fr/sweetTwiki/wiki.rdfs#&gt;
      select ?web ?name&#10;distinct display xml
      where {
          ?page wiki:name ?name .
          ?page wiki:hasForWeb ?web .
          ?page wiki:author
                    http://sweetwiki.inria.fr/user#MichelBuffa
     }"
     xslt="/xsl/query_result_as_table.xsl"
/>
```

Une telle approche déclarative rend l'extension des capacités de SweetWiki extrêmement facile. Imaginez que vous vouliez ajouter le support pour intégrer des vidéos - le modèle est facile à étendre. Voici les étapes à suivre :

4. Définir dans le modèle d'objet Wiki (l'ontologie wiki) le concept de "fichier vidéo", le lier au concept de page, et ajouter toute propriété nécessaire (longueur vidéo, codec, nom de fichier, etc.).
5. Ajouter un plugin à l'éditeur (bouton + callback pour insérer le code XHTML dans la page (en utilisant quelques) + insertion des métadonnées correspondantes dans RDFa).
6. Mettre à jour les deux feuilles de style pour la conversion de la vue XHTML à la vue dynamique des balises JSP.
7. Mettre à jour la librairie de balises afin que la balise JSP correspondante génère le code XHTML pour l'intégration d'un visualiseur pour le format de fichier vidéo.

Maintenant, dans n'importe quelle page du wiki, vous pouvez intégrer des requêtes SPARQL pour interroger toutes les vidéos dans les pages créées par un utilisateur donné.

3.5.4 Intégration des documents Office

SweetWiki possède également un module pour importer des documents Open Office ou Microsoft Office (support des fichiers Word et Excel). Les documents sont automatiquement traduits en pages SweetWiki qui peuvent à leur tour être éditées, étiquetées et partagées. Un serveur Open Office et des modules de conversion ad hoc sont utilisés en coulisses pour convertir les documents bureautiques.

3.6 Frameworks Web sémantique et marquage social (« *social tagging* »)

Dans cette section, nous décrivons comment le balisage social a été implémenté dans SweetWiki en utilisant des frameworks du web sémantique.

3.6.1 Support du marquage social dans le wiki

SweetWiki supporte le marquage/tagging social. Les utilisateurs peuvent marquer des pages, des images, des fichiers attachés, etc. Comme dans [97] et [98], nous proposons une approche mixte pour "organiser les balises/tags". Nous lions les tags ensemble dans une folksonomie décrite en utilisant les langages du web sémantique, où les tags sont organisés dans une hiérarchie et reliés les uns aux autres en utilisant des relations comme `subClassOf`, `seeAlso`, etc. Gruber va plus loin et a proposé dans [99] de définir une "écologie de l'Internet" pour les folksonomies, c'est-à-dire une ontologie pour décrire les folksonomies. De même, nous croyons que le marquage social minimise les coûts et maximise la participation des utilisateurs, donc nous soutenons le marquage social dans SweetWiki, mais nous pensons aussi que les tags doivent être organisés. Le système que nous avons mis en place aide les utilisateurs à construire une folksonomie utile tout en s'appuyant sur les technologies standards du web sémantique pour organiser et maintenir la folksonomie. SweetWiki utilise les folksonomies et le marquage social comme *une meilleure façon de catégoriser les documents wiki* [100][101]. D'autres efforts ont été menés récemment pour créer une ontologie standard des balises/tags, comme le RFC proposé par Newman [102], ou en incluant une définition des balises dans une ontologie plus large comme dans le cadre du SIOC. Dans notre cas, nous définissons les balises dans le modèle d'objet Wiki (voir section 3.5.1.1).

Comme décrit dans la section 3.5.3, SweetWiki intègre un éditeur WYSIWYG standard que nous avons étendu pour supporter directement les annotations sémantiques suivant l'approche du "social tagging". Les utilisateurs peuvent assigner des concepts à des pages et utiliser un éditeur pour modifier la hiérarchie des concepts.

Comme le montre la Figure 22 page suivante, lors de l'édition d'une page, l'utilisateur peut entrer librement certains mots-clés dans un champ de texte alimenté par des requêtes AJAX. Un mécanisme d'auto-complétion propose des mots-clés existants en émettant des requêtes SPARQL au serveur web sémantique afin d'identifier les concepts existants avec des étiquettes compatibles et montre le nombre de pages partageant ces concepts comme une incitation à les utiliser.

De plus, des catégories parentales sont également affichées afin d'identifier les ambiguïtés d'homonymie (deux tags ayant le même label mais ne signifiant pas la même chose ex : Apple/fruit et Apple/ordinateur). Avec cette approche, le balisage reste facile (utiliser un mot-clé) et devient à la fois motivant et sans ambiguïté. Des mots-clés inconnus sont collectés lors de leur première utilisation et attachés à de nouveaux concepts dans l'ontologie. Plus tard, les experts de la communauté pourront les repositionner dans l'ontologie, les éditer, leur attacher des tags parents etc. La rétroaction provenant des balises est utile pour améliorer l'ontologie et pour découvrir le vocabulaire d'une communauté.

Le moteur CORESE est utilisé pour générer des widgets de navigation à facettes. La sémantique des balises est utilisée pour suggérer des sujets connexes, interroger le moteur sur des pages similaires à l'aide de requêtes SPARQL, avertir les utilisateurs intéressés,

etc. (voir la Figure 23). Quand un document SweetWiki est demandé par un navigateur web, des templates sont utilisés pour intégrer les widgets de navigation à facettes autour du contenu de la page. Ces templates peuvent être modifiés à la manière des « skins » du moteur wiki TWiki (ils sont eux-mêmes des pages spéciales du wiki et peuvent être modifiés directement à l'aide de l'éditeur du wiki.)

3.6.2 Intégration d'un éditeur d'ontologie dans le wiki

Cet éditeur peut être utilisé par n'importe quel utilisateur pour maintenir la folksonomie et pour importer, créer et mettre à jour des ontologies externes. En laissant les utilisateurs eux-mêmes organiser et mettre à jour la folksonomie, il a été démontré que la recherche et la navigation sont généralement améliorées [100][101]. L'idée sous-jacente est que même si un petit pourcentage d'utilisateurs contribue à l'entretien de la folksonomie, l'ensemble des utilisateurs bénéficieront de chaque amélioration. Les expériences menées chez Electricité de France (EDF) [103] ont confirmé la promesse d'une telle approche (même si, dans ce cas, seuls les administrateurs pouvaient mettre à jour la structure de la folksonomie).

Des outils de supervision sont intégrés dans SweetWiki en s'appuyant sur le serveur web sémantique SEWESE. Ils sont utilisés pour surveiller l'activité du wiki en exécutant des requêtes SPARQL sur les métadonnées, par exemple pour connaître la fréquence d'utilisation de chaque tag (Figure 21), les tags introduits récemment, les pages orphelines, etc.

Afin de maintenir la folksonomie, SweetWiki réutilise également les éditeurs Web disponibles dans SEWESE (Figure 24). Dans nos exemples, nous avons taggué des pages contenant un cours pour apprendre le langage de programmation Java, en utilisant une ontologie décrivant les principaux concepts de ce langage. En sélectionnant cette ontologie dans l'éditeur, on peut ajouter, supprimer et éditer des concepts (tags, parents, commentaires) dont les labels sont utilisés pour tagger les pages. En particulier, si un concept a été récemment ajouté, il peut être repositionné dans la hiérarchie par l'utilisateur (Figure 24).

En utilisant cet éditeur, la folksonomie et les annotations peuvent être mises à jour. Par exemple, les experts de la communauté peuvent choisir une paire de balises et déclarer des relations sémantiques entre elles telles que `subClassOf`. Ils peuvent aussi fusionner des concepts lorsque deux tags sont synonymes, etc. Les améliorations de l'ontologie à leur tour améliorent également de façon transparente le partage de contenu : la recherche et la navigation à facettes bénéficient directement des mises à jour (par exemple, de nouvelles suggestions apparaissent dans les widgets de navigation). Et avec la façon dont le système est conçu, (gestion de version des pages - et donc des annotations, des ontologies, de la folksonomie) on ne peut pas briser la cohérence du système. Si une balise est soudainement manquante dans l'ontologie mais toujours utilisée dans les pages, elle devient simplement une nouvelle balise (et rajoutée comme telle) et si de nombreuses pages existent avec l'ancienne balise (les pages ne sont pas touchées dans le processus d'édition de la balise), la balise réapparaît (avec un nombre élevé de pages balisées,

encourageant d'autres personnes à l'utiliser). La réingénierie de l'ontologie est une façon de refactoriser le wiki - de nouveaux liens et groupes apparaissent au fur et à mesure que l'ontologie s'enrichit.

A partir de l'interface utilisateur standard, seules les manipulations de base sont autorisées sur l'ontologie, correspondant essentiellement à l'expressivité RDFS plus les caractéristiques algébriques des propriétés et la déclaration inverse des propriétés. La hiérarchie des propriétés peut également être modifiée, mais les nouvelles propriétés nécessitent de nouveaux widgets ou plug-ins pour générer les triplets RDF correspondants. Par exemple, lorsque la propriété "`interested by`" a été ajoutée, nous avons développé un widget pour les pages d'accueil des utilisateurs afin de générer les annotations correspondantes.

Keyword	Subclass Of	Pages
sweetwiki	wiki_enginewikisoftware	12
palette	.	10
model	.	8
CoP	communitypalette	7
Competency	palette	6
ux11	CoPcommunitypalette	5
mbds	filiere_de_formation	5
ejb	NewConcept	4
Collaboration	palette	4

Figure 21. Tags triés par popularité (résultats calculés par une requête SPARQL à l'aide de l'extension COUNT fournie par CORESE)

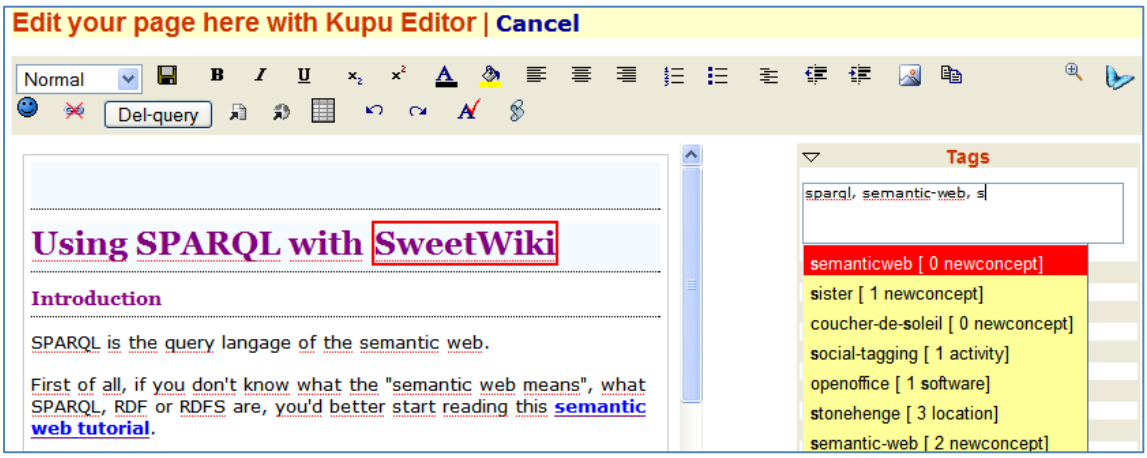


Figure 22. Les balises sont suggérées au fur et à mesure que l'utilisateur entre des mots-clés. Le nombre de pages utilisant chaque balise et la catégorie correspondante sont affichés.

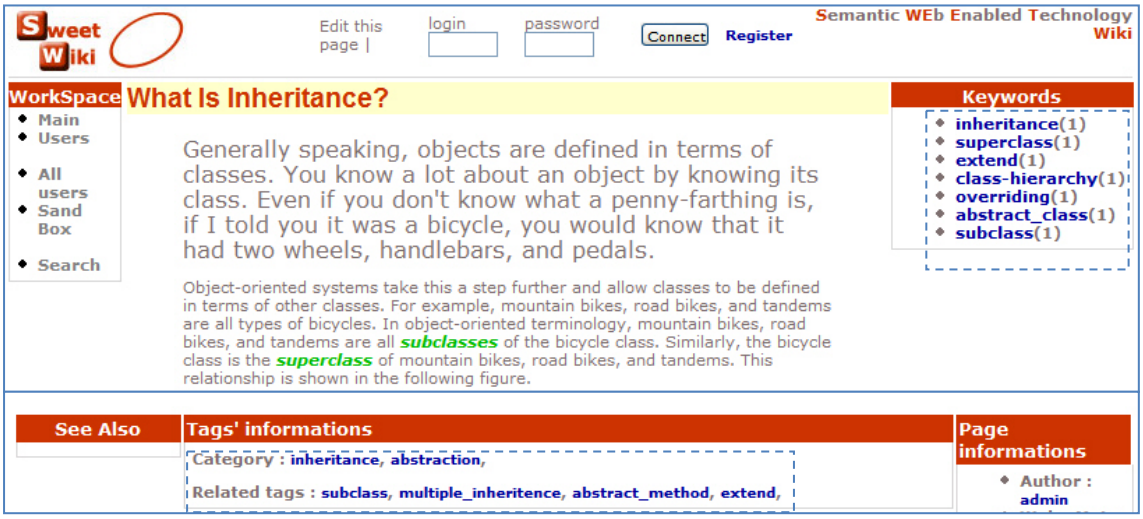


Figure 23. Liens de navigation à facettes extraits des balises.

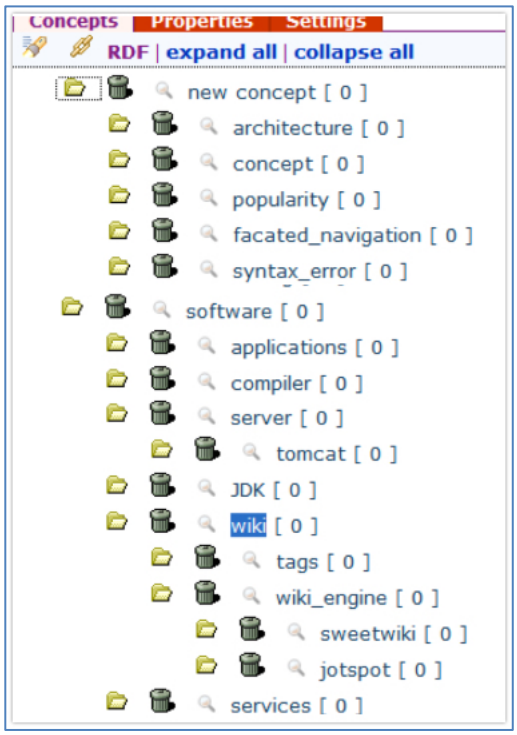


Figure 24. L'éditeur d'ontologie, avec la folksonomie des utilisateurs.

3.7 SweetWiki est un wiki d'application

Plusieurs moteurs wiki comme TWiki, JotSpot, Confluence ou XWiki sont appelés "wikis d'application" dans le sens où le langage WikiML utilisé pour formater les documents inclut des macros très puissantes. Ces macros facilitent l'écriture d'applications simples basées sur des tables, sans avoir besoin de créer une base de données ou d'écrire plus que quelques lignes de code (la section 2.4.3 a montré des exemples de telles applications développées dans le wiki de l'intranet d'ILOG).

Aucun de ces wikis n'utilise éditeur WYSIWYG pour écrire des applications wiki. Seuls quelques utilisateurs ont les compétences pour écrire de telles applications wiki (voir [59]

et [55]) mais cette fonctionnalité fait du wiki un outil puissant pour ajouter de "petites applications" sans avoir de connaissances avancées en programmation.

Dans SweetWiki nous pouvons intégrer du contenu dynamique comme des requêtes, dans les pages, directement depuis l'éditeur WYSIWYG et ainsi interroger n'importe quelle partie du wiki (à travers le modèle d'objet Wiki). Nous avons montré dans les sections précédentes qu'en associant les balises XHTML `<span>....</span>` et les balises JSP, nous pouvons facilement rendre n'importe quel contenu dynamique/macro modifiable dans l'éditeur WYSIWYG.

En permettant l'édition et l'interrogation dynamiques, tout utilisateur peut saisir des requêtes complexes dans n'importe quelle page et fournir des "*pages d'application*". La capture d'écran de la Figure 25 montre une page de recherche fournie par la distribution standard SweetWiki - la page est éditable et les utilisateurs peuvent modifier les requêtes SPARQL pour répondre à leurs besoins.

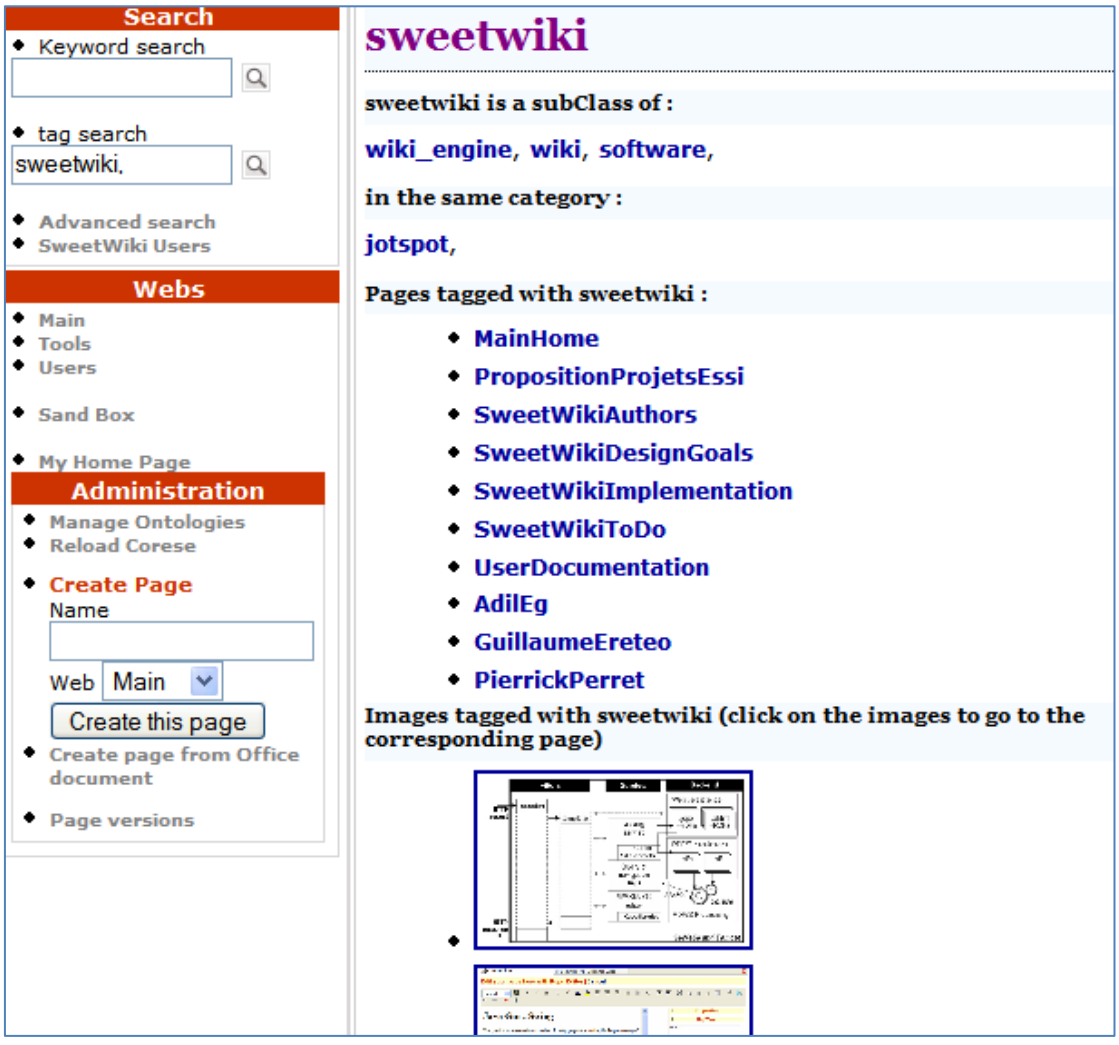


Figure 25. Les pages de recherche peuvent être éditées dans l'éditeur WYSIWYG.

L'exemple montre les résultats d'une recherche sur la balise "sweetwiki" : les pages et les images marquées avec cette balise sont affichées mais aussi les balises associées sont extraites de la folksonomie de l'utilisateur. Un point important est qu'une telle page a été

écrite comme une page SweetWiki "standard", en utilisant l'éditeur WYSIWYG. Les utilisateurs peuvent télécharger des feuilles de style XSLT afin de personnaliser la présentation des résultats des requêtes. Dans la version actuelle, l'éditeur WYSIWYG ne fournit pas d'aide à la construction de la requête, à l'exception d'une validation à la volée de la syntaxe SPARQL et d'un aperçu des résultats. Une autre limitation est l'absence d'un éditeur de formulaires riches pour paramétrer les requêtes.

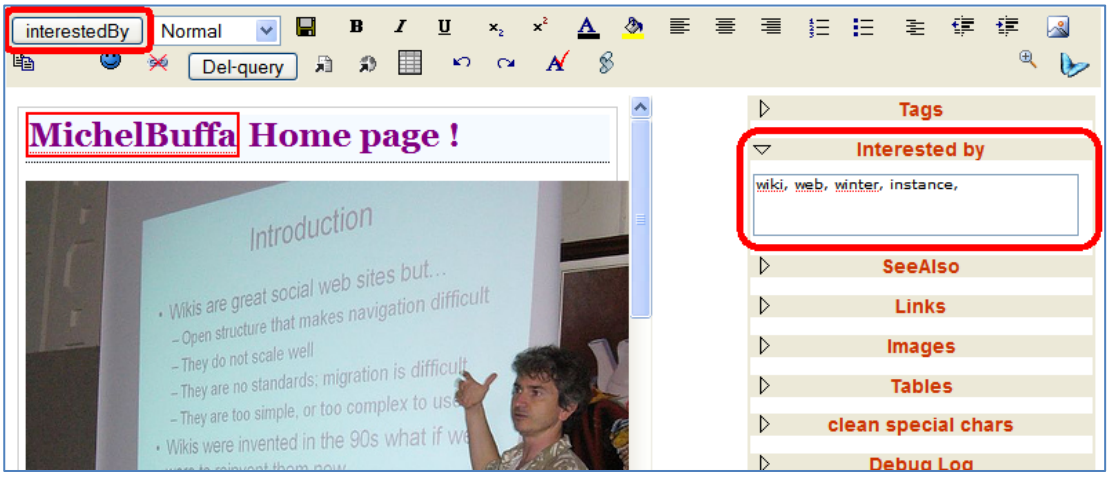


Figure 26. Une page d'accueil, étiquetée avec des intérêts personnels.

Lorsqu'un utilisateur s'inscrit, une page d'accueil est automatiquement créée. Les métadonnées de cette page d'accueil sont un peu différentes de celles des autres pages standard car elles représentent le profil de l'utilisateur. Dans la version actuelle, un utilisateur peut entrer une liste de balises correspondant à des sujets qui l'intéressent. Par exemple, un utilisateur qui se dit intéressé par les "wikis" aura des liens suggérés qui lui seront proposés ou envoyés par mail. Comme dans toute résolution de requête soumise à CORESE, l'héritage est pris en compte. Si certaines pages ou images sont étiquetées avec des sous-classes de la balise "wiki", nous supposons que ces pages "parlent aussi des wikis" et elles seront également suggérées. La Figure 26 montre la page d'accueil de l'utilisateur en mode édition. La Figure 27 montre une partie de la page d'accueil de l'utilisateur qui est rendue une fois que la page est sauvegardée et consultée : elle montre la liste des pages du wiki susceptibles d'intéresser. Dans cette approche, les utilisateurs peuvent anticiper de nouveaux sujets et définir l'intérêt a priori avant même qu'une page ne soit créée sur un sujet donné. Dans les wikis classiques, vous pouvez être notifié d'une page en cours de modification ; ici vous pouvez aussi être notifié quand une page est créée dans le futur, par exemple quand quelqu'un utilisera une spécialisation d'une balise qui vous intéresse, etc.

Last updated pages containing the tags which interest you

WikiPage	Web	Author	Description	Last update	Keyword
PropositionProjetsEssi	Main	GuillaumeEreteo	Description - not implemented	2006-11-15	sweetwiki
SweetWikiToDo	Main	GuillaumeEreteo	Description - not implemented	2006-11-15	sweetwiki
MyleneLeitzelman	Users	MyleneLeitzelman	Description - not implemented	2006-11-13	wiki
MainHome	Main	AdilElGhali	Description -	2006-11-10	sweetwiki

Figure 27. Pages suggérées en rapport avec les sujets d'intérêt d'un utilisateur. Intéressant pour la veille sur le contenu du wiki.

Il est également intéressant d'analyser le comportement des utilisateurs : ce qu'ils font, ce qu'ils savent, ce qu'ils veulent. Par exemple, en gardant la trace des pages créées et modifiées, des documents balisés et surveillés, et plus généralement en analysant le comportement des utilisateurs au fil du temps, il est possible de construire des réseaux de connaissances ou des communautés d'intérêt. Nous avons donc mis à jour le modèle d'objet wiki pour stocker toutes les métadonnées nécessaires. Ensuite, en raisonnant sur ces métadonnées avec des règles simples comme "si une personne consulte une page tagguée avec un sujet donné (par exemple, « ajax »), alors elle peut avoir des connaissances sur le sujet parent dans la folksonomie (par exemple, « web 2.0 »), et nous pouvons, par exemple, indiquer cette personne aux personnes intéressées par « web 2.0 ». Jusqu'à présent, nous avons mis en place de nouvelles fonctionnalités de navigation et de recherche basées sur ces idées :

- Chercher la personne la plus active sur un sujet donné pour aider à trouver des experts.
- Établir des similitudes de comportement entre les utilisateurs pour favoriser les communautés d'intérêt.
- Inférer les relations entre les tags qui ne sont pas explicites dans la folksonomie, comme la cooccurrence ou le partage de contexte.

3.8 Discussion

Pour résumer le scénario global exploré dans SweetWiki : nous avons proposé une approche innovante qui permet aux utilisateurs d'éditer les pages du wiki et de les baliser en utilisant une conceptualisation partagée et explicite, mais qui est cachée en coulisses. De plus, les experts de la communauté peuvent vérifier le modèle sous-jacent en cours de construction, regarder les balises proposées par les utilisateurs et les (ré)organiser. Si cela se produit, les annotations que les utilisateurs ont saisies ne sont pas modifiées, mais la navigation à facettes et la recherche basée sur des requêtes sémantiques sont améliorées automatiquement par de nouveaux liens.

SweetWiki n'est plus maintenu aujourd'hui (en 2018) mais il a été utilisé dans plusieurs projets pour partager l'information entre participants. Des communautés l'ont utilisé

comme plateforme de collaboration, principalement pour coordonner le travail des chercheurs. En effet, notre système s'adresse aussi bien aux utilisateurs non techniques qu'aux experts :

- Palette : un projet européen sur les outils et méthodes d'aide à l'apprentissage dans les communautés de pratique. Nous avons commencé à surveiller le comportement des communautés de pratique dans Palette (section 3 de [104]) et leurs commentaires et exigences (section 4 de 104]) Les résultats d'utilisabilité pour SweetWiki indiquent un intérêt particulier pour le balisage sans édition de pages, des moyens semi-automatiques pour organiser les balises, des améliorations ergonomiques dans les interfaces et des mécanismes de notification avancés.
- E-Wok : un projet national de recherche sur les techniques de captage du CO2 nécessitant une collaboration entre les instituts géologiques et pétroliers.
- Tests informels : nous avons réalisé des expériences avec des enfants, des enseignants, des élèves et des secrétaires leur demandant de l'utiliser pour l'une de leurs tâches quotidiennes.

Il est également utilisé dans plusieurs expériences :

- Dans l'un d'entre eux, on nous a demandé d'ajouter des options pour verrouiller des parties de la folksonomie lorsque la classification est jugée satisfaisante (les enseignants contribuant au wiki craignaient que la folksonomie puisse se dégrader avec le temps) ;
- Dans un autre cas (une société de robotique utilisant SweetWiki pour son intranet ainsi que pour son site communautaire public), nous avons décidé de ne pas annoncer la présence d'un éditeur de folksonomie. Après quelques mois, nous avons examiné le vocabulaire qui avait émergé et, avec un utilisateur principal de leur communauté, nous avons commencé à le réorganiser. Nous observons la réaction des utilisateurs face à la réorganisation induite dans le wiki, par exemple, de nouveaux liens suggérés. Cette expérience a montré une nette amélioration de la recherche dans le wiki et la société Robosoft (leader européen en robotique mobile) a fait de SweetWiki l'outil principal de maintenance de sa documentation interne et externe entre 2006 et 2015 (voir Figure 28 et Figure 29) [200].

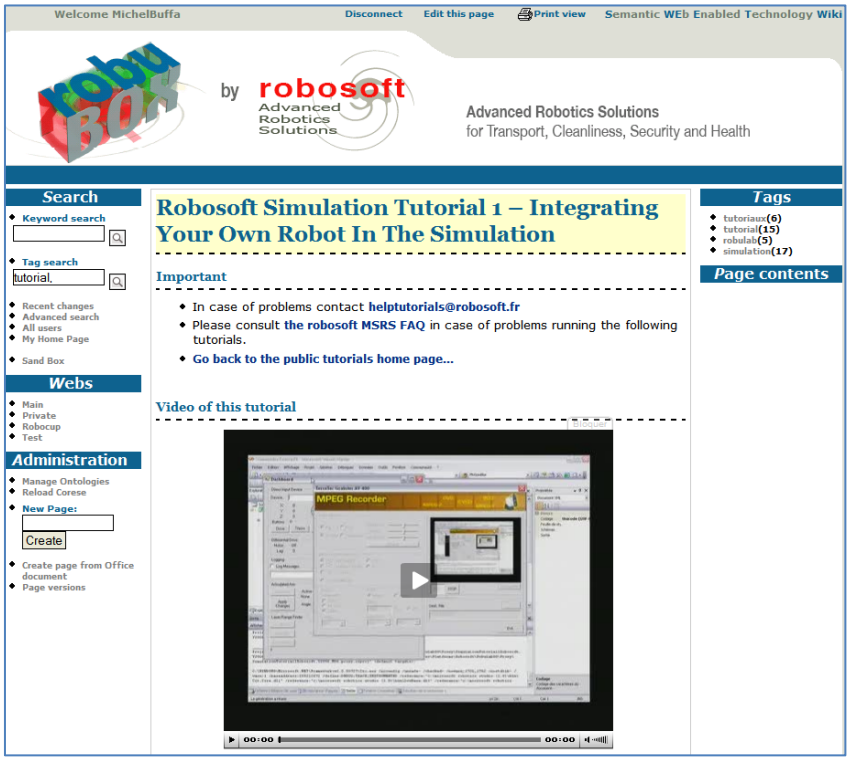


Figure 28. SweetWiki utilisé pour la documentation interne de la société Robosoft, leader européen dans la robotique mobile.

Figure 29. Intérêt d'utiliser une ontologie de domaine (folksonomie structurée) : la recherche dans la documentation permet de typer et hiérarchiser les résultats (plus général / plus spécialisé).

Dans ces cas, les folksonomies construites sont des "ontologies légères" ou des représentations intermédiaires qui pourraient être utilisées comme points de départ pour générer des ontologies plus formelles.

Avec SweetWiki, les utilisateurs occasionnels aussi bien que les experts peuvent utiliser le même outil pour écrire de petites applications dans le wiki ou pour améliorer l'ontologie utilisée avec l'éditeur intégré. La prise en charge des ontologies externes et la possibilité d'intégrer des contenus dynamiques, par exemple des requêtes SPARQL dans les pages, transforme SweetWiki en une "plate-forme d'application web sémantique". Même les concepts de base wiki comme les pages, les liens, les images, les auteurs, etc. peuvent être étendus et interrogés, en s'appuyant sur le modèle *Wiki Object Model* exprimé en OWL Lite.

3.9 Conclusion

SweetWiki a été une contribution importante de « la première vague des wikis sémantiques ». L'article « SweetWiki : a semantic wiki » paru dans la revue IJWS 2008 et dont nous avons repris les grandes lignes dans ce chapitre, est aujourd'hui un des articles les plus cités sur le sujet des « wikis sémantiques ».

Par ailleurs, la conception et le développement de SweetWiki ont permis de défricher les travaux qui vont suivre, concernant l'utilisation de wikis sémantiques comme outils intégrés dans un réseau social d'entreprise et véritablement programmables (au-delà de la possibilité d'inclure des requêtes ou des balises spéciales dans les pages).

L'état de l'art proposé dans ce chapitre se verra réactualisé car comme nous, d'autres groupes de chercheurs ont suivi la même voie (on pense à IkeWiki qui deviendra un framework intégré lui aussi dans un réseau d'entreprise, à Semantic Media Wiki qui sera étendu pour augmenter sa partie « programmation d'applications », etc.)

4. Wikis sémantiques d'application (mes travaux sur la période 2010-2014)

Contexte

En 2010 c'est la grande période des APIs REST, et l'apparition massive du "Web Programmable". Dans le cadre du projet ANR ISICIL[22] [105] [119] il a été décidé de pousser plus loin le concept de Wiki Sémantique en créant SweetDeki[23], un des premiers "*wikis sémantiques d'application*". Un wiki sémantique d'application est un Wiki programmable, directement depuis le navigateur. On pouvait ainsi créer des documents en pressant le bouton "edit" ou "nouvelle page" dans l'outil, mais on pouvait également écrire des applications, par exemple un mini forum, ou des programmes générant des pages d'index du contenu du wiki ou encore des pages visualisant les relations sociales entre utilisateurs. Nous avions déjà ouvert cette voie avec SweetWiki qui permettait d'intégrer des requêtes SPARQL dans des pages, mais pas de coder de réelles applications complexes.

Par ailleurs ce wiki était doté de plusieurs APIs REST permettant à des programmes externes d'exploiter les données qu'il hébergeait : les documents, les interactions entre utilisateurs (enrichissant le réseau social du projet ISICIL de manière dynamique, deux personnes ayant édité une même page se retrouvant automatiquement "reliées), et les programmes que le wiki contient dans ses pages. Toutes ces ressources pouvaient être échangées avec les outils externes du réseau social ISICIL. Dans les pages du Wiki on pouvait coder, de la même manière qu'on éditait un document, des requêtes SPARQL et des algorithmes dans un langage dédié (DekiScript) qui allaient ensuite servir à visualiser par exemple, les personnes du réseau social susceptibles d'être expertes dans un certain domaine. Voir les illustrations présentées plus loin.

En même temps, les premiers IDEs en ligne comme jsbin.com ou codepen.io sont apparus, lançant la mode du "*live coding*" et des éditeurs JavaScript/HTML/CSS en ligne. Ces outils ont inspiré WikiNext, un autre wiki sémantique programmable, plus expérimental, que j'ai co-conçu avec Pavel Arapov, dont j'ai dirigé seul la thèse.

Contributions

J'ai conçu, supervisé et co-développé le logiciel SweetDeki, principal livrable de mon équipe de recherche pour le projet ANR ISICIL. Il s'agit là d'un des premiers "wikis sémantiques d'application", mon travail a contribué à définir ce terme et ce concept, et à

[22] Intégration Sémantique d'Informations à travers des Communautés d'Intelligence en Ligne. Le site web du projet est archivé ici. Il contient de nombreuses vidéos de démonstration du wiki présenté ainsi que d'autres outils du projet qui sont aujourd'hui disponibles sur DailyMotion : https://www.dailymotion.com/isicil

[23] le nom est un mélange de SweetWiki et de DekiWiki, un wiki open source créé par la société Mindtouch.

l'implémenter. Le wiki était également intégré au réseau social sémantique créé pour le projet ISICIL, j'ai participé au développement des ontologies utilisées par ISICIL pour la modélisation du réseau social et du système de social tagging des ressources et des utilisateurs. Par la suite, entre 2010 et 2014 j'ai continué à travailler sur les wikis sémantiques d'application avec la thèse de Pavel Arapov, qui a consisté à prendre le meilleur des wikis sémantiques (création de documents enrichis sémantiquement) et le meilleur des IDEs en ligne (je code directement dans mon navigateur, je vois rapidement ce que cela donne) pour le développement d'applications tirant parti au maximum des technologies du Web sémantique et des nouveautés techniques de l'époque (HTML5, NodeJS, WebSockets, etc.). Ce travail a débouché sur la création de WikiNext, un wiki sémantique d'application permettant de programmer dans ses pages des applications exploitant le web de données.

Enfin, de manière transversale, j'ai conçu avec Catherine Faron l'ontologie AMO (Access Management Ontology) pour une gestion des droits d'accès aux ressources du projet ANR ISICIL. J'ai réalisé l'intégralité de son implémentation dans SweetWiki et participé à son intégration dans les serveurs de métadonnées du projet ANR ISICIL.

Collaboration, encadrement

J'étais responsable scientifique du projet ANR ISICIL dans mon équipe. J'ai encadré pendant trois ans un ingénieur recruté sur ce projet. Entre 2010 et 2014 j'ai encadré à 100% la thèse de Pavel Arapov ("*Wikis sémantiques d'applications*"). J'ai co-encadré à 50% avec Fabien Gandon les thèses de Guillaume Ereteo ("*Semantic Social Network Analysis*") et Freddy Limpens ("*Multi-points of View Semantic Enrichment of Folksonomies*"), dont une partie des contributions ont été intégrées dans les outils ISICIL développé par mon équipe.

J'ai écrit plusieurs cours en ligne sur HTML5 pour le W3C (la plateforme w3devcampus.com) qui sont devenus des MOOCs sur la plateforme edX du MIT/Harvard, et qui ont réuni aujourd'hui plus de 450.000 personnes. J'ai développé ces MOOCS pour le W3C et pour l'Université Côte d'Azur. Le premier a été sélectionné dans la liste des dix meilleurs MOOCs de la plateforme en 2016 (parmi plus de 1700)[24].

J'ai supervisé (avec Catherine Faron) le stage de master recherche de Anna Kolomoïska qui a travaillé sur AMO, l'ontologie pour la gestion des droits d'accès aux ressources du projet ISICIL.

Publications principales ayant résulté de ces travaux

Pavel Arapov, Michel Buffa, Amel Ben Othmane. *Programmer le web de données avec un "Wiki-based IDE"*. IC - 25èmes Journées francophones d'Ingénierie des Connaissances IC 2014, May 2014, Clermont-Ferrand, France. pp.249-260, 2014. 〈hal-01015312〉

Pavel Arapov, Michel Buffa, Amel Ben Othmane. "*Developing web of data applications from the browser.*" In : Proceedings of the 23rd International Conference on World Wide Web (WWW '14

[24] https://blog.edx.org/congratulations-first-ever-edx-prize-finalists

Companion, Developer's track). Seoul, Korea — April 07 - 11, 2014. ACM, 2014. p. 599-602. 〈hal-01083320〉

Serena Villata, Luca Costabello, Fabien Gandon, Catherine Faron Zucker, Michel Buffa. *Social Semantic Network-Based Access Control. Security and Privacy Preserving in Social Networks*, Springer, 2013, Lecture Notes in Social Networks, 〈hal-01187472〉

Michel Buffa, Guillaume Husson, Nicolas Delaforge. SweetDeki : *"Le wiki sémantique couteau suisse du réseau social ISICIL"*. 12e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2012), Jan 2012, Bordeaux, France. pp.387-398. 〈hal-01790117〉 **Honorable mention for the best application paper.** Also published In RNTI Journal, issue EGC 2012, vol. RNTI-E-23, pp.113-124

Pavel Arapov, Michel Buffa, Amel Ben Othmane. *"Semantic Mashup with the Online IDE WikiNEXT"*. WWW '14 Companion Proceedings of the 23rd International Conference on World Wide Web. Pages 119-122 , Seoul, Korea — April 07 - 11, 2014 〈10.1145/2567948.2577010〉. 〈hal-01083331〉

Pavel Arapov, Michel Buffa, Amel Ben Othmane. "*WikiNEXT: a wiki for exploiting the web of data"*. SAC 2014, 29th Symposium On Applied Computing, Mar 2014, Gyeongju, South Korea. pp.8, 2014, Proceeding SAC '14 Proceedings of the 29th Annual ACM Symposium on Applied Computing Pages 727-734. 〈hal-01083338〉

Michel Buffa, Nicolas Delaforge, Guillaume Ereteo, Fabien Gandon, Alain Giboin and Freddy Limpens, *"ISICIL: Semantics and Social Networks for Business Intelligence"*, conference SOFSEM 2013, 39th International Conference on Current Trends in Theory and Practice of Computer Science. January 26–31, 2013 Špindlerův Mlýn, Czech Republic.

Michel Buffa, Nicolas Delaforge, Fabien Gandon , *ISICIL : Sémantique et réseaux sociaux pour instrumenter la veille dans l'entreprise 2.0*, article présenté en démo session, Ingénierie des Connaissances IC'11, Chambery, 16-20 Mai 2011.

Buffa M., Faron-Zucker C. (2012) *Ontology-Based Access Rights Management*. In : Guillet F., Ritschard G., Zighed D. (eds) Advances in Knowledge Discovery and Management. Studies in Computational Intelligence, vol 398. Springer, Berlin, Heidelberg. (2011).

Pavel Arapov, Michel Buffa, Sébastien Comos, Olivier Corby, Nicolas Delaforge, Catherine Faron-Zucker, Fabien Gandon, Guillaume Husson and Freddy Limpens, *ISICIL : Intégration Sémantique d'Informations à travers des Communautés d'Intelligence en Ligne*, Démo paper, 11ème Conférence Internationale Francophone "Extraction et Gestion des Connaissances" EGC'10, Brest, France, 23-28 Janvier 2011.

Michel Buffa, Catherine Faron-Zucker, *Gestion sémantique des droits d'accès au contenu de sites web: l'ontologie AMO*, 10ème Conférence Internationale Francophone "Extraction et Gestion des Connaissances" EGC'10, Hammamet, Tunisie, 26-29 Janvier 2010.

Résumé de cette section

Première partie : intégration de la sémantique dans un Wiki programmable, captation d'activité en RDF pour intégration dans un réseau social

Les travaux présentés dans cette section ont été réalisés dans le cadre du projet ANR ISICIL, dont j'étais responsable scientifique au sein de l'équipe KEWI d'I3S. Nos contributions visaient à renouveler les pratiques de veille en entreprise, notamment par la mise en œuvre d'un ensemble d'outils issus du web 2.0 (réseaux sociaux, social tagging, social bookmarking "à la delicious", blogs, wikis, extensions de navigateur, notifications, etc.) basés sur les technologies du web sémantique. Le modèle du réseau social ainsi que les outils d'analyse de réseaux sociaux sont issus des travaux de la thèse de Guillaume

Erétéo, que j'ai co-encadrée, les outils de structuration semi-automatiques des tags sont issus des travaux de la thèse de Freddy Limpens que j'ai également co-encadrée, alors que le wiki sémantique est une évolution des travaux que j'ai menés depuis 2005 et présentés dans les chapitres précédents de ce manuscrit.

Ces outils ont été intégrés au sein du framework ISICIL, composé d'un ensemble de web services, d'agents de calculs et de logiciels clients (extensions pour navigateurs, applications web, clients pour smartphones).

Dans un premier temps, nous détaillerons le framework ISICIL développé dans le cadre du projet, puis dans les sections suivantes nous nous consacrerons à l'étude du wiki développé à cette occasion –SweetDeki- en montrant en quoi il a amélioré l'état de l'art sur les wikis sémantiques de l'époque avant de détailler son architecture et ses fonctionnalités.

SweetDeki est en effet le wiki sémantique intégré dans le framework du réseau social ISICIL dédié à la veille en entreprise. L'originalité de ce wiki est double : 1) les activités des utilisateurs enrichissent le réseau social, par exemple une coédition ou un commentaire déposé sur une page, génère une relation de collaboration, les activités des utilisateurs au sein du wiki sont recoupées avec leurs actions au sein d'autres outils utilisés dans le projet ; et 2) le wiki sert aussi de support de publication pour des synthèses de toute l'activité ISICIL (diagrammes, courbes, tableaux, flux d'activité, tendances, statut de l'auteur).

Ces synthèses sont intégrables dans des documents du wiki. Par exemple, un rapport sur les "pompes à chaleur" pourra inclure une visualisation de l'ensemble des personnes travaillant sur le sujet, le profil détaillés des auteurs ayant publié du contenu sur le sujet ou sur un sujet relié, etc. En d'autres termes, il permet de qualifier l'information publiée dans le wiki (mesurer le niveau de fiabilité de l'information présentée en fonction des auteurs/contributeurs), il assure la fonction d'édition collaborative de documents et il exploite les activités individuelles des utilisateurs (y compris avec d'autres outils).

Les informations complémentaires sur les auteurs, leur importance au sein de la communauté, leur activité, etc., sont calculées à partir des outils d'analyse du réseau social. Par ailleurs le système de fichiers versionné du wiki sert aussi à archiver des instantanés de documents capturés sur le web (des "web scraps") en vue là aussi de réutilisation et partage dans des documents collaboratifs.

Seconde partie : comment faciliter la programmation et l'apprentissage d'applications pour le Web Sémantique ? La thèse de Pavel Arapov que j'ai encadrée à 100% a consisté à prendre le meilleur des wikis sémantiques (création de documents enrichis sémantiquement) et le meilleur des IDEs en ligne (permettant de coder directement dans un navigateur, et de voir rapidement le résultat) pour le développement d'applications tirant parti au maximum des technologies du Web sémantique et du Web d'applications.

Le wiki imaginé, conçu et testé dans cette thèse, WikiNEXT était donc un wiki à la croisée des wikis sémantiques et des outils de développement en ligne apparus à l'époque (« web

based IDEs »). En permettant de coder directement dans le navigateur des applications exploitant le web de donnée et manipulant des données sémantiques, WikiNEXT a étendu le concept de wiki sémantique et répondait au problème *« comment faciliter la programmation et l'apprentissage d'applications pour le web sémantique ? »*.

WikiNEXT s'adressait à plusieurs profils d'utilisateurs, cependant son originalité résidait surtout dans les aspects « wiki programmable » qui concernent principalement les développeurs web. Les wikis sémantiques de l'époque permettaient principalement d'insérer du contenu dynamique dans les pages, mais ne partageaient pas cette approche programmable / dynamique. Dans cette section nous montrons en quoi WikiNEXT a amélioré l'état de l'art de l'époque dans le domaine. L'outil a servi à l'époque à développer de nombreux tutoriaux interactifs, notamment pour montrer comment exploiter le SPARQL endpoint de DBPedia.fr.

WikiNEXT a été initialement créé pour écrire des documents qui intègrent des données provenant de sources de données externes du Web de données, telles que DBPedia.org ou FreeBase.com, ou pour écrire des didacticiels interactifs (par exemple un tutoriel HTML5, un tutoriel de programmation web sémantique) qui mélangent du texte et des exemples interactifs dans la même page.

Le système combine des aspects puissants des (i) wikis, tels que facilité d'utilisation, collaboration et ouverture, (ii) du web sémantique tels que rendre l'information traitable par des machines et (iii) des IDEs basés sur le Web permettant de faire du "live coding".

WikiNEXT est destiné à l'écriture de documents / pages ainsi qu'à l'écriture d'applications Web manipulant des données sémantiques, soit localement, soit à partir du Web de données. Ces applications peuvent être créées, éditées ou clonées dans le navigateur et peuvent être utilisées pour intégrer des visualisations de données dans des pages wiki, pour annoter du contenu avec des métadonnées, ou pour tout type de traitement. WikiNEXT est particulièrement adapté à l'enseignement des technologies Web ou à l'écriture de documents intégrant des données provenant du Web de données.

Dans cette section nous détaillerons les idées principales qui ont mené à la conception à partir de zéro de ce wiki (contrairement au wiki sémantique d'application du projet ISICIL, qui était bâti sur un moteur de wiki existant), puis nous insisterons sur sa conception (il intègre un moteur RDF/SPARQL côté client et côté serveur, il exploite les WebSocket HTML5 permettant des traitements locaux ou distants de manière transparente), et sur des cas d'utilisation.

Nous verrons en quoi il diffère des autres wikis sémantiques, et comment il s'inscrit dans la continuité de SweetDeki, le wiki sémantique du projet ANR ISICIL, tout en étant beaucoup plus expérimental.

Troisième partie : gestion centralisée des droits d'accès aux ressources, l'ontologie AMO (Access Management Ontology)

Dans cette section nous présentons nos travaux sur la gestion sémantique des accès aux ressources du réseau social ISICIL : un état de l'art sur les modèles de gestion des accès,

l'ontologie AMO que nous avons développée, et l'implémentation de la gestion des droits à l'aide d'annotations, de requêtes SPARQL et de règles.

4.1 Intégration de la sémantique dans un Wiki programmable, captation d'activité en RDF pour intégration dans un réseau social : le wiki SweetDeki du projet ANR ISICIL

Résumé

Le projet ANR ISICIL [105] mixe les nouvelles applications virales du web avec des représentations formelles et des processus d'entreprise pour les intégrer dans les pratiques de veille en entreprise. Plus précisément, ISICIL propose de concevoir, d'étudier et d'expérimenter l'utilisation de nouveaux outils d'assistance aux tâches d'intelligence en entreprise. Ces outils s'appuient sur les interfaces avancées des applications du web 2.0 (blog, wiki, social bookmarking, extensions de navigateurs) pour les interactions et sur les technologies du web sémantique pour l'interopérabilité et le traitement de l'information. Le présent article décrit plus précisément le wiki sémantique développé dans le cadre de ce projet et son intégration au cœur du framework ISICIL. Nous verrons que ce wiki sémantique programmable permet notamment (en plus de ses fonctionnalités classiques de wiki) de réaliser des tâches de reporting automatique, de suivi de l'activité des utilisateurs du réseau social ISICIL (en proposant de nombreuses visualisations graphiques paramétrables), d'archivage automatique de ressources du web en vue de partage et intégration dans des documents ("web scraping"), il est par ailleurs connecté au réseau social et génère des annotations sémantiques pour chaque action effectuée, par exemple lorsque deux personnes ont co-édité une page du wiki, on génère une annotation indiquant que ces deux personnes ont collaboré; le wiki étant connecté au serveur de tags du projet, on saura de quel sujet parle la page et à quoi s'intéressent ces deux personnes. Les outils décrits dans cet article vont ont été déployés fin 2011 chez l'ADEME et Orange Labs.

4.1.1 Introduction

Le projet ANR ISICIL [105], vise à renouveler les pratiques de veille en entreprise, notamment par la mise en œuvre d'un ensemble d'outils issus du web 2.0 (réseaux sociaux, social tagging, social bookmarking, blogs, wiki, extensions de navigateur, notifications, etc.) basés sur les technologies du web sémantique. Le modèle du réseau social ainsi que les outils d'analyse de réseaux sociaux sont issus des travaux de la thèse de Guillaume Ereteo [106], les outils de structuration semi-automatiques des tags sont issus des travaux de thèse de Freddy Limpens [170] (thèses que j'ai co-encadrées) alors que le wiki sémantique est une évolution des travaux que j'ai menés depuis 2005 et qui font l'objet des chapitres précédents de ce manuscrit. Ces outils sont intégrés au sein du framework ISICIL, composé d'un ensemble de web services, d'agents de calculs et d'un ensemble de clients (extensions Firefox, applications web, clients pour smartphones). Dans la section 4.1.2 nous détaillerons le framework ISICIL, puis dans les sections suivantes nous nous

consacrerons à l'étude du wiki en présentant tout d'abord une mise à jour de l'état de l'art sur les wikis sémantiques (section 4.1.3) avant de détailler son architecture et ses fonctionnalités (section 4.1.4). Enfin, nous conclurons avec des évaluations effectuées lors des tests d'utilisation du framework (et donc du wiki) avec un déploiement chez Orange Labs et l'ADEME en 2012.

4.1.2 Le framework ISICIL

Le projet ISICIL vise à moderniser les pratiques existantes, en s'inspirant des pratiques du web 2.0 et en élargissant la portée des outils à un cercle plus large que celui des seuls "veilleurs de métier". Les outils développés facilitent le partage, la recherche et l'exploration de ressources (documents, personnes). Parmi les tâches que les outils du projet doivent accomplir certaines sont très importantes : "*trouver les bonnes personnes pour constituer un groupe d'experts sur un sujet donné*" et "*proposer un système de recherche et de navigation efficace pour y parvenir*".

ISICIL propose :

- Intégration de sources de données hétérogènes,
- Authentification unique sur l'ensemble des outils,
- Réseau social "à la Facebook",
- Suggestion de contacts basée sur les intérêts,
- Gestion de tags pour classifier les ressources (personnes, documents, bookmarks) incluant un système de structuration semi-automatique [30],
- Outils d'analyse du réseau social (calculs des principaux indices tels que diamètre, centralités, extraction de communautés [106],
- Extension Firefox dédiée et clients mobiles permettant de faire des recherches (de documents en texte plein, de personnes ou de documents par tags), d'explorer le réseau des tags ou des personnes,
- Des notifications,
- Un suivi d'activité de personnes,
- Un outil de bookmarking [107] incluant tagging, géolocalisation et copies / archivage des documents (web-scrapping), et un système de contrôle d'accès aux ressources basé sur l'ontologie S4AC [108].

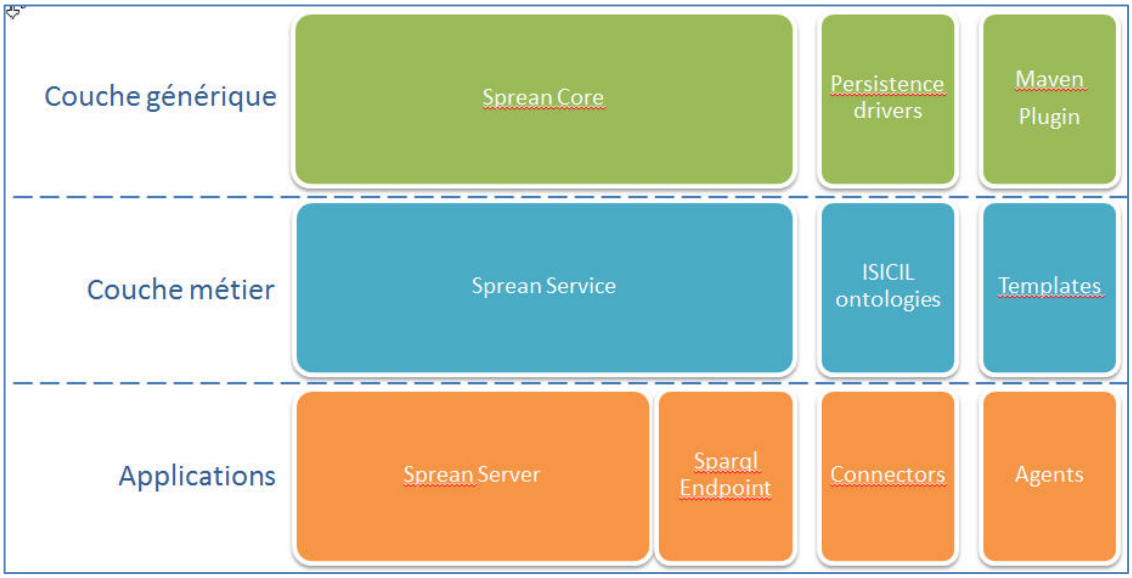


Figure 30. Couches logicielles du server SPREAN du projet ISICIL

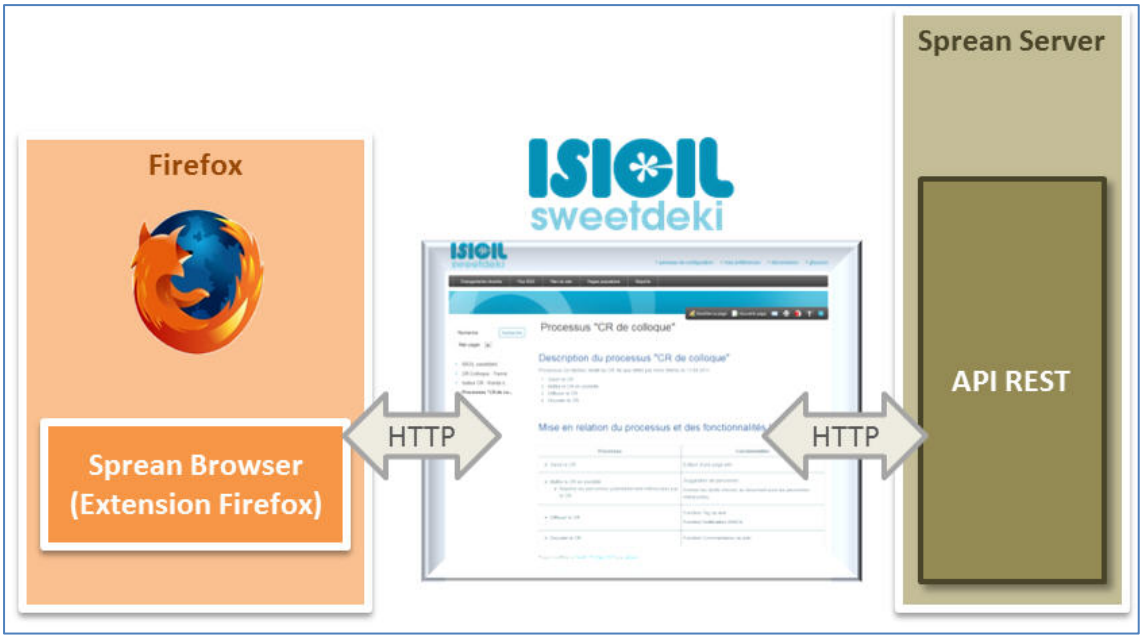


Figure 31. Positionnement de SweetDeki, le wiki d'ISICIL

Le framework ISICIL s'articule autour du serveur SPREAN[25] exposant un ensemble de web services. Ce dernier comporte trois niveaux d'abstraction : une couche générique, une couche métier et une couche pour les applications. La gestion des annotations sémantiques (au format RDF) est à la charge de la couche de plus bas niveau (en vert dans la Figure 30). Elle utilise le moteur sémantique CORESE pour la gestion de requêtes SPARQL et un SGBD relationnel pour implémenter un système transactionnel au-dessus de RDF. La couche métier (en bleu) fait appel à l'ontologie ISICIL (qui réutilise et étend

[25] Pour Semantic Platform for REsources sharing And social Networking

une vingtaine d'ontologies populaires comme FOAF[26], SIOC[27], SCOT[28] etc.) pour modéliser et implémenter le réseau social, la gestion des documents manipulés, du social tagging, des droits d'accès, de la mesure de confiance entre acteurs du réseau.La couche d'application (en orange) fournit une API REST, et un point d'accès SPARQL.

Le wiki sémantique SweetDeki s'interface avec le serveur SPREAN et l'extension Firefox ISICIL, comme illustrés par la Figure 31.

4.1.3 Wikis sémantiques : rappels et état de l'art réactualisé en 2010

Nous l'avons vu dans les chapitres précédents, les wikis sémantiques utilisent les technologies et langages du web sémantique pour formaliser les ressources qu'ils manipulent, et entre 2005 et 2010 de nombreux moteurs de wiki sémantiques ont été proposés par la communauté. Le lecteur pourra se référer au chapitre 3 de ce manuscrit pour un état de l'art complet de la première génération de wikis sémantiques (voir aussi l'état de l'art fait par Orlandi [113]). On retrouve en général deux approches, parfois utilisées conjointement : celle considérant le wiki comme un outil de maintenance d'ontologies (les pages représentant des concepts et les liens typés des propriétés), les moteurs les plus connus ayant émergé de cette première vague étant Semantic Media Wiki [109], IkeWiki [110] et OntoWiki [111], et celle permettant l'utilisation d'ontologies par le wiki, pour annoter des documents ou pour formaliser le modèle même du wiki (pages, utilisateurs, liens, etc.) comme l'avait proposé le moteur SweetWiki [34]. Ces wikis sémantiques se comportent parfois comme des "bases de données sémantiques" et intègrent des outils comme JENA (IkeWiki) ou CORESE (SweetWiki) permettant de répondre à des requêtes et implémenter des fonctionnalités telles que la navigation ou la recherche d'information, plus abouties et "intelligentes" que dans les wikis classiques (suggestions, inférences, etc.).

Ainsi que le font remarquer Meilender and al. [112] dans un état de l'art centré sur les wikis sémantiques encore actifs fin 2011, les plus populaires se sont enrichis de nombreuses fonctionnalités (extension Halo[29] pour Semantic MediaWiki - qui propose des outils plus ergonomiques pour l'édition de formulaire ou pour l'annotation de documents) ou au contraire, ont externalisé certaines parties comme KiWi (Knowledge In Wiki) [114], successeur d'IkeWiki, qui privilégie un framework plutôt qu'un unique outil.

SweetDeki, le wiki développé pour le projet ISICIL n'est pas utilisé pour créer des ontologies comme permet de le faire OntoWiki, Semantic Media Wiki et dans une moindre mesure IkeWiki/Kiwi. Ce n'est pas non plus un outil d'annotation de contenu à partir d'ontologies externes. A la manière des wikis cités, il utilise des ontologies pour modéliser les contenus (pages, documents attachés, images, vidéos, révisions, etc.), les descripteurs du contenu (par exemple les tags, les commentaires laissés sur les pages, les

[26] http://www.foaf-project.org/
[27] http://sioc-project.org/
[28] http://www.scot-project.org/
[29] http://www.projecthalo.com/

recommandations) et les utilisateurs. Ces modèles, ainsi que les métadonnées sont exploités par le moteur sémantique CORESE interfacé avec le wiki via des Web Services.

Ce qui distingue SweetDeki :

- Il est adossé au réseau socio-sémantique ISICIL,
- Ses utilisateurs, auteurs, contributeurs bénéficient d'un modèle riche (réutilisant des ontologies populaires telles que FOAF ou SIOC) décrivant leurs interactions.
- Il utilise un système de tagging hybride folksonomie/thésaurus basé sur la détection automatique de relations entre tags.

Nous détaillerons ces points dans les sections suivantes.

4.1.4 SweetDeki : le couteau suisse du réseau social ISICIL

Une approche décentralisée

SweetWiki [34] a été un des premiers wiki sémantiques (2005) à proposer un éditeur WYSIWYG pour l'édition à la fois des données et des métadonnées, l'intégration de requêtes SPARQL dans un document, l'utilisation d'un modèle sémantique du wiki lui-même, l'intégration d'un moteur de raisonnement et d'un outil de structuration des tags. Bien que novateur, ses points faibles étaient surtout liés aux fonctionnalités "classiques" des wikis : éditeur peu fiable, gestion de la sécurité non centralisée avec le réseau de l'entreprise, etc.

Ainsi, au démarrage du projet ISICIL, il a été décidé pour SweetDeki (l'évolution de SweetWiki) :

1. D'externaliser les fonctionnalités « sémantiques » du wiki dans le serveur SPREAN déporté, avec lequel le wiki communiquera via des Web Services,
2. De construire un ensemble d'extensions au-dessus d'un moteur de wiki de qualité industrielle au lieu de partir de la base logicielle de SweetWiki, qui avait été écrit de zéro.

Notre choix s'est porté sur le logiciel open source Mindtouch Core[30], édité par la société mindtouch.com. Les raisons pour lesquelles nous avons choisi ce logiciel sont les suivantes :

- Édition WYSIWYG conviviale, fiable (à la Microsoft Word) et extensible,
- API REST très complète (et extensible) permettant de contrôler le wiki depuis des applications externes,
- Personnalisation aisée (*look and feel*, système d'authentification, etc.),
- Extensibilité par plugins permettant entre autres de rendre le wiki « client » de web services externes,

[30] Le premier nom de Mindtouch Core était « DekiWiki », ce qui explique l'origine du nom SweetDeki, mélange de SweetWiki et de DekiWiki. SweetWiki avait été développé "from scratch", SweetDeki s'appuie sur Mindtouch Core, créé par la société MindTouch, créée par d'anciens employés de Microsoft.

- Possibilité d'installer des "hooks" pour intercepter et modifier la sauvegarde et le rendu des documents à la volée. Le wiki peut ainsi devenir un outil de reporting pour des applications externes, les rapports générés étant des pages du wiki.

Nous le verrons dans les sections suivantes, ces caractéristiques ont facilité l'intégration du wiki avec le framework ISICIL.

```
<div property="sioc:has_creator"
  content="http://ns.inria.fr/isicil/id/spreanaccount/ghusson">
</div>
<div about="http://ns.inria.fr/isicil/id/person/ghusson"
  property="foaf:account" content="http://ns.inria.fr/isicil/id/spreanaccount/ghusson">
        <div property="foaf:firstname"  content="Guillaume"></div>
        <div property="foaf:familyname" content="HUSSON"></div>
</div>
```

Figure 32. Exemple de RDFa présent dans le XHTML des pages de SweetDeki.

```
<isicil:WikiPage rdf:about="http://sabre.inria.fr/ADEME/ISICIL"/>

<isicil:WikiPageRevision
  rdf:about="http://sabre.inria.fr/index.php?title=ADEME/ISICIL&amp;revision=4">
  <dcterms:title><![CDATA[ISICIL]]></dcterms:title>
    <sioc:has_container rdf:resource="http://sabre.inria.fr/ADEME/ISICIL"/>
  <isicil:revision>4</isicil:revision>
    <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#datetime">
      2011-10-10T18:01:43.488+0200</dcterms:created>
    <sioc:has_creator
       rdf:resource="http://ns.inria.fr/isicil/id/spreanaccount/mbuffa"/>
  <skos:hiddenLabel><![CDATA[isicil]]></skos:hiddenLabel>
</isicil:WikiPageRevision>

<rdf:Description rdf:about="http://ns.inria.fr/isicil/id/person/mbuffa">
  <rel:collaboratesWith rdf:resource="http://ns.inria.fr/isicil/id/person/ghusson"/>
</rdf:Description>

<isicil:SocialAction rdf:about="http://ns.inria.fr/isicil/id/action/1318262204703">
  <sioc:has_creator rdf:resource="http://ns.inria.fr/isicil/id/spreanaccount/mbuffa" />
  <dc:title><![CDATA[Nouvelle relation]]></dc:title>
  <dc:description><![CDATA[${Michel BUFFA, michel.buffa}
     a ajouté ${Guillaume HUSSON, guillaume.husson} à son réseau social. ]]>
     </dc:description>
  <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#datetime">
    2011-10-10T17:56:44.703+0200</dc:date>
</isicil:SocialAction>
```

Figure 33. Haut : révisions d'une page du wiki. Bas : les annotations générées. On les retrouve dans le graphe du réseau social présenté Figure 35.

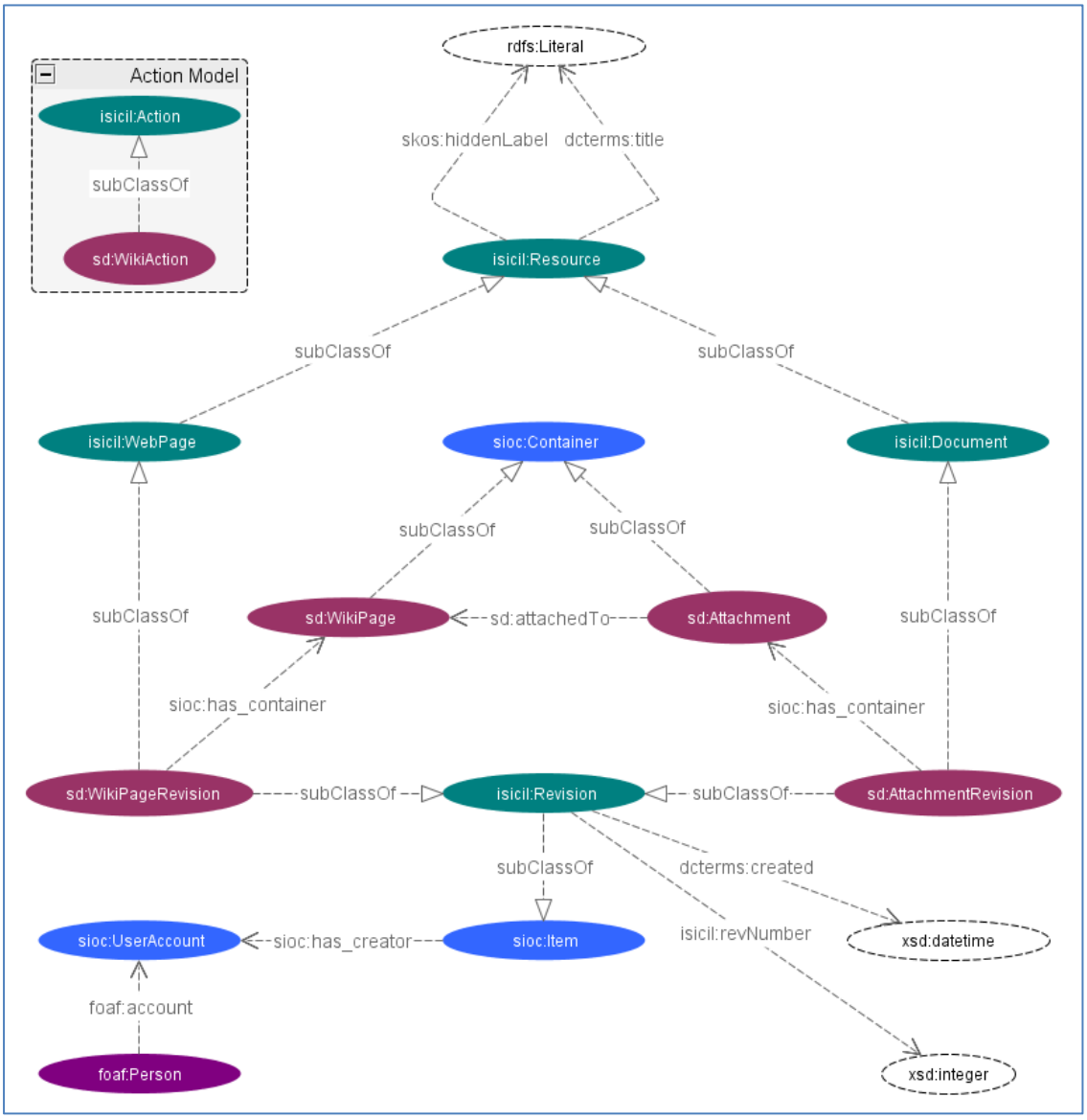


Figure 34. Extrait du modèle de ressources de SweetDeki.

Modèle de ressources

SweetDeki est un wiki sémantique dans le sens où il utilise des ontologies pour formaliser les ressources qu'il met en œuvre (documents, images, vidéo, fichiers attachés aux pages, révisions, utilisateurs, etc.), ainsi les documents qu'il crée et les actions de ses utilisateurs génèrent des annotations sémantiques (voir par exemple les lignes 1-15 de la Figure 33). Ces dernières sont envoyées au fur et à mesure au serveur SPREAN via des requêtes REST. Les métadonnées relatives aux documents du wiki sont par ailleurs réinjectées par le wiki au format RDFa, dans le XHTML des pages, au moment du rendu, ce qui les rend analysables par des moteurs de recherche tels que Google, ou utilisables par des extensions Firefox à même de les détecter. La Figure 32 montre un exemple de RDFa injecté dans une page.

La Figure 34 montre un extrait[31] du modèle de ressources : il se compose de plusieurs ontologies reliées entre elles. En rose foncé on retrouve l'ontologie SweetDeki qui modélise les pages (classe `sd:WikiPage`), les documents que l'on peut attacher (classe `sd:attachement`, relation `sd:attachedTo`) ou inclure dans une page, les révisions (`sd:WikiPageRevision`), et les actions des utilisateurs (en haut à gauche `sd:WikiAction`), notamment celles qui vont nourrir le réseau social, par exemple des actions de collaboration. A noter que les classes de l'ontologie SweetDeki étendent ou réutilisent des classes de l'ontologie SIOC qui modélise l'activité des utilisateurs sur les plateformes du web 2.0 et les documents qu'ils produisent : les documents du wiki sont des sous-classes de `SIOC:container` par exemple et `SIOC:UserAccount` modélise un contributeur ou un créateur de documents. Par ailleurs, les classes correspondant aux documents et aux actions des utilisateurs sont des sous-classes de l'ontologie ISICIL, afin de s'intégrer dans l'écosystème ISICIL et rendre possible des opérations de recherche ou de suivi d'activité impliquant d'autres outils.

Enrichissement du réseau social, suivi d'activité

Les utilisateurs du wiki sont gérés par le modèle d'utilisateur ISICIL et font partie du réseau social. Ce modèle reprend celui créé dans le cadre de la thèse de G. Ereteo [106] et permet notamment d'identifier plusieurs relations entre les membres du réseau (réutilisation des ontologies FOAF et RELATIONSHIPS[32])

Nous allons illustrer ici les annotations de collaboration que génère le wiki lorsque plusieurs personnes coéditent une même page.

La Figure 33 montre les révisions d'une page, les différents contributeurs, ainsi que les annotations générées. On remarque l'apparition de `rel:collaborateWith` (ligne 16) et de `isicil:SocialAction` correspondant à un événement ISICIL[33] pouvant être notifié et pouvant apparaitre dans le suivi d'activité d'une personne.

La Figure 35 montre le réseau social égocentré sur Michel Buffa. On distingue les relations de collaboration existantes entre lui et les contributeurs de la page.

[31] Nous avons omis volontairement la modélisation des tags, pour des raisons de concision.
[32] http://vocab.org/relationship/.html
[33] De telles annotations sont créées pour chaque action, quel que soit l'outil ISICIL utilisé.

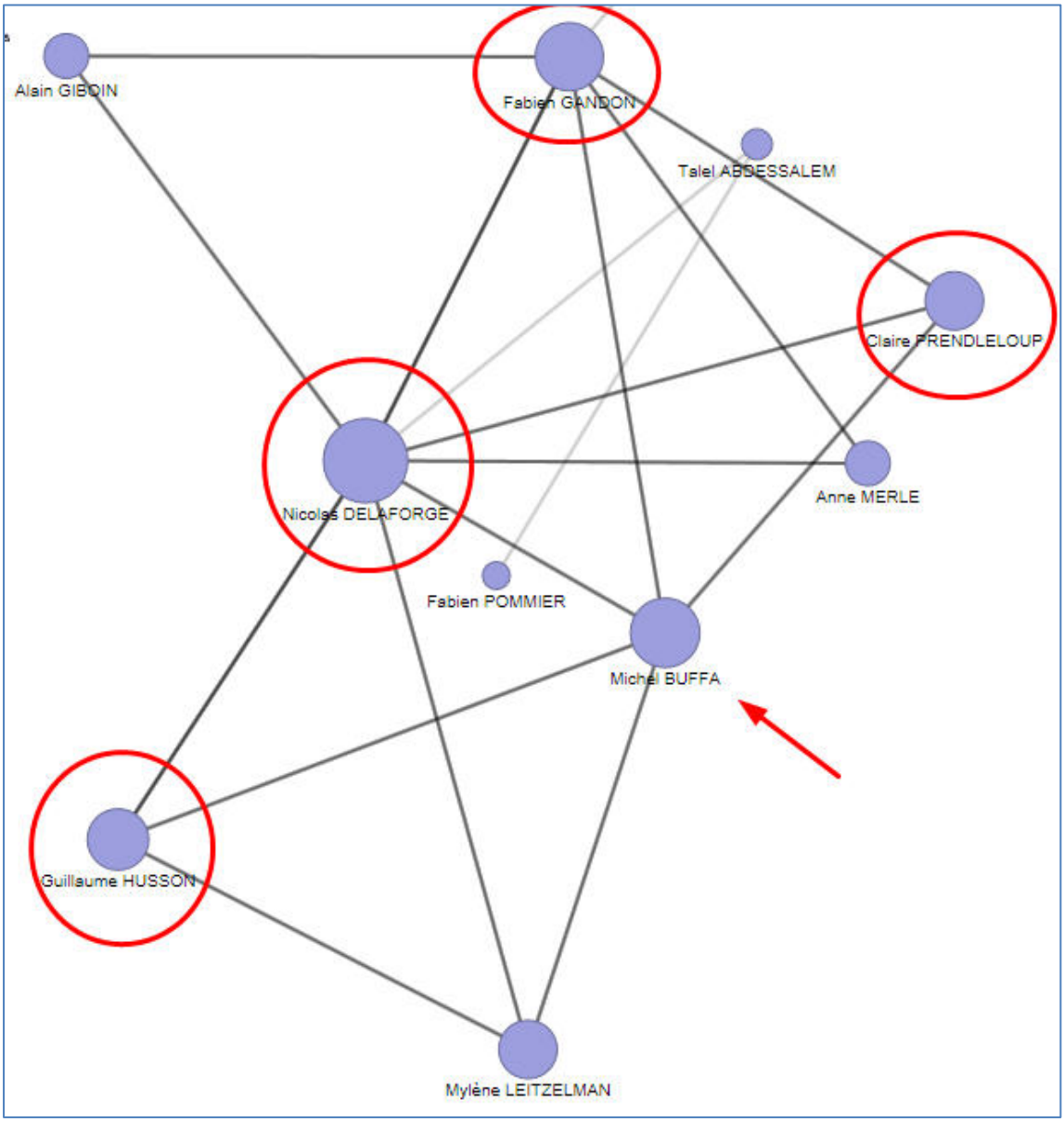


Figure 35. Réseau social ISICIL égocentré, après la révision d'une page du wiki. Les cercles rouges correspondent aux relations générées à partir des différentes coéditions de pages.

Outil de reporting, insertion de visualisations

Un des besoins exprimés par l'ADEME, partenaire du projet, est de pouvoir identifier des réseaux d'experts, de suivre l'activité de certaines personnes, de visualiser les collaborations. Plusieurs extensions du wiki ont été développées, permettant de générer des visualisations graphiques et des rapports d'activité insérables dans n'importe quel document.

Par exemple, on pourra insérer une vue de l'activité du réseau social autour de certains thèmes tels que « la pollution » ou « les pompes à chaleur » (Figure 37). Ces vues peuvent être statiques, dynamiques et interactives (parties cliquables, visualisation par facettes

etc.). Ces visualisations sont des parties intégrantes des pages du wiki, et sont modélisées sous forme d'instances de `sd:Attachment`.

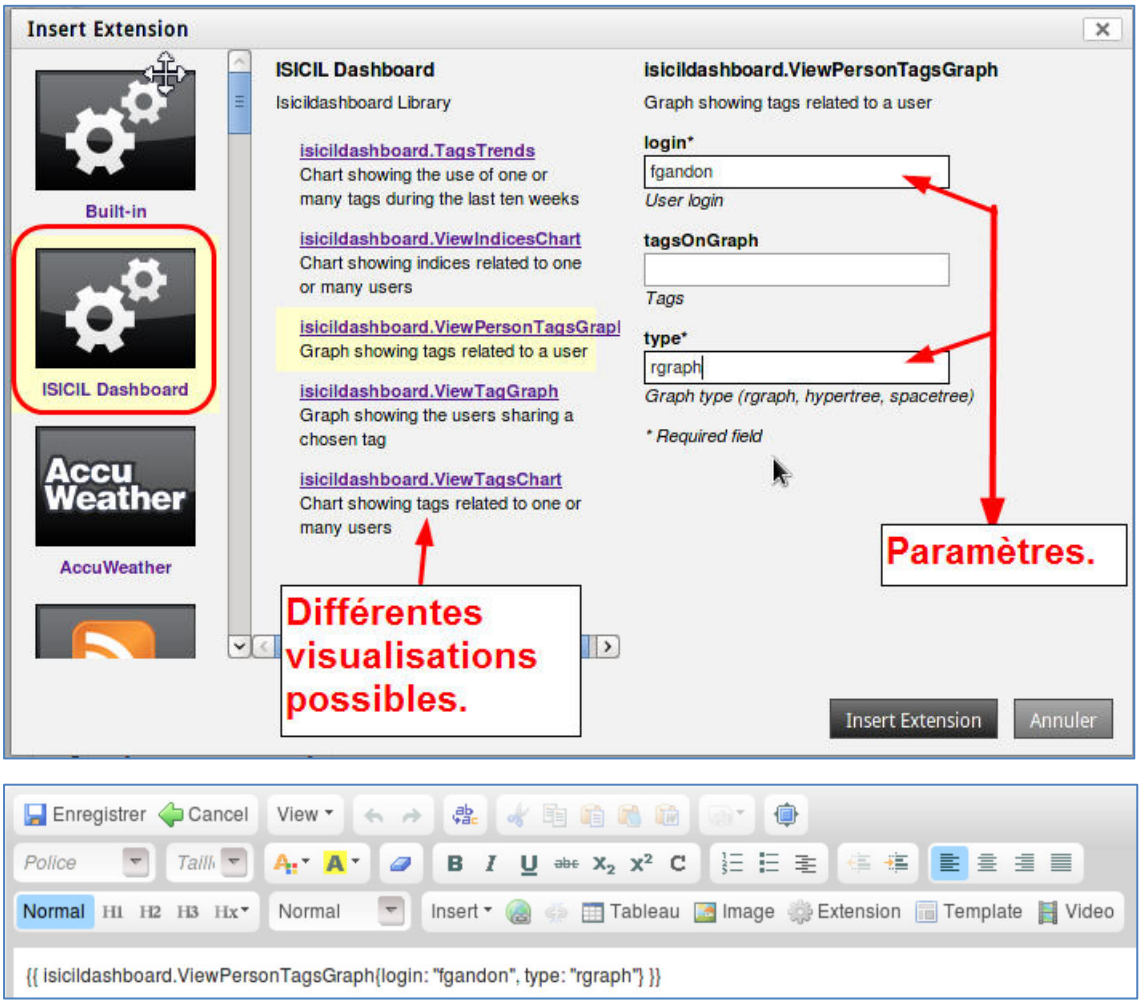


Figure 36. Haut : *wizard* permettant d'insérer une visualisation dans un document. Bas : macro correspondante dans l'éditeur. Au rendu, après sauvegarde de la page, on obtient la vue de la Figure 38.

Figure 37. Génération de rapports dans le Wiki

Des boîtes de dialogues ergonomiques ont été ajoutées à l'éditeur WYSIWYG de SweetDeki et permettent de choisir le type de visualisation et de spécifier les différents paramètres, comme le montre la Figure 36. Les visualisations apparaissent dans le document en mode édition sous la forme de macros que l'on peut copier/coller d'une page à l'autre. Actuellement quatre types de vues sont implémentés : visualisations de tendances (*« quels sont les tags/sujets les plus actifs ? »*), visualisation d'indices de l'analyse du réseau social (degré, centralités, *« quel est l'utilisateur le plus influent ? »,* etc.), activité autour d'une personne *(« sur quel sujet travaille-t-elle ? »*, et *« avec qui partage-t-elle les mêmes intérêts ? »*), les collaborations autour d'un tag *(« qui travaille sur le même sujet que untel ? »*), les partages d'intérêts, les collaborations autour d'une personne (illustré par la Figure 38 : « *sur quoi travaille Fabien et avec qui ? »*).

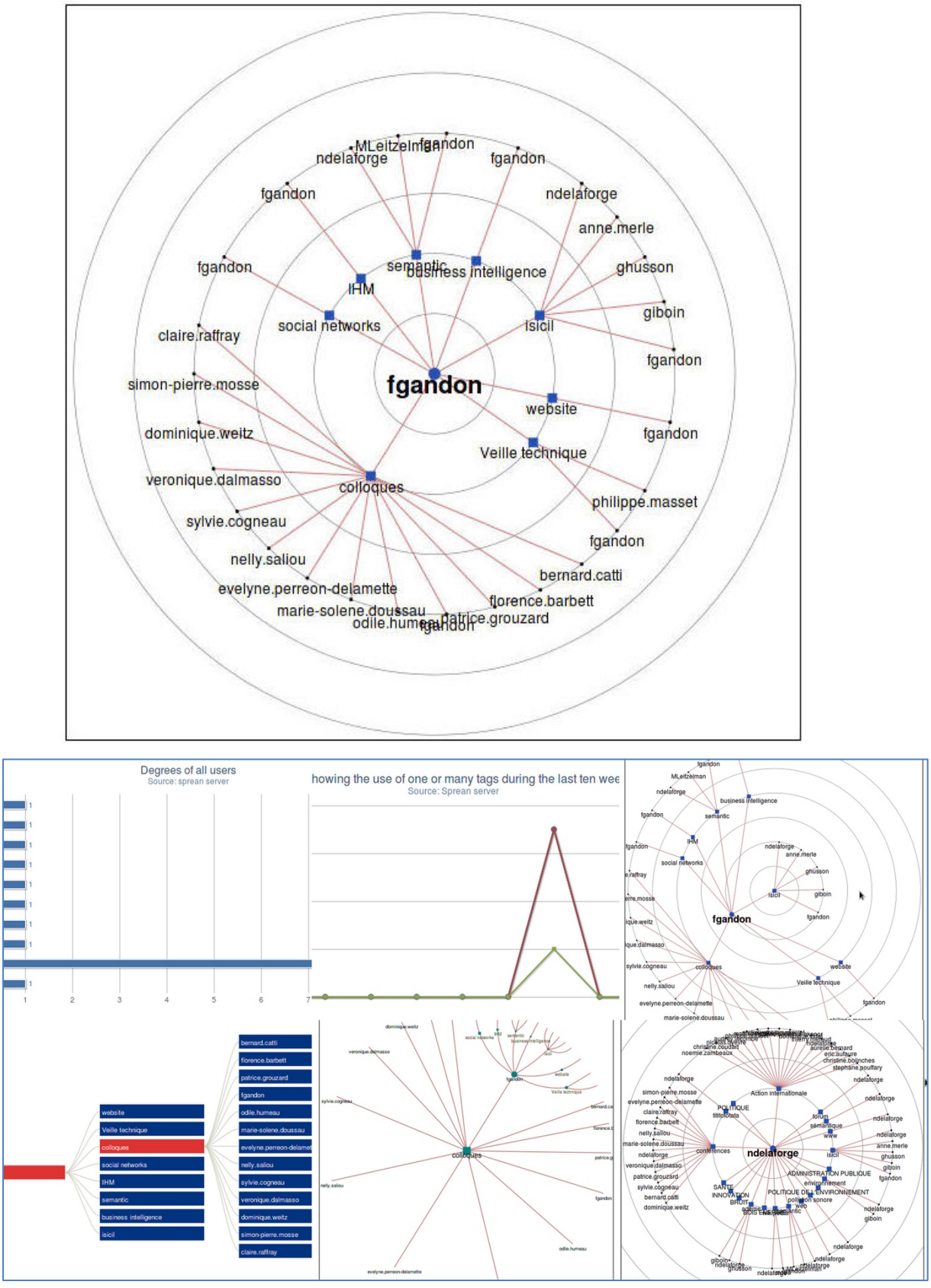


Figure 38. Haut : visualisation de l'activité de Fabien Gandon, on voit sur quoi il travaille (tags) et les personnes qui ont utilisé les mêmes tags. Bas : d'autres visualisations.

Insertion de "copies de bookmarks" (web-scrapping)

Nous avons implémenté des web services dans le wiki permettant à l'extension Firefox ISICIL d'envoyer des copies complètes ou partielles de documents pour archivage et partage/réutilisation. Par exemple, si la « Une » du site web lemonde.fr est pertinente, on peut en faire une "capture" totale ou partielle et l'envoyer au wiki, puis l'insérer dans une page pour illustrer un rapport.

On appelle ces copies des « web scraps ». La page (ou partie de page) capturée est préparée par l'extension Firefox ISICIL sous la forme d'une archive augmentée d'annotations RDF, puis envoyée au wiki par appel d'un web service. Le wiki stocke alors les documents ainsi récupérés sous forme de documents attachés à la page de l'utilisateur courant. Ces "scraps" sont des documents wiki de première classe [107] et peuvent être à leur tour inséré à l'intérieur d'une page de wiki (ils deviennent alors des `sd:Attachment`). Plusieurs vues sont proposées : sous forme d'une iframe XHTML, sous forme d'un lien hypertexte, etc. Les Figure 39 et Figure 40 montrent les différentes interfaces utilisateurs mises en jeu pour la capture ainsi que le le *wizard* permettant d'insérer un document capturé dans un document en cours d'édition.

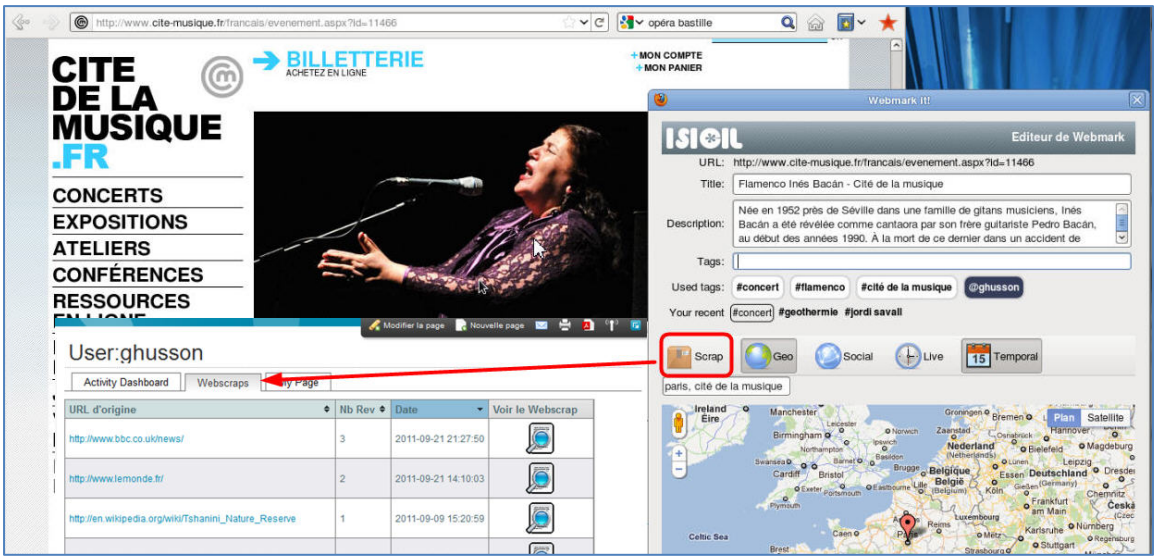


Figure 39. La fenêtre de dialogue à droite est celle de l'extension Firefox ISICIL, permettant de « capturer » des documents ou parties de documents web, de les annoter et les envoyer vers le wiki d'ISICIL pour archivage.

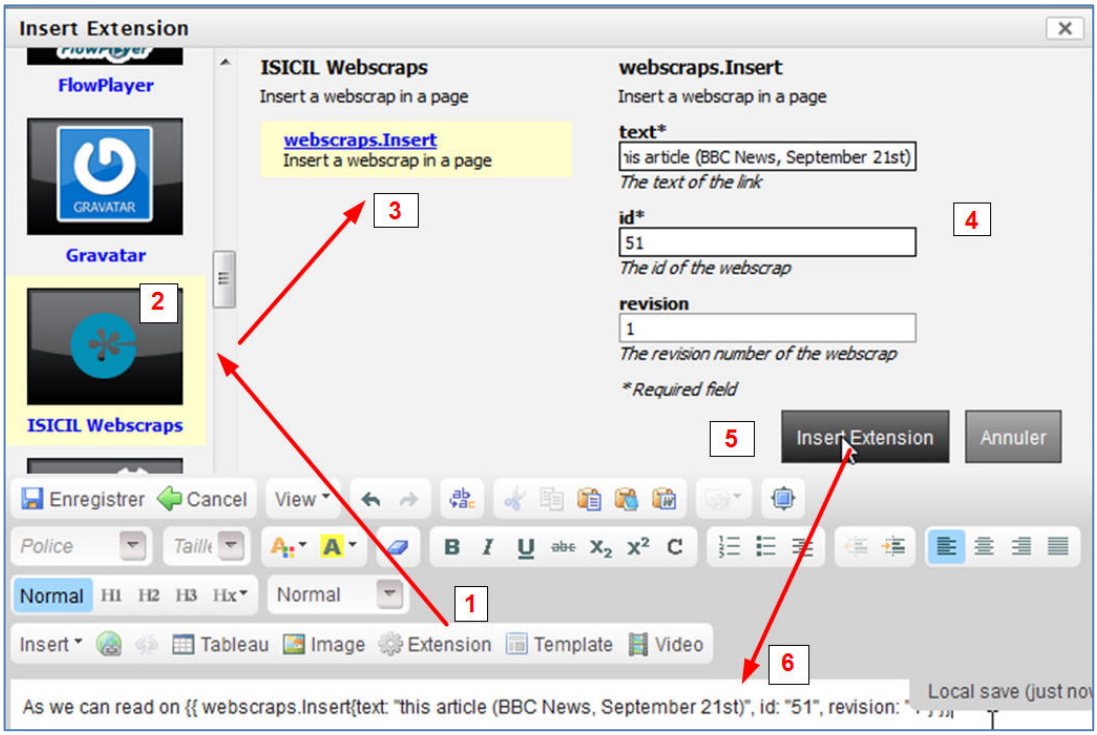


Figure 40. Interface du wiki permettant d'insérer dans une page du wiki un document capturé par l'extension présentée dans la Figure 39.

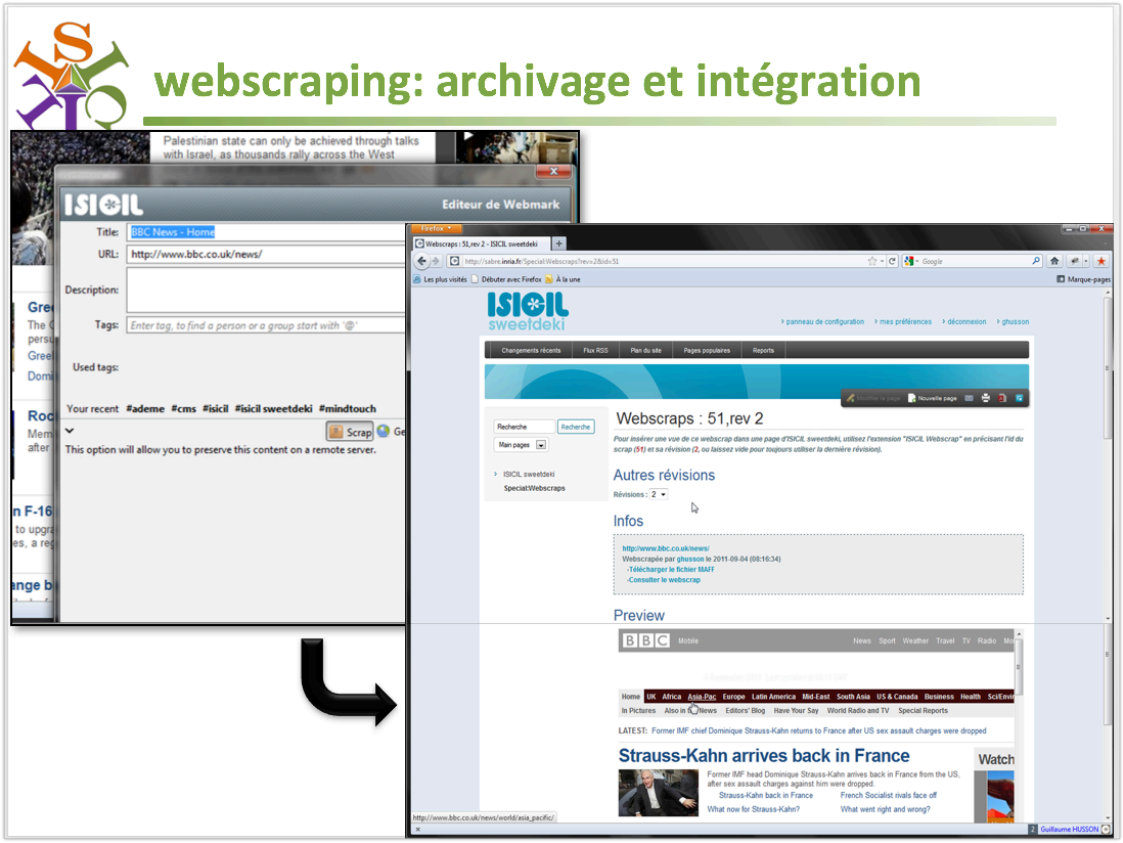


Figure 41. Exemple de document capturé intégré dans une page du Wiki, ici un article sur Dominique Strauss-Kahn.

Suivi d'activités

Une autre macro permet de suivre l'activité d'une personne sur l'ensemble des outils ISICIL (réseau social, wiki, etc.). Cette macro interroge le serveur SPREAN et génère un tableau de bord d'activité (exemple Figure 42). Ce tableau de bord intègre les annotations générées par le wiki mais aussi par les autres outils ; on peut voir qu'une personne a taggué un document, en a bookmarqué un autre, a commenté une page du wiki, etc. La macro peut être insérée dans n'importe quel document du wiki, par exemple dans une page qui résume l'activité d'un groupe de personnes travaillant ensemble. Les interfaces utilisateurs sont similaires à celles présentés pour les fonctionnalités précédentes.

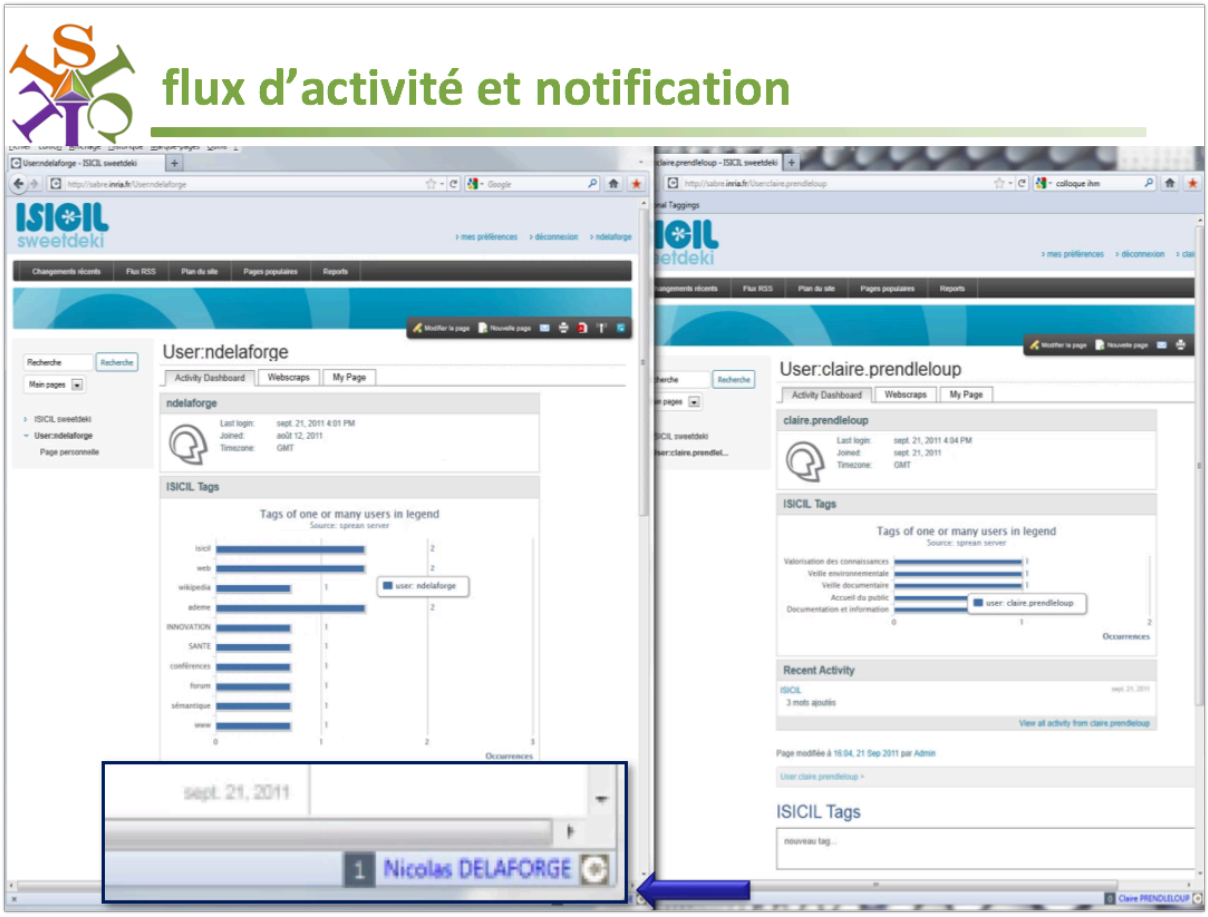


Figure 42. Suivi d'activités de personnes par tags utilisés. Directement programmé dans des pages du wiki d'ISICIL SweetDeki. Les données sont consommées en temps réel depuis le serveur SPREAN du projet ISICIL.

4.1.5 Modalités d'évaluation

Trois phases de tests d'usage des outils ISICIL ont été réalisées. La première phase (Décembre 2011) a eu pour objectif de vérifier que les outils sont utilisables de manière générale et prêts à être mis entre les mains d'utilisateurs ciblés ; elle a impliqué des membres de l'équipe de conception et des représentants des utilisateurs de l'ADEME. La technique d'évaluation utilisée dans cette phase est principalement une technique de relevé de bugs empêchant l'usage. La deuxième phase (à échelle restreinte dans l'organisation cliente) a eu pour objectif de vérifier l'utilisabilité et l'utilité des outils

ISICIL auprès d'un échantillon restreint d'utilisateurs représentatifs des différentes fonctions assurées dans l'organisation cliente. Les utilisateurs ont été invités à réaliser, à l'aide des outils, un certain nombre de tâches significatives pour eux (en l'occurrence des tâches de veille). Les techniques d'évaluation utilisées dans cette phase ont été principalement des techniques issues de l'ingénierie de l'utilisabilité [115] et de la conception de l'interaction [116]. Elle a eu lieu en Janvier et Février 2012. La troisième phase (à échelle moyenne dans l'organisation cliente), avait le même objectif que la précédente, mais différait par le nombre plus grand d'utilisateurs participant au test.

Les résultats ont été publiés dans le livrable « Tests d'usages de la plateforme ISICIL à l'ADEME »[34] et ont montré que l'ensemble des outils, notamment le wiki sémantique, correspondaient aux attentes des veilleurs de l'ADEME.

L'externalisation de la partie « serveur d'annotations et moteur de raisonnement » a été un choix pertinent qui a simplifié le développement et l'articulation des modules implémentant le réseau social. Partir d'un wiki industriel extensible à de multiples niveaux (API, éditeur, hooks pour intercepter la sauvegarde de documents etc.) a été un choix payant : le wiki était plus ergonomique et plus fiable que SweetWiki, tout en ayant les mêmes capacités (à part l'éditeur d'ontologies, qui était externalisé). L'analyse du réseau social et les fonctionnalités développées autour du social tagging et de la gestion de la folksonomie ont été considérablement étendues par rapport à l'approche prototypée dans SweetWiki.

4.2 Comment faciliter la programmation et l'apprentissage d'applications pour le Web Sémantique ? WikiNext !

Résumé : WikiNEXT est un wiki à la croisée des wikis sémantiques et des outils de développement en ligne récents (« web based IDEs »). En permettant de coder directement dans le navigateur des applications exploitant le web de donnée et manipulant des données sémantiques, WikiNEXT étend le concept de wiki sémantique et répond au problème *« comment faciliter la programmation et l'apprentissage d'applications pour le web sémantique ? »*. WikiNEXT s'adresse à plusieurs profils d'utilisateurs, notamment les utilisateurs classiques d'un wiki désirant bâtir un corpus documentaire de manière collaborative, etc. WikiNEXT possède toutes les caractéristiques d'un wiki classique (en particulier celles décrites dans la section 2.3.1). Cependant, il va plus loin : la partie décrite ici se focalisera sur les aspects « wiki programmable » qui concernent principalement les utilisateurs ayant des compétences (même légères) en développement web (HTML/CSS/JavaScript). Les wikis sémantiques actuels permettent d'insérer du contenu dynamique dans les pages mais nous montrerons en quoi WikiNEXT améliore l'état de l'art dans le domaine. A l'époque où cet outil était en ligne, il contenait de de nombreux tutoriaux interactifs, dont un pour apprendre à écrire des applications interactives exploitant DBPedia.fr. On retrouve encore sur YouTube des screencasts et vidéos de démonstration de WikiNEXT.

[34] http://isicil.inria.fr/v2/res/docs/livrables/LivrableTestsUsageISICILAdeme.pdf

4.2.1 RDF comme structure de données de base, JavaScript comme langage de programmation pour la couche applicative

Le problème général auquel WikiNEXT essaie de répondre est « *comment programmer plus facilement des applications exploitant le web de données et les données sémantiques ?* »

Par exemple, nous voulons construire un graphique de la population des pays n'ayant pas de frontière maritime, et ayant plus de 1.500000 habitants, à partir de données de DBpedia.org (Figure 43).

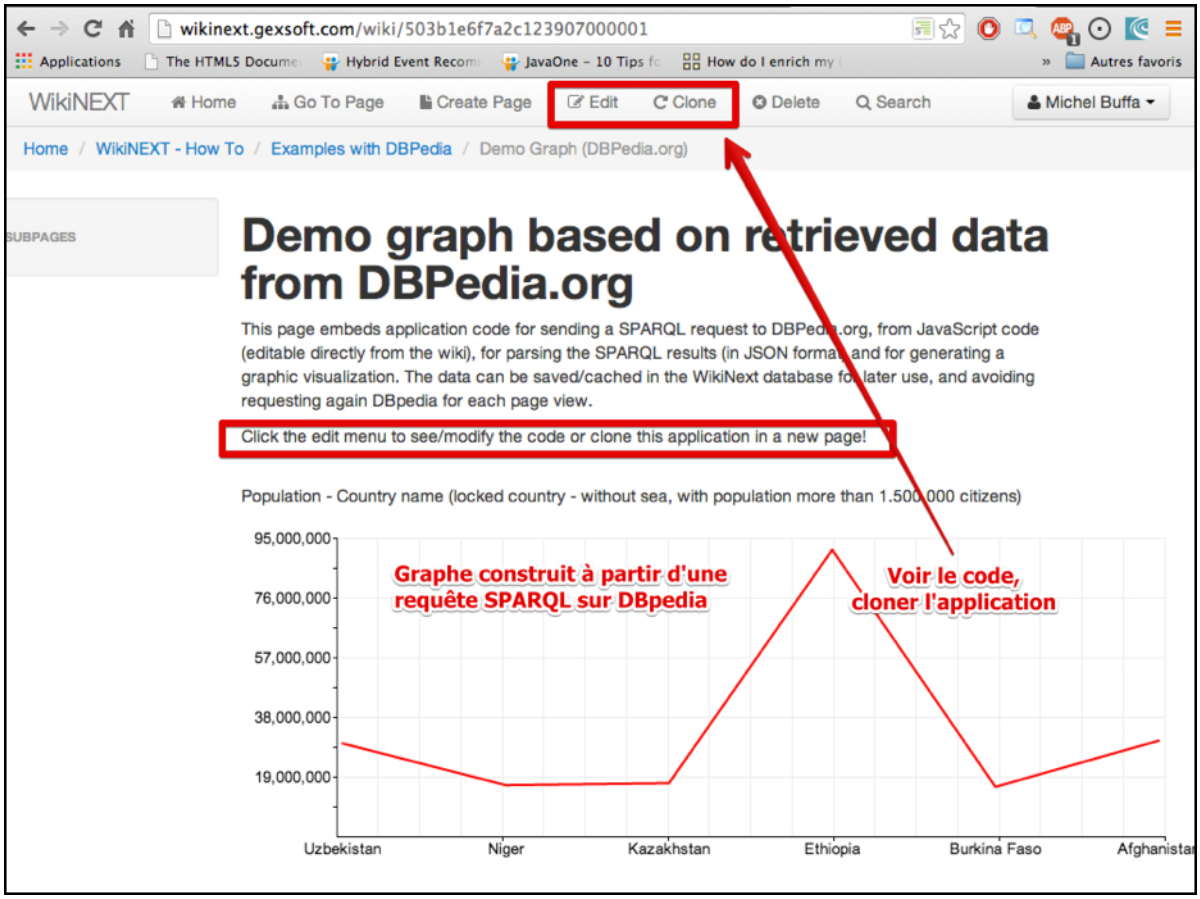


Figure 43. Exemple de contenu dynamique dans WikiNext.

… on pourra utiliser l'éditeur JavaScript/HTML intégré, qui permet de voir le code de l'application, ou encore, on pourra cloner l'application pour l'examiner et la modifier. Cette approche par mimétisme est caractéristique des wikis et demande peu d'expertise (Figure 44). Bien entendu, si on désire uniquement modifier le texte de la page, un troisième éditeur est disponible : un éditeur WYSIWYG HTML, qui se limitera aux parties « non dynamiques » de la page.

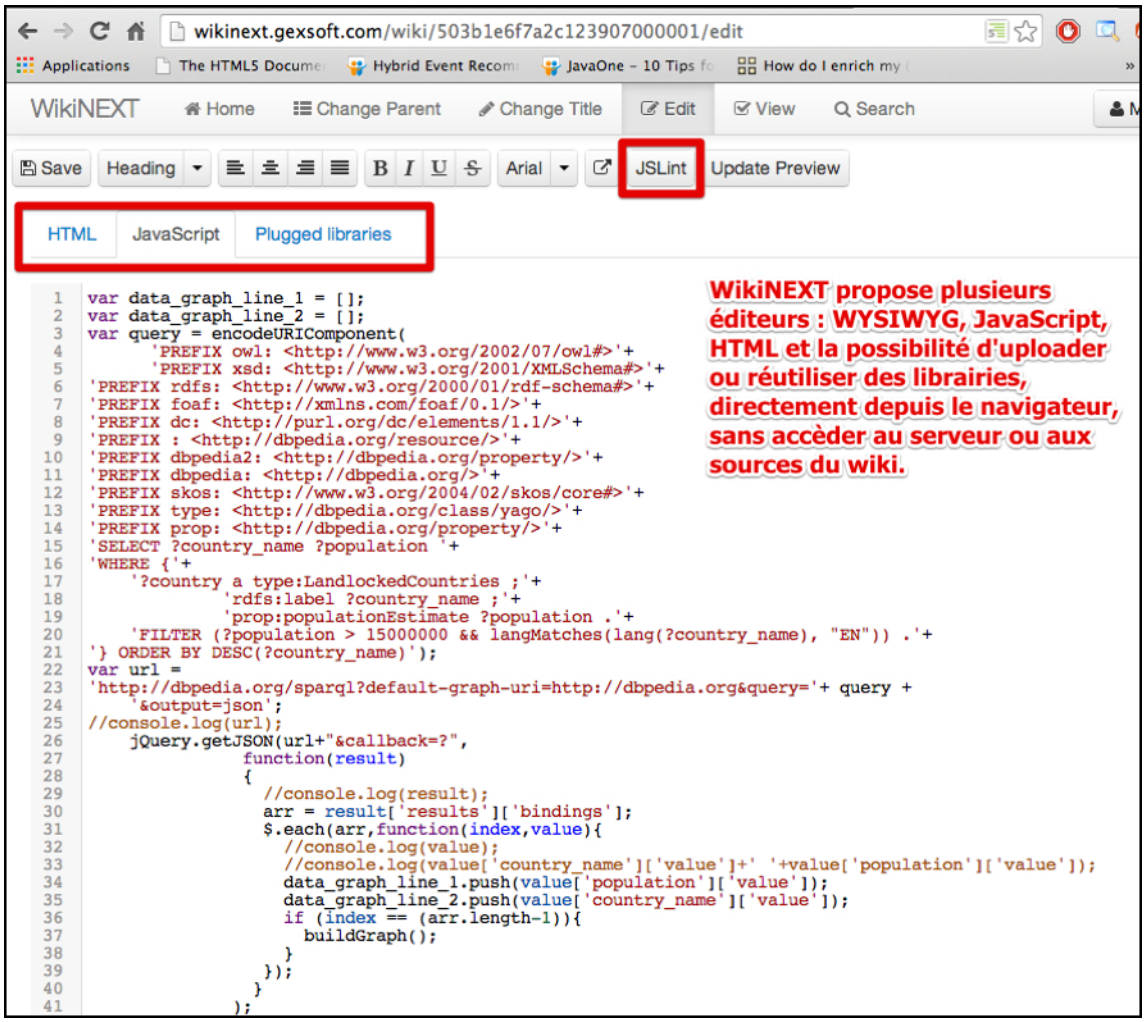


Figure 44. L'envers du décor de l'application web / page du wiki présentée dans la Figure 43. Ici on utilise l'éditeur JavaScript embarqué dans WikiNEXT pour modifier la partie « applicative » de la page.

WikiNEXT se programme directement dans le browser et exploite les tendances récentes des outils de développement basés web rendues possible par les avancées de HTML 5 et de JavaScript. En proposant des fonctionnalités de clonage et en mélangeant document et application, l'apprentissage par mimétisme se retrouve facilité. Le développement de WikiNEXT a duré trois ans (Arapov et Buffa 2012 [120]), et a été écrit à partir de zéro. Les motivations viennent d'un constat d'expérience : nous avions déjà développé deux wikis sémantiques depuis 2006 : SweetWiki (Buffa et Al. 2008 [34], voir section 3.4) et DekiWiki (Buffa et Al. 2012 [117] voir section 4.1), et nous nous sommes heurtés aux limitations que partagent aussi les autres wikis sémantiques les plus populaires : ils ne permettent pas *facilement* d'écrire des applications exploitant les données sémantiques et le web de données. La raison principale est le choix d'une approche « par macros » ou « par extensions » au lieu de proposer une interface de programmation complète directement dans le navigateur.

Pour réaliser l'application présentée dans la Figure 43, avec les wikis actuels, on aurait inséré une macro dans un document, que l'on aurait pu paramétrer (pour indiquer la requête, le type de graphe, etc.), mais le code « derrière » une macro nécessite des outils

ad hoc lourds et un accès au serveur. Seuls des experts peuvent s'attaquer à ce type de développement. L'originalité de WikiNEXT est qu'il est à la fois un wiki sémantique, et un IDE[35] en ligne (Integrated Development Environment), *il considère les pages comme des applications* contenant du texte, des images ou autres médias, des métadonnées et aussi une partie programmable. Les avantages par rapport à l'approche basée « macros » sont nombreux :

- *Programmation instantanée* (« live coding ») : je code et je vois le résultat quasiment en temps réel de mes modifications, cette approche facilite les expérimentations. Elle est très adaptée à des applications pédagogiques,
- Partage des applications créées dans le wiki (clonage, partage),
- *Documentation « sur place » des applications* : une page du wiki peut contenir un tutorial ou la documentation de sa partie applicative,
- *Utilisation de langages standards (JavaScript, JSON, RDF, RDFa, SPARQL, etc*.). et non de langages spécialisés (DekiScript dans l'outil DekiWiki, macros de TWiki ou de l'extension HALO de Semantic Media Wiki, etc.)

WikiNEXT propose également des outils pour aider le développeur : il fournit une riche API accessible depuis le code JavaScript, des aides interactives (pour la création et la validation à même l'éditeur de requêtes SPARQL par exemple), un système de templates / modèles de pages pour l'affichage de données sémantiques, une base de données de graphes compatible SPARQL 1.1 pour le raisonnement et pour l'implémentation d'un SPARQL endpoint, une base de données objet pour la persistance, etc.

L'outil est open source, et propose de nombreux exemples et tutoriaux montrant comment collecter des données depuis DBPedia, depuis FreeBase, et/ou depuis la base de connaissance locale du wiki. Les démonstrations et tutoriaux sont variés : mashup de plusieurs sources de données, annotation automatique de documents, génération de vues personnalisées sous forme de page de wiki ou sous forme de visualisations graphiques (templates), utilisation d'un cache permettant de ne pas requêter le web de données à chaque affichage de page enrichie, avec gestion de la durée de vie des données, etc.

WikiNEXT s'adresse à plusieurs profils d'utilisateurs. Un utilisateur lambda peut utiliser l'éditeur WYSIWYG et créer des documents classiques, cependant nous avons mis dans ce chapitre l'accent sur le côté « wiki pour écrire des applications ». La description que nous donnerons de WikiNEXT concerne ici principalement les développeurs d'application web et web sémantique, mais aussi les enseignants ou ceux qui utilisent déjà des petits « web IDEs » pour tester du code.

[35] IDE = Integrated Development Environment. Eclipse ou Netbeans sont des IDEs classiques, jsbin.com, jsfiddle.net sont des IDEs en ligne, ou « web based IDEs ».

Convergence entre wikis classiques, wikis d'application, wikis sémantiques et « environnement de développement basés web »

Wikis d'application : Issus du Web 2.0, les wikis, dont le représentant le plus illustre est WikiPedia, sont connus pour favoriser la convergence de l'écriture collaborative de documents. Rapidement, certains moteurs de wikis ont étendu cette approche pour aller au-delà de l'écriture de simples documents textuels. Les « wikis d'application », aussi appelés « wikis d'entreprise », ont ajouté le moyen de « programmer dans les pages du wiki ». TWiki (encore populaire de nos jours), en 1998, a été un pionnier de cette approche en proposant d'utiliser des macros pour générer du contenu dynamique dans les pages. Par exemple on pouvait générer des tableaux, des formulaires, des champs des recherche dans les pages, les données manipulées étant elles aussi stockées dans des pages ayant une structure particulière (sous forme de tables ou de liste HTML). En 2005, d'autres wikis, comme JOT (devenu par la suite Google Sites, puis Google Docs), XWiki ou Mindtouch Core[36] sont allés plus loin en intégrant de véritables langages de script.

Ainsi des utilisateurs avancés ayant des notions de programmation pouvaient développer des « petites applications dans le wiki » incluant la définition de modèles de données, leur traitement et leur présentation. Tout en conservant la possibilité pour d'autres utilisateurs de copier/coller une page/application et de modifier légèrement le travail cloné pour en faire une version personnelle.

C'est ce que Ward Cunningham appelle *« the Wiki Way »* et qui transforme les wikis en puissants outils pour *« apprendre en regardant le travail des autres »* (Leuf et Cunningham 2001 [60]). Le code source de ces applications et les données ne sont pas visibles par les utilisateurs qui consultent les pages, mais éditer une page révèle les secrets de fabrication (le script, les macros) de l'application qu'elle renferme. Ces macros sont soit prédéfinies, soit écrites par les utilisateurs à l'aide d'un langage de script (Velocity[37] ou Groovy[38] pour XWiki, DekiScript, pour le wiki Mindtouch Core, etc.) Les ensembles de macros peuvent utiliser des plugins ou des extensions ajoutés côté serveur (par exemple un plugin pour se connecter à une base de données SQL externe, fournira une macro utilisable dans les pages). *Dans ce cas, le développeur du plugin ou de l'extension doit avoir accès au code source du Wiki, ou à un SDK fourni avec le moteur de wiki. Parfois le nombre d'outils nécessaires pour pouvoir développer un plugin est très important.*

Les Wikis sémantiques : déjà présentés dans la section 3.3, ces wikis permettent à leurs utilisateurs d'ajouter du contexte sémantique aux documents tout en préservant la simplicité et l'essence collaborative des wikis. Les ressources sémantiques sont en général conservées en interne (dans une base de connaissance) sous la forme de données RDF, OWL ou sous forme de graphes conceptuels [121]. Les données sémantiques produites sont déjà intéressantes en elles-mêmes (par exemple dans le cas de production

[36] http://www.mindtouch.com/,utilise DekisScript, un langage inspiré par JavaScript, pour écrire du code dans les pages.
[37] http://velocity.apache.org/
[38] http://groovy.codehaus.org/

collaborative d'ontologie, ou d'annotation de documents), dans la mesure où elles sont naturellement difficiles à produire, mais surtout, les wikis équipés de moteurs sémantiques peuvent les exploiter pour du raisonnement, implémenter des outils de recherche augmentée, des systèmes de suggestion, etc. Les wikis sémantiques proposent souvent des langages intermédiaires proches du langage naturel (on écrit dans une page du wiki « *this is a class* » dans Semantic Media Wiki pour créer une classe qui sera un concept dans une ontologie) ou à base de langage de markup popularisé par la « culture wiki ». Semantic Media Wiki, ou SemperWiki [122] utilisent une syntaxe à base de crochets pour intégrer des annotations, qui seront ensuite traduites en RDF une fois la page sauvegardée. Cette approche, très simple, basée sur les usages des wikis de l'époque, a posé quelques problèmes de cohérence dans les premières générations de wikis sémantiques (2005-2010). IkeWIki [123] a combiné l'approche classique des wikis pour la création de documents avec une interface à base de formulaires avec auto-complétion pour faciliter la réutilisation de vocabulaires et la vérification de cohérence. OntoWiki [124] quant à lui est un véritable éditeur d'ontologie avec une interface utilisateur proposant différentes vues pour naviguer et organiser l'ontologie produite. SweetWiki le premier wiki sémantique que j'ai développé et co-conçu en 2005 proposait un système de tagging social sémantique et conceptualisait le wiki lui-même sous la forme d'une « ontologie du wiki », implémentant ainsi la quasi-totalité de ses fonctionnalités autours des technologies du web sémantique.

En 2013, peu de wikis sémantiques sont encore maintenus[39] on pourra retrouver un état de l'art récent dans (Meilendera et al. [125]) que nous résumerons ici : il existe encore des wikis de l'ancienne génération qui ont (1) étendu leurs fonctionnalités en devenant des hybrides de CMS (Content Management Systems) ou (2) au contraire en se « coupant en morceaux » et en externalisant certaines fonctionnalités à des entités externes dédiées. Dans la première catégorie nous retrouvons Semantic Media Wiki avec aujourd'hui de très nombreuses extensions, comme Halo, qui propose des formulaires d'annotation avec auto-complétion, un éditeur WYSIWYG, l'intégration de fichiers multimédia et la mise en place d'un « SPARQL endpoint ». On trouve aussi des wikis basés sur Semantic Media Wiki, comme Moki (Rospocher et al 2009 [126]), spécialisé sur la modélisation de processus d'entreprises, ou OWiki (Di Lorio et al. 2010 [127]), dédié à la génération de contenu dirigé par des ontologies. Dans la seconde catégorie on trouve le projet KiWi - Knowledge In Wiki- (Shaffert et al. 2009 [114]), qui implique les auteurs de IkeWiki, refondu dans un projet comprenant de nombreux modules, ou SweetDeki (Buffa et Husson 2012 [117]), le wiki qui a succédé à SweetWiki, fondu dans le projet ANR ISICIL (Gandon et al. 2009 [105]) intégrant un réseau social, un serveur de tags, un gestionnaire de ressources sémantiques externes, etc. Ces projets ont essayé de mettre le wiki sémantique dans un contexte industriel (ISICIL a donné naissance à la société Mnemotix.com, KiWi est appuyé par des fabricants de CMSs). Bien que les wikis sémantiques cités permettent la création d'ontologie ou l'annotation de documents,

[39] Il y en avait 37 de recensés lors du workshop qui leur était dédié lors de la conférence ESCW 2006, moins d'une dizaine sont encore actifs en 2013.

l'exploitation des données sémantiques par l'utilisateur se limite souvent à l'insertion de requêtes ou macros dans les pages, faisant références à des extensions ou plugins éventuellement installés. *Dans ce sens, ces wikis sémantiques modernes ont convergé vers les wikis d'applications « historiques » présentés dans la page précédente, et partagent également leurs défauts : développer un nouveau plugin ou une extension requiert un SDK ou l'accès au code source du wiki. On peut développer de petites applications « dans le wiki », mais avec des contraintes importantes.*

Outils de développement basés web (*« Web-based IDEs »*) : L'idée consistant à écrire des applications directement dans un navigateur web n'est pas récente : les premières expérimentations datent de 1996 (Crespo et al. 1996 [128]). Depuis 2005, avec la technologie Ajax et le développement de JavaScript, les éditeurs de code source pouvant fonctionner dans une page web n'ont cessé de s'améliorer. Des éditeurs comme Code Mirror[40], ACE Cloud9[41] ou ternjs[42] proposent la colorisation de syntaxe, l'auto-complétion, des fonctions d'édition complètes, et équipent plusieurs sites web connus proposant des environnements de développement basés web[43]. Certains supportent plusieurs langages de programmation comme compile-online.com, ideone.com ou compilr.com, mais les plus intéressants sont sans doute les environnements dédiés à la programmation JavaScript/HTML/CSS, très appréciés des *développeurs web*.

Des sites comme jsbin.com, jsfiddle.net, codepen.io ou tinkerbin.com proposent d'écrire du code directement dans le navigateur et de voir l'exécution en temps réel, pendant la saisie. On parle de « live coding », ou « codage vivant », qui est particulièrement appréciable pour évaluer rapidement des algorithmes, des idées, ou tout simplement pour tester des fonctions d'APIs de HTML5. Ces outils proposent souvent une option de clonage d'une application existante. La plupart de ces outils, cependant, ne sont que des bacs à sable pour tester du code ou pour écrire des exemples à but pédagogique. Enfin, on trouve des outils de développement comme cloud9ide.com, shiftedit.net, coderun.com ou codeanywhere.net qui ambitionnent de remplacer des outils comme Eclipse, en proposant de gérer en ligne le développement de véritables projets composés de multiples fichiers JavaScript/HTML/CSS.

WikiNEXT reprends certaines fonctionnalités de la dernière génération des outils présentés dans cette section.

A notre connaissance, en 2013, aucun wiki sémantique n'a encore adopté une telle approche pour « programmer dans le wiki ».

4.2.2 Les principes fondamentaux de WikiNEXT

Motivation : wikis d'application, wikis sémantiques, IDEs basés web : WikiNEXT est un hybride, un mélange de ces trois approches. Tout d'abord, dans WikiNEXT, *chaque page est une application web versionnée*, composée de code JavaScript/HTML/CSS.

[40] http://codemirror.net/
[41] http://ace.c9.io/
[42] http://ternjs.net/
[43] See http://en.wikipedia.org/wiki/Comparison_of_JavaScript-based_source_code_editors

Nous l'avons vu dans l'introduction, plusieurs éditeurs sont fournis : un éditeur WYSIWYG pour des documents classiques, et un éditeur de code pour la partie « application ». Par ailleurs WikiNEXT expose une API donnant aux applications accès aux mécanismes internes du wiki, et facilitant la manipulation de données sémantiques (requêtes, affichage, persistance).

Considérons par exemple une page classique créée par un utilisateur ne sachant pas programmer. Il pourra demander à une personne ayant des compétences en développement web « d'augmenter sa page » en écrivant un bout de code qui collecte des informations d'une source de données externe comme DBPedia afin d'annoter le document qu'il a écrit, ou pour extraire des éléments du texte afin de suggérer une classification, insérer une visualisation graphique, ou sauvegarder les données collectées dans la base de connaissances du wiki.

Il sera possible par la suite de cloner la page/application existante pour l'étudier et en faire une version différente. Comme dans tout wiki, il est nécessaire *d'amorcer le wiki* en proposant une base de données initiale d'exemples variés, de tutoriaux, afin de faciliter l'apprentissage par clonage/modification.

WikiNEXT propose des services pour l'annotation de textes, la persistance « back end », côté serveur, en fournissant notamment une base de données de graphe compatible RDF/SPARQL 1.1, et une base de données objet, WikiNEXT est aussi un SPARQL endpoint.

Nous avons mis l'accent sur cet aspect « programmation » du wiki, car les applications sémantiques sont souvent associées avec des « documents » qui sont soit considérés comme des sources de données, soit comme des cibles pour des annotations ou des enrichissements.

Par ailleurs, WikiNEXT a aussi une vocation pédagogique, en tant qu'outil permettant de développer des tutoriaux interactifs. On peut, par exemple, écrire un document avec du texte explicatif, mélangé avec l'application qu'il décrit : un tutoriel sur « comment requêter DBPedia » contiendra *à la fois le texte et le code de l'application*, présentera les résultats d'exécution en temps réel et proposera de modifier directement le code pour voir le résultat des modifications.

La couche « application » d'un document est constituée de code JavaScript qui peut être édité de la même manière que l'on peut éditer le code HTML de la page. Ainsi il est possible de manipuler depuis le code JavaScript le Document Object Model (DOM) de la page comme on le fait dans la programmation JavaScript classique : rajouter un `id` sur un `<div>` HTML pour afficher des résultats, ou bien utiliser le système de templates de présentation de WikiNEXT, présenté dans la section suivante.

Certaines tâches comme « créer une nouvelle page du wiki », « requêter la base de connaissance du wiki », « annoter un document et maintenir les vues sur ces annotations à jour lorsque les valeurs changent » (liste non exhaustive) sont réalisées par des fonctions de l'API interne du wiki, alors que d'autres peuvent s'appuyer sur des librairies JavaScript

externes telles que D3.js[44], une puissante librairie graphique pour la visualisation de données.

WikiNEXT propose par ailleurs un outil intégré permettant d'uploader des librairies externes et de sélectionner celles utilisées par chaque page/application. Le prototype est open source[45]. Il inclut aujourd'hui de nombreux exemples et tutoriaux. Nous l'utilisons avec des étudiants de Master pour illustrer certaines utilisations du web de données et pour apprendre à utiliser les langages RDF/SPARQL.

Architecture du logiciel : dans le paradigme de développement agile, des chercheurs ont montré que les wikis peuvent être utilisés comme plate-forme pour l'échange d'objets logiciels légers réutilisables (Rech, Bogner et al. 2007 [129]), cependant il s'agissait de « bouts de logiciels » externes au wiki.

WikiNEXT reprend ces principes et les mets en œuvre dans le paradigme MVC :

- La partie JavaScript de la page formant « la couche métier et le contrôleur »,
- La partie « vue » se faisant dans le code HTML / CSS de la page,
- La partie « modèle » étant composée de métadonnées RDF.

La synchronisation de la vue et des modèles est implémentée par WikiNEXT. Il est également possible de synchroniser les modèles avec les bases de données de WikiNEXT, situées côté serveur, via l'API WikiNEXT.

Chaque page WikiNEXT est associée avec un ensemble de métadonnées qui décrivent ses principales caractéristiques : titre, auteur, contributeurs, dernière date de modification, version, etc. Mais les pages sont aussi des « containers » : elles contiennent des métadonnées qui auront pu être rajoutées manuellement ou par une application d'annotation.

WikiNEXT s'appuie sur les ontologies de schema.org[46] pour décrire la structure des pages, les utilisateurs, etc. Chaque page est l'instance d'un `Article`, elle est représentée par un graphe nommé constitué d'annotations RDFa/RDFAlite[47], dont le nom est basé sur l'URI de la page (Figure 45).

- Nous appelons l'ensemble des métadonnées localisées dans une page la *base de connaissances locale*.
- L'ensemble des pages du wiki, est appelé *base de connaissances globale*, elle est constituée de l'union de tous les graphes nommés.

Les métadonnées générées par la partie programmable d'une page, par exemple une liste de pays provenant d'une requête sur DBPEdia.org, pourront être injectées dans la page

[44] http://d3js.org/
[45] https://github.com/pavel-arapov/wikinext
[46] http://schema.org/and http://schema-rdfs.org/ont été créés par les principaux moteurs de recherche (Google, Microsoft, Yahoo et Yandex). Ils proposent des vocabulaires partageables RDF/S et pour les microdatas de HTML5, qui couvrent les principaux domaines.
[47] http://www.w3.org/TR/xhtml-rdfa-primer/ et http://www.w3.org/TR/rdfa-lite/

sous forme d'annotations RDFa/RDFalite. Elles seront également sauvegardées dans un triple store intégré à la partie serveur du Wiki lors de la sauvegarde de la page.

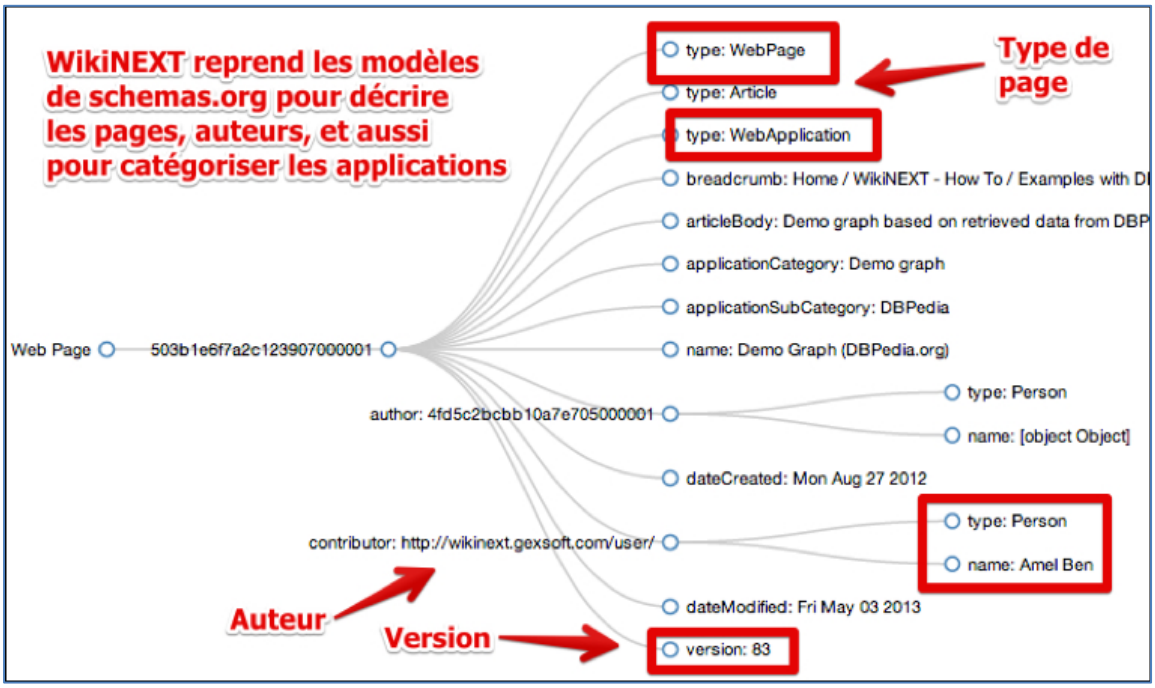


Figure 45. Le modèle des pages de WikiNEXT

Implémentation (Figure 46) : WikiNEXT a été écrit de zéro en JavaScript. Il s'appuie sur le micro-serveur Node.js[48] et sur la base de données MongoDB[49]. Node.js embarque l'interpréteur JavaScript V8 de Google, qui équipe entre autres le navigateur Google Chrome. Ainsi, Node.js permet ce que l'on appelle le « server-side JavaScript », une tendance émergente dans le développement web. MongoDB est une base de données NoSQL orientée documents (indexation du contenu, etc.). Au lieu de stocker des objets sérialisés sous forme de lignes et colonnes dans des tables comme on le ferait avec une base de données relationnelle traditionnelle, ici on stocke des objets JavaScript au format BSON (une forme binaire du format de sérialisation JSON[50] de JavaScript). Voir Figure 46 et Figure 47.

[48] http://nodejs.org/
[49] http://mongodb.org/
[50] http://www.json.org/ une notation textuelle pour décrire des objets JavaScript.

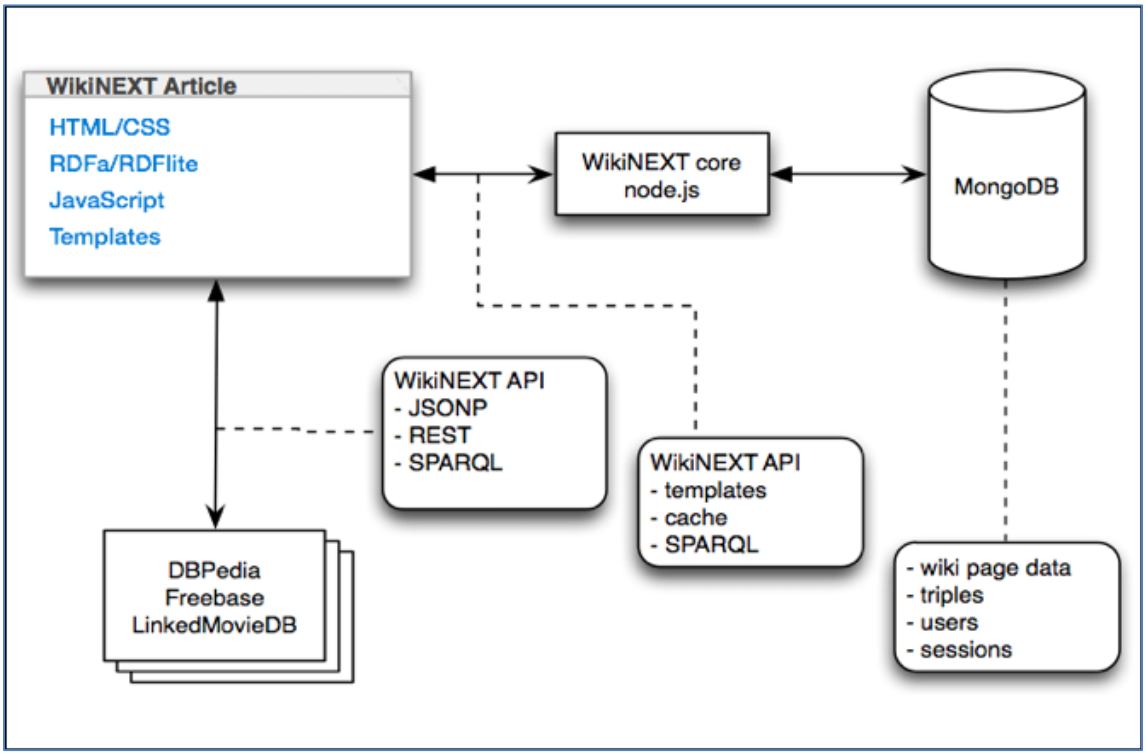


Figure 46. Architecture technologique de WikiNEXT

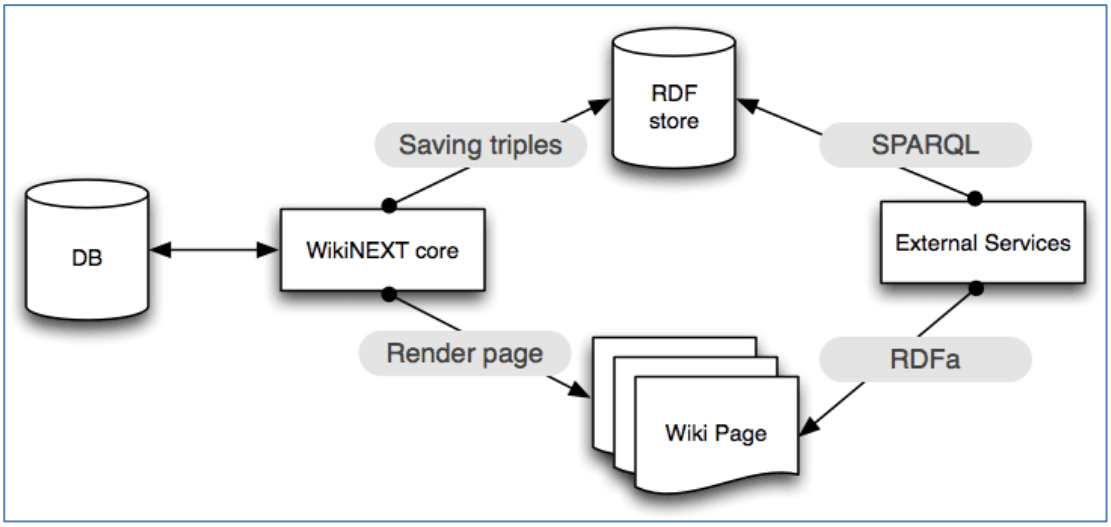


Figure 47. Cycle de vie des métadonnées sémantiques

Nous utilisons pour gérer les triplets RDF une version améliorée de RDFStore-js (Hernandez et Garcia 2012 [130]), un moteur de graphes compatible SPARQL 1.1, écrit en JavaScript. Le contenu traditionnel de la page est stocké sous forme d'objets dans la base de données MongoDB (notamment, nous générons des indexes pour le contenu textuel de la page). Nous utilisons également MongoDB comme couche de persistance pour RDFStore-js. Ainsi, les triplets RDF sont accessibles à la fois à travers des requêtes SPARQL, mais aussi directement via MongoDB. On dispose d'un accès direct aux nœuds du graphe RDF, ce qui est utile pour implémenter de manière très efficace certaines tâches (recherche d'entités nommées, notamment).

Pour l'affichage et la visualisation d'annotations sémantiques, WikiNEXT propose un système de templates. Le cas d'utilisation classique consiste à créer des pages WikiNEXT faisant office de modèles de présentation (templates) qui seront réutilisées par d'autres pages, par leur partie applicative. Le principe est simple et permet de découpler l'interface utilisateur de la définition des données.

Les templates, la manipulation de requêtes SPARQL, la manipulation d'annotations RDF, la persistance, le cache des données, toutes ces fonctionnalités sont accessibles par le code des applications que l'on écrit dans le wiki, au travers de l'API WikiNEXT. Avec ces choix technologiques, nous avons essayé de minimiser le nombre de langages de programmation utilisés et le nombre de formats de données.

Avec WikiNEXT, JavaScript est utilisé le long de la chaîne complète, depuis le code fonctionnant dans le navigateur web (code client) jusqu'au code serveur et au format de persistance.

4.2.3 Mashup sémantique et utilisation de la base de connaissance globale de WikiNEXT

NOTE : l'exemple décrit dans cette section fait l'objet d'une vidéo en ligne[51]. Nous encourageons les lecteurs à la visionner !

WikiNEXT permet de développer des applications plus complètes et plus variées que l'exemple très simple présenté dans l'introduction. Nous décrivons ici un scénario d'usage complet : le développement d'une application qui utilise le SPARQL endpoint de DBPedia.org pour récupérer des données RDF concernant des villes (description, population, photos, etc.).

Les résultats sont utilisés pour créer à la volée des pages du wiki –une par ville-, basée sur un modèle de présentation, lui aussi créé dans le wiki. Les pages sont annotées et les annotations sont sauvegardées dans le triple store RDF du wiki. Le nom des villes à chercher se fait à l'aide d'un formulaire, lui aussi créé dans le wiki ; enfin, une dernière page réalise un « mashup sémantique » en construisant une carte présentant l'ensemble des villes qui ont été récupérées, avec un résumé et des photos de chacune d'entre elles.

Cette application était disponible en ligne sur le site de WikiNEXT[52], c'était un des tutoriaux que l'on pouvait consulter, éditer et dont on pouvait voir le code, le modifier ou le cloner[53].

Ce tutorial guide pas à pas le développeur : on commence par créer en HTML un formulaire de saisie (Figure 48) permettant de choisir la ville à rechercher, le pays, la langue. Les paramètres saisis servent à requêter le SPARQL endpoint de DBPEdia.org.

Les résultats sont ensuite utilisés pour créer à la volée une page dans le wiki. Pour cette application on ne va pas conserver le modèle original de DBPedia.org pour les villes mais

[51] https://www.youtube.com/watch?v=0mTbdVC115s
[52] https://web.archive.org/web/20140329131634/http://wikinext.gexsoft.com:80/home
[53] Il suffit de se connecter au Wiki (inscription traditionnelle ou via identifiants Facebook) et d'éditer la page de l'application « City ».

on va décrire les villes en utilisant des propriétés du modèle http://schema.org/city, plus simple (Figure 50). Lorsqu'une page du wiki est sauvegardée (c'est le cas de la page créée à la volée pour chaque ville-résultat, voir la Figure 49), les métadonnées sont également ajoutées à la base de connaissance de WikiNEXT (au format RDF). L'intérêt de cet exemple est de montrer l'utilisation des fonctions d'API pour manipuler les annotations sémantiques et pour illustrer l'utilisation de la base de connaissance RDF/SPARQL de WikiNEXT.

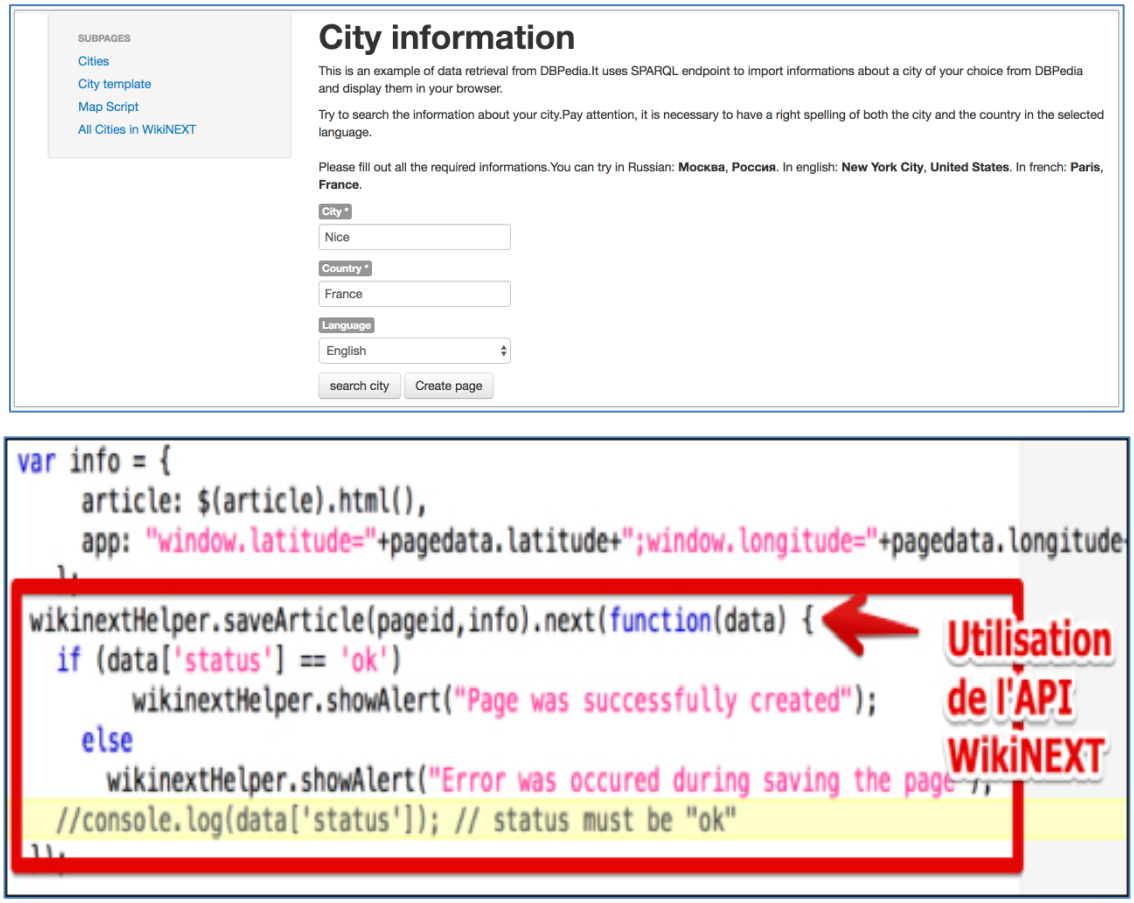


Figure 48. Formulaire de saisie de l'application « City », un des tutoriaux de WikiNEXT, et exemple de code utilisant l'API de WikiNEXT pour créer une page à la volée avec les résultats d'une requête SPARQL.

L'API fournit d'autres fonctionnalités comme la possibilité de cacher/mémoriser des données avec une certaine durée de vie. Par exemple, la page de la ville de Paris contient des annotations concernant la population, mais on peut indiquer que ces données ont une certaine durée de vie. Passé un délai, l'affichage de la page déclenchera un rafraichissement des données depuis DBPedia.org (au lieu de les prendre dans le cache local, c'est-à-dire dans la base de connaissance locale, voir un des exemples en ligne disponibles[54]).

[54] Exemple d'application wikiNEXT utilisant un cache avec durée de vie.

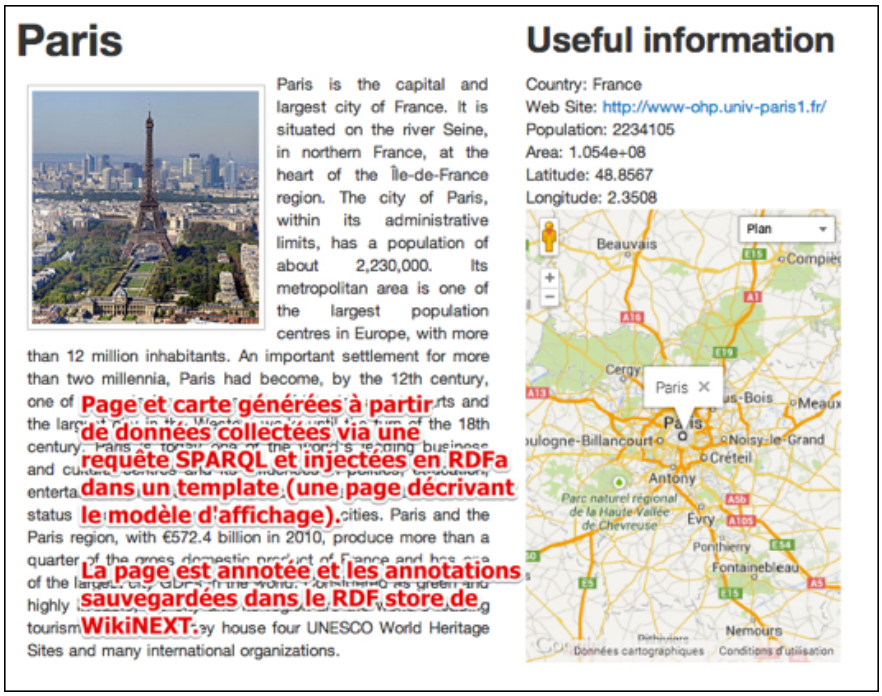


Figure 49. Exemple de page générée

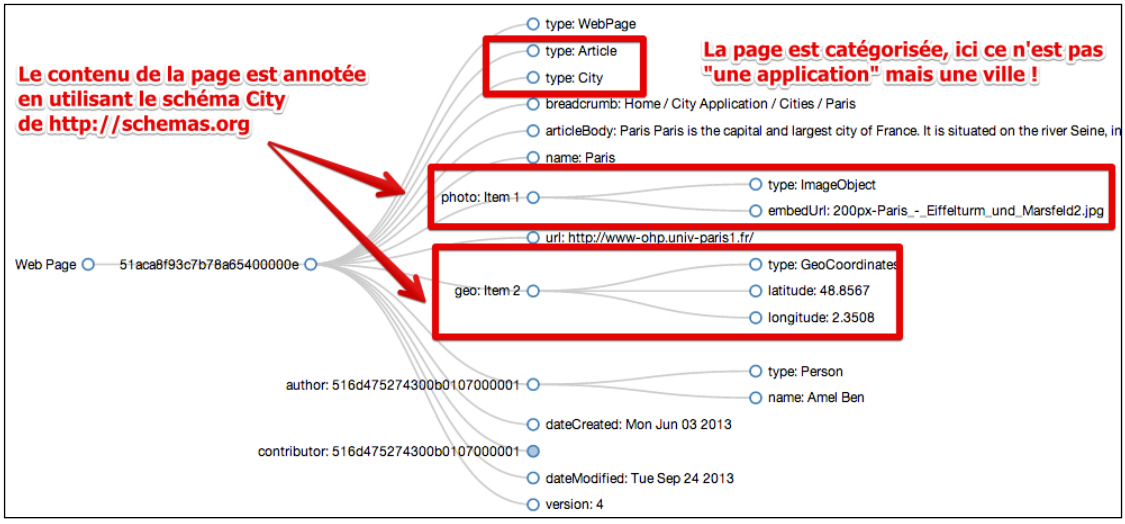
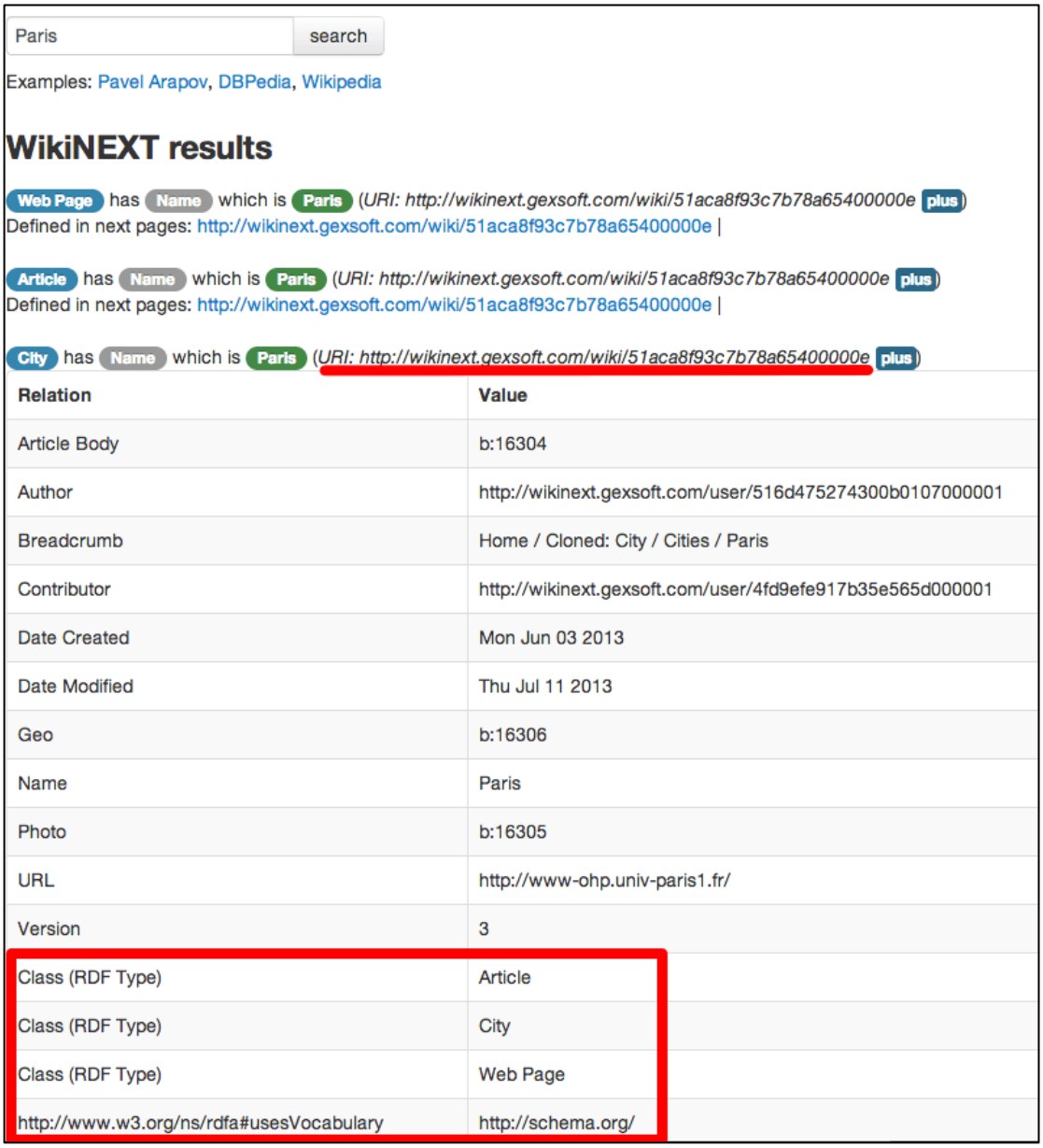


Figure 50. Annotations d'une page créée dynamiquement, pour la ville de Paris.

La Figure 51 montre un exemple de recherche sur le mot clé « Paris » et on trouve bien des résultats de type « City ». Noter que d'autres résultats sont retournés, avec des types différents (une « page » a pour nom « Paris » etc.)


Paris search

Examples: Pavel Arapov, DBPedia, Wikipedia

WikiNEXT results

Web Page has Name which is Paris (URI: http://wikinext.gexsoft.com/wiki/51aca8f93c7b78a65400000e plus)
Defined in next pages: http://wikinext.gexsoft.com/wiki/51aca8f93c7b78a65400000e |

Article has Name which is Paris (URI: http://wikinext.gexsoft.com/wiki/51aca8f93c7b78a65400000e plus)
Defined in next pages: http://wikinext.gexsoft.com/wiki/51aca8f93c7b78a65400000e |

City has Name which is Paris (URI: http://wikinext.gexsoft.com/wiki/51aca8f93c7b78a65400000e plus)

Relation	Value
Article Body	b:16304
Author	http://wikinext.gexsoft.com/user/516d475274300b0107000001
Breadcrumb	Home / Cloned: City / Cities / Paris
Contributor	http://wikinext.gexsoft.com/user/4fd9efe917b35e565d000001
Date Created	Mon Jun 03 2013
Date Modified	Thu Jul 11 2013
Geo	b:16306
Name	Paris
Photo	b:16305
URL	http://www-ohp.univ-paris1.fr/
Version	3
Class (RDF Type)	Article
Class (RDF Type)	City
Class (RDF Type)	Web Page
http://www.w3.org/ns/rdfa#usesVocabulary	http://schema.org/

Figure 51. Exemple de résultats de recherche sur le mot-clé « Paris ». On voit bien le typage des URLs retournés *« city has name which is Paris »*….

Réutilisation des données : pour illustrer l'utilisation de la base de connaissance globale de WikiNEXT, nous pouvons également réaliser un mashup à partir des métadonnées associées aux pages des villes construites à partir de requêtes envoyées depuis le formulaire présenté précédemment.

Rappelons que les métadonnées correspondant à chaque ville sont à la fois présentes dans les pages mais également dans le triple store de WikiNEXT. Pour l'application de mashup (une carte avec l'ensemble des villes et une liste contenant une vue résumée de chaque ville), nous créons une nouvelle application/page dans le wiki mais cette fois-ci nous requêtons en SPARQL la base de connaissance du wiki au lieu de requêter le web de données, les résultats sont présentés Figure 52. WikiNEXT pré-charge les ontologies de schema.org, les données sur les villes stockées dans WikiNEXT ont été, dans cet exemple,

écrites en se référant au schéma http://schemas.org/City. La Figure 53 montre le workflow de l'application. Elle illustre la capacité de WikiNEXT à gérer différentes sources de données dans ses applications pour créer des mashups sémantiques.

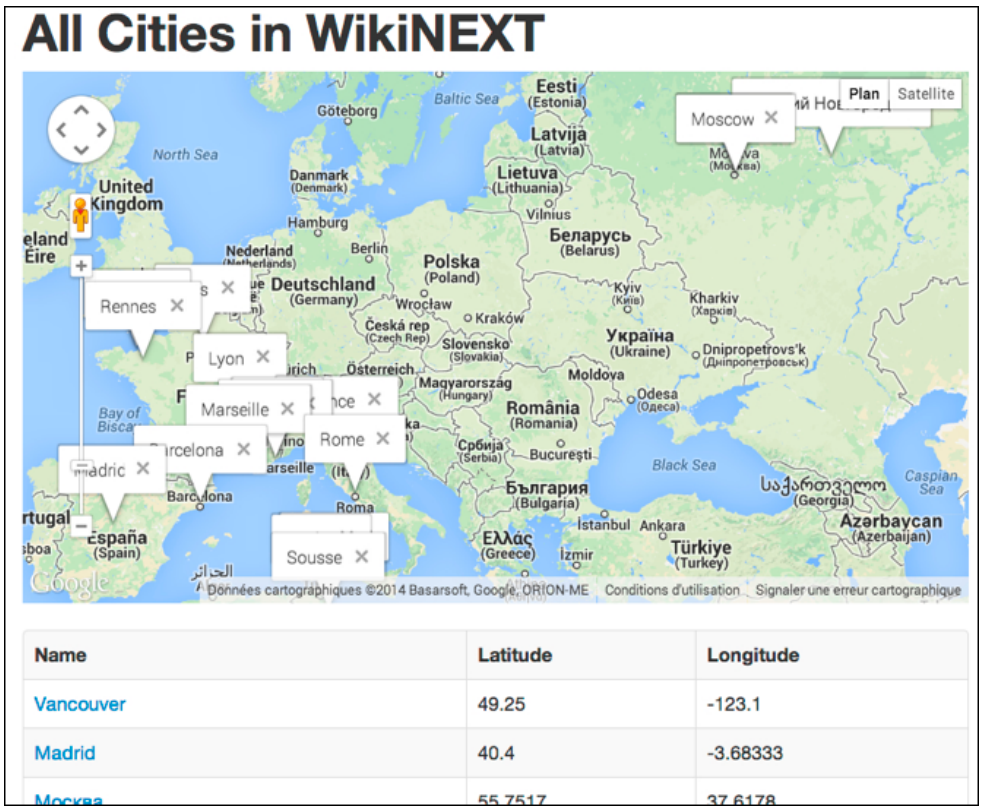


Figure 52. Mashup sémantique à partir de la base de connaissances de WikiNEXT.

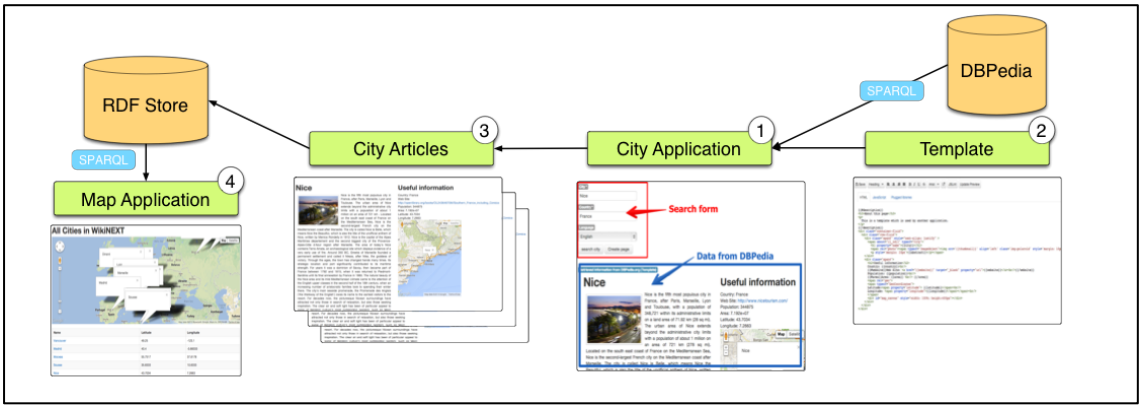


Figure 53. Workflow de l'application « Cities » codée dans des pages de WikiNEXT.

4.2.4 Usages et évaluation de WikiNEXT

Aujourd'hui plus maintenu, WikiNEXT a été développé principalement par une personne dans le cadre de sa thèse, mais la nature ouverte du wiki fait qu'il a été testé par un groupe de 5 à 6 personnes au cours de son développement. La plupart des exemples en ligne ont été développés par ce noyau de béta testeurs. L'outil est stable et nous avons procédé à des évaluations auprès de personnes n'ayant pas participé à son développement [118]. Nous avons défini un protocole de test consistant à effectuer divers tutoriaux, notamment

l'application décrite dans la section 4.2.3. Le groupe de test était constitué de développeurs web et d'étudiants en master ou en thèse d'informatique. Nous avons mesuré le temps passé pour chaque étape, observé les comportements et noté tous les problèmes rencontrés. Finalement nous avons conduit des interviews avec chaque personne et demandé de comparer cette expérience avec un développement classique pour effectuer les mêmes tâches. Nos web développeurs ont suivi un cours sur le web sémantique et ont une petite expérience de développement dans ce domaine. Les résultats[55] ont montré que WikiNEXT est jugé simple à utiliser et propose une manière originale et rapide pour requêter le web de données et réutiliser les données sémantiques. La principale difficulté rencontrée par les utilisateurs a été l'écriture de requêtes SPARQL : la syntaxe en elle-même a parfois posé problème, notamment la nécessité d'encadrer certains caractères avec des anti slash en JavaScript, mais aussi l'apprentissage de vocabulaires nouveaux. Nous avons par la suite écrit des outils d'aide au requêtage (traitement automatique des caractères spéciaux pour JavaScript, widget avec des requêtes prêtes à l'emploi, historique des requêtes récentes, auto-complétion sur les propriétés des vocabulaires, etc.). Ces applications[56] d'aide ont toutes été écrites dans WikiNEXT, ce qui montre son potentiel.

Des expérimentations à plus grande échelle, ont suivi auprès d'étudiants qui suivent le cours Web Sémantique en master 2 informatiques à l'Université de Nice, et avec les étudiants internationaux qui suivent le cours HTML5 du W3C[57]. Notamment, une série de tutoriaux montrant comment exploiter en SPARQL DBPedia.fr a été développée au cours du stage d'une étudiante de Master 2 et intégré au site web DBPedia.fr et accessible en ligne[58].

4.2.5 Bilan des nouveautés apportées par WikiNEXT

WikiNEXT propose une approche originale en mélangeant wiki sémantique et environnement de développement basé web (« web-based IDE »). S'appuyant sur des technologies émergentes, il a proposé un environnement intégré pour programmer le web de données. Ses applications sont principalement pédagogiques (pour enseigner le web de données ou HTML5) mais aussi, il concurrence directement les outils de développement basés web comme jsbin.com ou jsfiddle.net, en ajoutant un moteur de templates, une base de connaissances, une puissante API pour manipuler des données sémantiques et pour requêter le web de données. Des tests d'usage ont été menés auprès de développeurs web et d'étudiants de master informatique, avec succès. En ajoutant l'aspect « *programmation d'application dans la navigateur* » il étend le concept de wiki sémantique en ouvrant de nouveaux champs d'application.

Pour conclure, on peut dire que WikiNEXT a anticipé la mode du *« live coding d'applications sémantiques* ». Il a été précurseur de la montée en puissance de JavaScript,

[55] Le protocole de test, les résultats, le questionnaire, sont disponibles.
[56] Voir par exemple l'outil d'aide au requêtage SPARQL intégré dans WikiNext.
[57] Le cours HTML5 du W3C est disponible sur http://w3devcampus.com. WikiNEXT a été utilisé pour écrire certains exemples interactifs et tutoriaux de ce cours.
[58] http://fr.dbpedia.org/sparqlTuto/tutoSparql.html

de la programmation centré « côté client », en exploitant des serveurs RESTful. Par ailleurs, l'approche « orientée API » de WikiNEXT :

- Chaque page est une application web,
- Chaque page est aussi un web service qui expose les métadonnées de la page (*base de connaissance locale*)
- Chaque page (application) exploite l'API du wiki pour la persistance, comme le ferait n'importe quel client externe,
- L'ensemble des métadonnées de toutes les pages compose une « *base de connaissance globale* » exploitable par toutes les pages / applications du wiki,
- L'utilisation d'un moteur SPARQL/RDF en JavaScript, a été une preuve de concept intéressante, à une époque où NodeJS était encore une technologie émergente,
- WikiNEXT a été pensé pour exploiter le web de données (capture des données, cache, aide au requêtage, etc.)
- WikiNEXT est idéal pour concevoir des tutoriaux interactifs, approche qui sera popularisée plus tard avec les WorkBooks Python. Le tutorial officiel de DBPedia.fr a contenu de 2013 à 2016 des exemples hébergés dans WikiNEXT.

4.3 Gestion des droits d'accès via l'ontologie AMO

Remarque importante : le travail transversal présenté dans cette section a été mené au début du projet ANR ISICIL, et visait à apporter une solution de gestion des droits d'accès aux ressources partagées par tous les modules logiciels impliqués dans le projet : wikis, extensions pour navigateurs, partages de favoris, outils de web-scraping, moteur de recherche, gestionnaire de tagging social, etc.

Ce travail est illustré dans cette section par une implémentation dans le logiciel SweetWiki dont nous disposions au début du projet ISICIL. Par la suite, SweetWiki a été abandonné au profit de SweetDeki et les parties « social tagging », « réseau social » et « gestion des droits d'accès » ont été déportées dans un serveur sémantique centralisé.

En revanche, les exemples de requêtes SPARQL, les annotations RDFa ajoutées dans le HTML des pages, etc. sont demeurés les mêmes. L'ontologie AMO présentée est également restée inchangée et a finalement été au cœur de la gestion sémantique des droits d'accès du projet ANR ISICIL.

4.3.1 Introduction

La sécurité, la protection, le contrôle de l'accès représentent un enjeu majeur des systèmes de gestion de contenu. Cette problématique est centrale dans les sites web collaboratifs et dans les réseaux sociaux du web 2.0 où le principe même de l'édition collaborative de documents et la notion de partage posent la question de la définition des droits d'accès. De fait, le contrôle de l'accès aux ressources est un des défis lancés au web sémantique.

Dans cette section nous nous intéressons aux systèmes de gestion de contenu qui reposent sur des serveurs web sémantiques et nous proposons une approche de la gestion des droits

d'accès aux ressources fondée sur les modèles et techniques du web sémantique. Plus précisément, nous présentons une ontologie dédiée à la représentation des droits d'accès donnés sur un document à certains utilisateurs ou classes d'utilisateurs. Nous avons baptisé cette ontologie AMO, acronyme signifiant **A**ccess **M**anagement **O**ntology. AMO consiste d'une part en un ensemble de classes et propriétés permettant d'annoter les ressources dont il s'agit de gérer l'accès et en une base de règles d'inférence modélisant la stratégie (*policy* en anglais) de contrôle des droits à mettre en œuvre. Appliquées sur la base d'annotations des ressources, ces règles permettent de contrôler l'accès à ces ressources selon une stratégie donnée.

Cette modélisation déclarative sous forme d'une base de règles assure une adaptation facile de l'ontologie à différentes stratégies de contrôle des droits d'accès qui dispense ainsi de modifier les annotations des documents dans le cas d'un changement de stratégie.

Dans le cadre du projet ANR ISICIL, nous utilisons l'ontologie AMO pour gérer l'accès aux ressources partagées par un réseau de veilleurs d'entreprise : documents produits par des outils de gestion de contenu, wikis ou blogs, documents HTML statiques, dont certains issus d'opérations de web-scraping, bookmarks, etc.

Un des enjeux de ce projet orienté autour des techniques du web 2.0 et du web sémantique concerne la gestion de l'accès aux ressources partagées par le réseau social des veilleurs d'entreprise. Parmi les documents produits par les veilleurs figurent ceux d'un site collaboratif géré par le wiki sémantique SweetWiki que nous avons développé et qui nous sert à illustrer dans cet article l'utilisation d'AMO. Ce wiki intègre les technologies du web sémantique pour améliorer la structure, la recherche et la navigation. Plus précisément, il associe aux pages wiki des annotations RDF/S qui rendent le contenu de ces pages exploitable par le moteur sémantique CORESE (Corby et al. [7]).

La section suivante est dédiée au positionnement de notre approche par rapport aux travaux existants sur la gestion de l'accès aux ressources dans les systèmes de gestion de contenu et par rapport aux modèles sémantiques du web 2.0. Nous présentons l'ontologie AMO dans la partie 4.3.3

Nous montrons dans la partie la section 4.3.4 l'utilisation qui est faite d'AMO dans SweetWiki et ce faisant nous mettons en lumière la facilité d'adaptation d'AMO à différentes stratégies de contrôle.

4.3.2 Positionnement

4.3.2.1 Les langages XML pour le contrôle d'accès et les droits numériques

La plupart des mécanismes de contrôle d'accès implémentés dans les systèmes de gestion de contenus sont basés sur des langages XML permettant de décrire des politiques de contrôle d'accès et de gestion des droits numériques (ou DRM, acronyme pour Digital Rights Management). Ces systèmes exploitent des métadonnées associées aux ressources dont il s'agit de contrôler l'accès et ces métadonnées respectent les schémas XML

correspondant à ces langages dédiés. Parmi ceux-ci, les plus connus sont XrML[59] (eXtensible Right Markup Language) utilisé comme base du langage d'expression des droits du standard MPEG-21[60], ODRL[61] (Open Digital Right Language) implémenté par l'Open Mobile Alliance (OMA) et XACML[62] (Extensible Access Control Markup Language) développé par OASIS. Le modèle ODRL repose sur les notions d'*Asset*, *Party*, *Permission*, *Constraint*, *Requirement*, *Condition*, *Rights holder*, *Context*, *Offer*, *Agreement*, et *Revoking rights*. Le modèle XACML permet de représenter des stratégies de contrôle d'accès sous forme de règles et repose sur les notions de *Rule*, *Policy* et *Policy Set*, ces notions pouvant être précisées avec celles de *Subject*, *Resource*, *Action*, *Environment*. Une *Rule* comprend des *Conditions* et *Effects*, et une *Policy* des *Rules* et *Obligations*.

4.3.2.2 Les approches sémantiques du contrôle d'accès

Avec l'émergence du web 2.0 et du web sémantique, de nouveaux systèmes de gestion de contenu ont vu le jour qui reposent sur ces nouvelles approches du web pour gérer l'accès au contenu. Notamment, Alam et al. [159] montrent bien les limites des solutions de gestion des droits d'accès utilisant des langages de description non sémantiques. Ils proposent une ontologie OWL pour décrire les accès à des services web, inspirée du modèle XACML. Plus généralement, nous retrouvons dans les quelques modèles sémantiques de gestion de l'accès au contenu qui existent certaines notions déjà présentes dans les langages XML plus anciens.

Citons également le système de contrôle d'accès basé sur RDF qu'utilise le W3C depuis 2001 pour contrôler l'accès aux fichiers de ses serveurs : W3C ACL System[63]. Récemment, Hollenbach et al. [160] ont proposé une évolution de ce système pour contrôler les accès de manière décentralisée, à grande échelle, la partie autorisation du système associant à un répertoire une liste d'accès sous forme de métadonnées RDF. L'ontologie utilisée se nomme Basic Access Control Ontology[64] et est présentée comme une base solide pour développer des systèmes plus sophistiqués. Contrairement à l'ontologie AMO présentée ici, le contrôle d'accès à base de règles d'inférence n'est pas proposé (bien que les auteurs l'évoquent comme une possible future évolution), ni le contrôle au niveau des documents (plutôt qu'à celui des répertoires).

Les besoins en termes de gestion des droits d'accès dans le projet ISICIL s'apparentent à ceux des bibliothèques numériques dont Coyle [161] propose une synthèse. Cependant, une des problématiques essentielles aux bibliothèques numériques n'est pas pertinente dans le cadre d'ISICIL : celle du respect des copyrights des documents rendus accessibles et pour ce faire la protection des documents par des DRMs. Les documents manipulés par les veilleurs demeurent en effet dans l'intranet des entreprises ou bien sont des documents

[59] http ://www.xrml.org/
[60] http ://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
[61] http ://www.w3.org/TR/odrl/
[62] http ://www.oasis-open.org/committees/xacml/
[63] http ://www.w3.org/2001/04/20-ACLs
[64] http ://www.w3.org/ns/auth/acl

publics sur le web. Parmi les travaux sur la gestion des accès dans les bibliothèques numériques, citons ceux de Lagoze et al. [162] sur l'architecture Fedora de gestion des ressources numériques et ceux de Kruk et al. [164] sur la bibliothèque numérique sémantique JeromeDL.

Les auteurs de Fedora proposent un modèle appelé DARS (acronyme de Distributed Active Relationships) permettant d'associer des métadonnées aux objets d'une bibliothèque numérique, en particulier pour la gestion des droits d'accès. Bien qu'une partie du modèle de gestion des accès se trouve ainsi dans une ontologie, le système Fedora utilise cependant également des métadonnées XACML associées aux ressources qu'il manipule.

La gestion des accès dans JeromeDL repose sur l'ontologie EAC[65] (acronyme pour **E**xtensible **A**ccess **C**ontrol) développée par Kruk [163]. EAC permet d'associer des licences à des ressources, chaque licence correspondant à une politique d'accès. Par exemple, une licence peut indiquer que seules les personnes d'une organisation donnée peuvent accéder aux ressources associées. Le but d'EAC est donc de filtrer l'accès aux ressources alors que celui d'AMO est de définir des droits d'accès associés à des rôles d'utilisateur.

4.3.2.3 *Les standards du web social et du web de données*

Une spécificité de l'approche que nous adoptons avec AMO est de s'intégrer aux modèles adoptés dans les domaines du web social et du web de données. Plus précisément, SweetWiki utilise les ontologies FOAF[66] et SIOC[67] pour annoter ses ressources et AMO vient compléter ces ontologies pour permettre de gérer l'accès au contenu. FOAF (acronyme de Friend Of A Friend) est un vocabulaire RDF utilisé dans les réseaux sociaux pour décrire les personnes et les liens qu'elles entretiennent entre elles. SIOC (acronyme de **S**emantically **I**nterlinked **O**nline **C**ommunities) est un autre vocabulaire RDF qui modélise les concepts des applications sociales du web : forums, les blogs, les wikis. Il réutilise certains concepts de FOAF et d'autres ontologies populaires (Dublin Core, SKOS, etc.) et s'est imposé comme standard. Il est aujourd'hui intégré dans de nombreuses applications comme le moteur de blog WordPress et son adoption au sein du projet Linked Data[68] confirme sa popularité. Un module de SIOC existe également pour la modélisation des droits d'accès[69] mais est resté vide très longtemps. Tout récemment, à la fin de la publication de l'ontologie AMO (20 septembre 2009), ce module a été complété et reprend certaines idées. D'ores et déjà nous préciserons dans la partie suivante les correspondances évidentes.

FOAFRealm [158] est une extension de FOAF proposée pour filtrer collaborativement l'accès aux ressources en fonction des profils des utilisateurs et des relations qu'ils entretiennent dans un réseau social. Ce vocabulaire est par exemple utilisé dans

[65] http ://www.jeromedl.org/eac/1.0/spec/index.html
[66] http ://xmlns.com/foaf/spec/
[67] http ://sioc-project.org/ontology
[68] http ://linkeddata.org/
[69] http ://rdfs.org/sioc/access

JeromeDL pour un filtrage basé sur des mesures de confiance dans un réseau social. Un tel filtrage peut être complémentaire du contrôle d'accès basé sur les rôles des utilisateurs et les types d'accès aux ressources que permet AMO.

Enfin, le problème de l'autorisation d'accès aux ressources auquel répond AMO est connexe à celui de l'authentification des agents qu'il s'agira d'aborder dans le cadre général du projet ISICIL. Nous envisageons pour cela de reposer sur le protocole FOAF-SSL Story et al. [165].

4.3.3 L'ontologie AMO[70]

Dans un système de fichiers comme dans un système de gestion de contenu, des rôles (administrateur, propriétaire, etc.) sont associés aux utilisateurs ou catégories d'utilisateurs et différents types d'accès aux ressources (écriture, lecture, etc.) sont définis, l'accès aux ressources variant d'un utilisateur à l'autre selon son rôle. Cette analyse nous a conduits à définir un ensemble de classes et propriétés nécessaires à la description des droits d'accès à une ressource. C'est ce que nous décrivons dans la partie 4.3.3.1.

Si les systèmes de gestion de contenu partagent les mêmes principes généraux pour le contrôle de l'accès aux ressources, ils adoptent cependant des stratégies qui peuvent varier d'un système à l'autre. Pour permettre une adaptation facile de l'ontologie sur laquelle reposera la gestion de l'accès aux ressources en fonction de la stratégie adoptée, nous modélisons dans AMO cette stratégie sous la forme déclarative d'une base de règles d'inférence qui peuvent être modifiées à loisir sans incidence sur la base d'annotations des ressources à gérer. Nous décrivons dans la section 4.3.3.2 une base de règles correspondant à la stratégie de contrôle de l'accès aux ressources adoptée dans le wiki sémantique SweetWiki.

4.3.3.1 Classes et propriétés de AMO

AMO repose sur quelques principes fondateurs que partagent tous les systèmes de gestion de contenu. Les agents d'un système de gestion de contenu sont les utilisateurs, groupes d'utilisateurs, services qui interagissent avec le système. Ces agents ont des rôles ; par exemple, dans un wiki, il s'agit des rôles d'invité (agent non enregistré dans le système), de contributeur, d'administrateur ; d'autres rôles peuvent être modélisés pour d'autres types de système. A chaque rôle est associée une liste d'actions autorisées. Dans le cas d'un wiki, les actions possibles sur une ressource sont la création, la lecture, la modification et la destruction de contenu, la modification des droits, la modification de la liste des agents autorisés sur une ressource, la modification du type d'accès défini pour une ressource ; d'autres actions peuvent être modélisées pour d'autres types de systèmes.

Il existe différents types d'accès aux ressources du système. Dans le cas des wikis, une ressource peut être publique (tous les utilisateurs y ont accès en lecture et écriture), privée

[70] AMO est publique sur le web depuis avril 2009

(seuls les agents autorisés ont accès en lecture et écriture) ou semi-privée (accès libre en lecture, accès en écriture uniquement pour les agents autorisés). D'autres types d'accès peuvent être ajoutés pour d'autres types de systèmes. Enfin, les actions autorisées à un agent sur une ressource dépendent du rôle de l'agent et/ou du type d'accès défini pour la ressource.

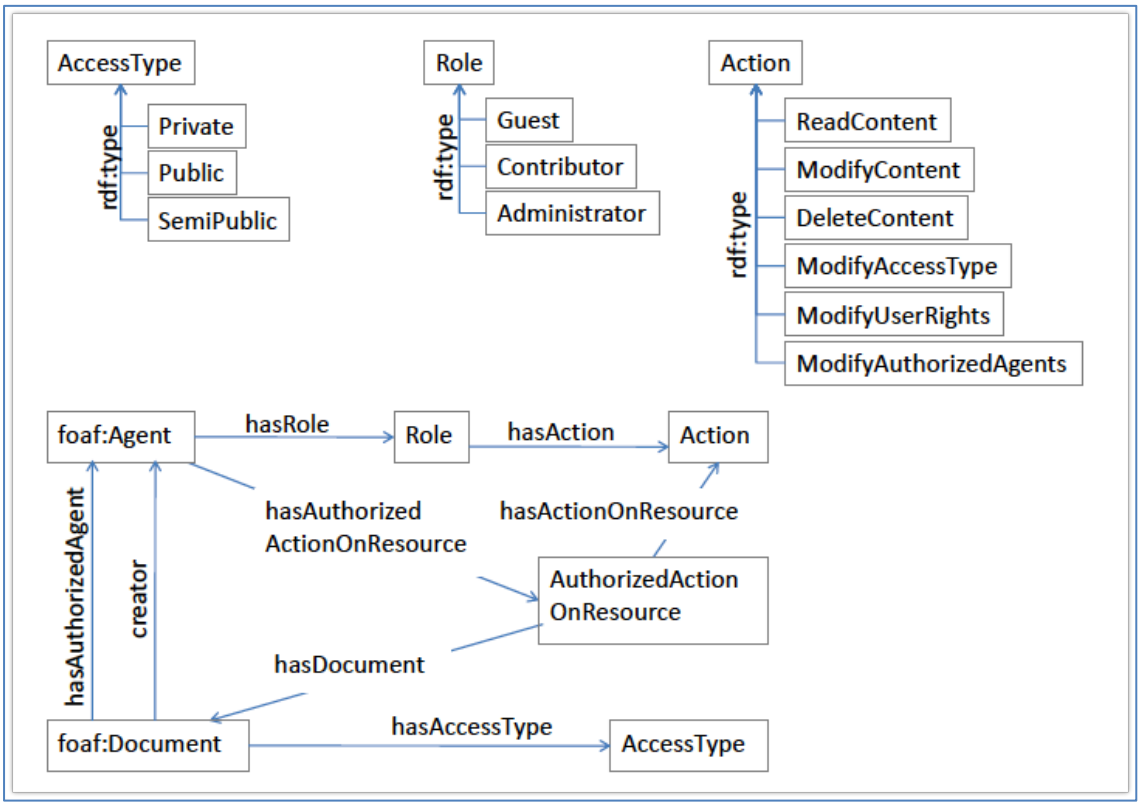


Figure 54. Classes et propriétés de l'ontologie AMO

L'ontologie AMO présentée sur la Figure 54 regroupe les concepts nécessaires pour représenter ces connaissances. Les trois classes `Role`, `Action` et `AccessType` sont centrales dans AMO. Probablement, ces classes pourront être alignées avec celles respectivement de `Role` dans *SIOC Core Ontology*, et `Permission` et `Status` nouvellement ajoutées dans le module *SIOC Access Ontology*. La classe `Role` d'AMO est la méta-classe des classes Administrator, Contributor et Guest. La classe Action est la méta-classe des classes `ReadContent`, `ModifyContent`, `DeleteContent`, `ModifyUserRights`, `ModifyAccessType` et `ModifyAuthorizedAgents`. Les classes `Private`, `Public` et `SemiPublic` sont instances de la méta-classe `AccessType`.

Trois classes du vocabulaire FOAF sont également centrales dans AMO : la classe `Agent` et sa sous-classe `Group`, et la classe `Document`. Elles sont utilisées comme domaine ou codomaine de propriétés AMO ainsi que dans les règles AMO.

Les propriétés `creator` et `hasAuthorizedAgent` associent un agent à un document (elles ont pour domaine la classe `Document` et pour codomaine la classe `Agent`) ; la propriété `hasRole` associe un rôle à un agent et la propriété `hasActionOnResource`

associe une action à un rôle ; la propriété `hasAccessType` associe un type d'accès à un document.

En outre, pour représenter dans un modèle de propriétés binaires la relation ternaire qui stipule qu'un agent est autorisé à effectuer une action sur une ressource, nous avons réifié cette relation en introduisant une sous-classe `AuthorizedActionOnResource` de la classe `Action`, une propriété `hasAuthorizedActionOnResource` qui associe une instance de `AuthorizedActionOnResource` à un agent, et les propriétés `hasDocument` et `hasAction` qui associent à une instance de `AuthorizedActionOnResource` respectivement un document et une action.

AMO est un vocabulaire RDFS, qui peut donc être utilisée pour annoter dans le langage RDF les ressources dont il s'agit de contrôler l'accès.

4.3.3.2 *Base de règles de AMO*

Les systèmes de gestion de contenu adoptent des stratégies de contrôle d'accès aux ressources qui peuvent varier d'un système à l'autre. Plutôt que de faire varier les annotations des ressources en fonction des stratégies de contrôle auxquelles ces ressources seront soumises, nous proposons de modéliser de façon déclarative la stratégie de contrôle dans l'ontologie AMO, sous la forme d'une base de règles, certaines règles pouvant varier en fonction de la stratégie modélisée.

La base de règles présentée ici est celle de SweetWiki dont la stratégie de contrôle d'accès s'inspire de celle du moteur de wiki open source DekiWiki[71] très largement utilisé et qui a servi de base au wiki que nous avons finalement développé pour le projet ISICIL : SweetDeki (le projet ISICIL a finalement adopté AMO pour la gestion des droits d'accès par l'ensemble des outils mis en œuvre, dont ce dernier wiki).

Par défaut, les administrateurs ont tous les droits sur toutes les ressources. Les contributeurs ont tous les droits relatifs au contenu des ressources ; ceux déclarés agents d'une ressource par l'auteur de celle-ci ont en outre certains droits d'administration sur cette ressource. Les invités ont uniquement le droit de lecture du contenu des ressources.

La Figure 55 résume les droits d'accès à une ressource en fonction de son type d'accès et du rôle de l'utilisateur qui cherche à accéder à la ressource.

[71] http ://www.mindtouch.com/, détails ici : http://miageprojet2.unice.fr/Intranet_de_Michel_Buffa/Droits_d'accès_dans_SweetWiki

	Public	Semi-Public	Private
Guest	ReadContent	ReadContent	
Contributor	ReadContent	ReadContent ModifyContent DeleteContent	
AuthorizedAgent	ReadContent ModifyContent DeleteContent ModifyAuthorizedAgents ModifyAccessType		
Administrator	ReadContent ModifyContent DeleteContent ModifyAuthorizedAgents ModifyAccessType ModifyUserRights		

Figure 55. Une stratégie de contrôle des droits d'accès modélisée dans AMO.

Nous modélisons cette stratégie dans AMO de façon déclarative par six règles d'inférence correspondant chacune à une situation décrite sur la Figure 55. Par exemple, la règle 1 ci-dessous spécifie les droits attribués aux agents donnés à une ressource. D'autres règles stipulent des lois générales telles que « un membre d'un groupe hérite du ou des rôle(s) attribués à son groupe » (règle 2) ou encore « le créateur d'une ressource est un agent de cette ressource » (règle 3).

Ces règles sont exprimées dans le langage SPARQL en utilisant des requêtes de la forme `CONSTRUCT / WHERE` : une telle requête permet de construire des graphes RDF en substituant aux variables de sa clause `CONSTRUCT` les valeurs qui satisfont sa clause `WHERE` (retrouvées en recherchant les appariements possibles de sa clause `WHERE` avec les données RDF interrogées). Une requête `CONSTRUCT/WHERE` peut donc être vue comme une règle appliquée en chaînage avant la clause `WHERE` étant la prémisse et la clause `CONSTRUCT` la conclusion. Ce format de règle peut cependant aussi bien être utilisé pour des règles appliquées en chaînage arrière, comme c'est le cas dans le moteur sémantique Corese.

Règle 1

```
CONSTRUCT {
?agent amo:hasAuthorizedActionOnResource ?a ?a amo:hasResource
    ?resource
?a amo:hasActionOnResource amo:ReadContent.
?a amo:hasActionOnResource amo:ModifyContent.
?a amo:hasActionOnResource amo:DeleteContent.
?a amo:hasActionOnResource amo:ModifyAccessType.
?a amo:hasActionOnResource amo:ModifyAuthorizedAgents
}
WHERE {
    ?resource rdf:type foaf:Document. ?resource
    amo:hasAuthorizedAgent ?agent
}
```

Règle 2

```
CONSTRUCT {
?agent amo:hasRole ?role
}
WHERE {
    ?group amo:hasRole ?role
    ?group foaf:member ?agent
}
```

Règle 3

```
CONSTRUCT {
    ?resource amo:hasAuthorizedAgent ?agent
}
WHERE {
    ?resource amo:creator ?agent
}
```

Cette modélisation déclarative de la stratégie de gestion des droits d'accès assure une grande facilité de maintenance. Le changement des droits d'une catégorie d'utilisateurs - et ce pour toutes les ressources concernées - ne nécessitera que l'ajout ou la suppression de triplets dans la conclusion d'une règle. De même, l'ajout de nouveaux rôles ne demanderait que l'ajout d'une classe représentant ce rôle et des règles représentant les droits d'accès associés à ce rôle.

4.3.4 Gestion des droits d'accès dans SweetWiki

L'ontologie AMO est utilisée dans le projet ISICIL pour annoter des ressources partagées par un réseau social de veilleurs d'entreprise. La gestion de l'accès à ces ressources dans le moteur SweetWiki repose sur (1) l'exploitation de ces annotations sémantiques, (2) des inférences sur ces annotations basées sur les règles d'AMO et (3) la formulation de requêtes SPARQL afin de retrouver des connaissances relatives aux accès autorisés à un utilisateur donné sur une ressource donnée. Dans SweetWiki les annotations des ressources reposent sur les ontologies FOAF, SIOC et AMO et des requêtes SPARQL

sont utilisées dans la plupart des fonctionnalités implémentées : les annotations RDF nourrissent le moteur sémantique CORESE embarqué dans SweetWiki. En particulier, en utilisant les possibilités de recherche approchée de Corese (Corby et al. [166]) et un système de tagging sémantique des documents, SweetWiki offre une navigation "intelligente" au travers de suggestions.

4.3.4.1 *Annotation des ressources d'ISICIL avec AMO*

Lors de la création d'une page wiki, l'identité de son créateur est enregistrée ainsi que le type d'accès à la page décidé par celui-ci et éventuellement un ou plusieurs agents autorisés sur la page, désignés également par le créateur. Dans SweetWiki ces informations sont traduites sous la forme d'annotations RDF associées aux pages créées. Par exemple, l'annotation 1 suivante résulte de la création d'une page wiki privée par l'utilisateur `AnnaKolomoiska` qui déclare `MichelBuffa` agent autorisé de cette page. Cette annotation utilise les propriétés `creator`, `hasAuthorizedAgent` et `hasAccessType` d'AMO (et la classe `WikiArticle` de SIOC).

Annotation 1

```
<rdf:RDF xmlns="http://sweetwiki.i3s.unice.fr/AMO.rdfs#" ... >
   <sioc:WikiArticle rdf:about="#TestPage">
...
        <creator rdf:resource="#AnnaKolomoiska"/>
        <hasAuthorizedAgent rdf:resource="#MichelBuffa"/>
        <hasAccessType rdf:resource="#Private"/>
   </sioc:WikiArticle>
</rdf:RDF>
```

Lors de l'enregistrement d'un utilisateur sur SweetWiki, cette information est également représentée sous la forme d'une annotation RDF. Par exemple, l'annotation 2 suivante stipule que `MichelBuffa` est un contributeur du wiki. Elle utilise pour cela la classe `Contributor` et la propriété `hasRole` d'AMO (et la classe `Agent` de FOAF).

Annotation 2

```
<rdf:RDF xmlns="http://seetwiki.i3s.unice.fr/AMO.rdfs#" ... >
   <foaf:Agent rdf:about="#MichelBuffa">
      ...
      <hasRole rdf:resource="#Contributor"/>
   </foaf:Agent>
</rdf:RDF>
```

D'autres annotations expriment des connaissances relatives aux groupes d'utilisateurs du wiki. Par exemple, l'annotation 3 stipule que `AnnaKolomoiska` et `CatherineFaron` sont membres du groupe des administrateurs du wiki. Elle utilise

pour cela la propriété `hasRole` d'AMO (et les classes `Group` et `Agent` et la propriété `member` de FOAF).

Annotation 3

```
<rdf:RDF xmlns="http://seetwiki.i3s.unice.fr/AMO.rdfs#" ... >
   <foaf:Group rdf:about="#AdminGroup">
      <foaf:member>
         <foaf:Agent rdf:about="#AnnaKolomoiska"/>
      </foaf:member>
      <foaf:member>
         <foaf:Agent rdf:about="#CatherineFaron"/>
      </foaf:member>
      <hasRole rdf:resource="#Admin"/>
   </foaf:Group>
</rdf:RDF>
```

4.3.4.2 *Inférences exploitant la base de règles de AMO*

Appliquées sur les annotations des ressources d'ISICIL, les règles de l'ontologie AMO permettent d'inférer les droits des utilisateurs du wiki sur ces ressources. Considérons par exemple à nouveau la règle 1 qui illustre la section 4.3.3.2. Sa prémisse s'apparie avec l'annotation 1 qui illustre la section 4.3.4.1: la ressource `TestPage` est de type `Document` et est en relation `hasGivenAgent` avec l'utilisateur `MichelBuffa`. Appliquée sur l'annotation 1, cette règle permet de conclure que l'utilisateur `MichelBuffa` a les droits de lecture, modification et destruction du contenu de la ressource `TestPage` annotée et ceux de modification de son type d'accès et de sa liste d'agents.

De la même manière, la règle 2 appliquée sur l'annotation 3 permet de déduire que l'utilisateur `CatherineFaron` possède le rôle d'administrateur. Une autre règle d'AMO décrivant les droits d'un agent ayant le rôle d'administrateur sur toute ressource permet alors de conclure que l'utilisateur `CatherineFaron` possède tous les droits sur la ressource `TestPage`

Enfin, les règles 1 et 3 appliquées sur l'annotation 1 permettent de déduire que l'utilisateur `AnnaKolomoiska` créateur de la ressource `TestPage` a les droits d'un agent sur cette ressource (lecture, modification et destruction du contenu et modification du type d'accès et de la liste d'agents).

4.3.4.3 *Requêtes SPARQL pour la gestion des droits d'accès*

L'accès à une ressource particulière par un utilisateur donné dépend, comme l'ensemble des actions dans SweetWiki, de la réponse du moteur CORESE à une requête sémantique lancée sur la base d'annotations des ressources. Pour cela, CORESE combine chaînage arrière sur la base de règles AMO et appariement de graphes sur la base d'annotations. Par exemple, la requête SPARQL suivante indique si l'utilisateur `CatherineFaron` possède le droit de modification du contenu de la ressource `TestPage` :

Requête 1

```
prefix amo: <http://sweetwiki.unice.fr/AMO.rdfs#>
ASK {
   <http://sweetwiki.unice.fr#CatherineFaron>
     amo:hasAuthorizedAccessOnResource ?x
     ?x amo:hasActionOnResource amo:ModifyContent
     ?x amo:hasResource http://sweetwiki.unice.fr#TestPage
}
```

En fonction des différentes fonctionnalités de SweetWiki, d'autres requêtes sont mises en œuvre. Par exemple, la requête SPARQL suivante permet de retrouver la liste des utilisateurs ayant des droits sur la ressource `TestPage` et pour chacun la liste de ses actions autorisées sur `TestPage` :

Requête 2

```
prefix amo: http://sweetwiki.unice.fr/AMO.rdfs#
SELECT ?agent ?action {
   ?agent amo:hasAuthorizedAccessOnResource ?x
   ?x amo:hasActionOnResource ?action
   ?x amo:hasResource <http://sweetwiki.unice.fr#TestPage>
} order by ?agent
```

4.3.5 Bilan sur la gestion des droits d'accès sémantique intégrée au projet ISICIL

Nous avons présenté l'ontologie AMO dédiée à la gestion de l'accès dans les systèmes de gestion de contenu. AMO consiste d'une part en un ensemble de classes et propriétés qui permettent d'annoter les ressources dont il s'agit de contrôler l'accès. AMO propose d'autre part une base de règles d'inférence qui représentent de façon déclarative une stratégie d'accès aux ressources ; cette base peut être facilement modifiée, sans incidence sur les annotations des ressources, en fonction de la stratégie à mettre en œuvre sur telle ou telle application web à gérer. Nous montrons l'utilisation qui peut être faite d'AMO par des exemples d'annotations de ressources dans SweetWiki et nous validons ce premier prototype par des requêtes SPARQL permettant d'interroger la base d'annotations relatives aux accès à l'aide du moteur sémantique CORESE.

AMO s'appuie sur FOAF et sur SIOC et se rend ainsi compatible avec les applications qui ont déjà basé leur développement sur ces deux ontologies populaires pour décrire leurs ressources.

L'utilisation d'une base de règles permet de s'affranchir de l'implémentation complexe des mécanismes de calcul des droits, qui font appel à l'héritage (héritage des rôles des groupes auquel appartient un agent, union des actions permises par ces rôles), à l'ordre des opérateurs (interdire puis autoriser, ou le contraire...), à des mécanismes implicites (le créateur d'une ressource bénéficie automatiquement de certains droits sur cette même ressource). Dans l'état de l'art nous avons vu qu'aucune des approches sémantiques de gestion des accès n'utilise de règles, les auteurs de l'ontologie ACL du W3C ayant simplement évoqué cette possibilité d'extension. Comparée aux langages XML antérieurs

au web sémantique, comme XACML qui bénéficient de bibliothèques logicielles importantes et complexes pour implémenter toutes les stratégies de gestion des droits, et qui ont demandé un travail de développement considérable, AMO en revanche demeure simple à mettre en œuvre et à étendre (par exemple en réutilisant l'ontologie FOAF Realm pour définir des règles d'accès basées sur des mesures de confiance).

Nous avons présenté des exemples d'utilisation d'AMO avec les documents produits par SweetWiki. AMO est également utilisée par d'autres applications dans le cadre du projet ANR ISICIL. Précisément, les classes et propriétés d'AMO sont utilisées dans les annotations des documents du serveur des profils des utilisateurs. L'ontologie SemSNI utilisée pour ces annotations, sert à modéliser les interactions entre utilisateurs d'un réseau social. Elle fait référence à AMO pour la définition des droits d'accès aux ressources partagées par chaque utilisateur. SemSNI et AMO ont été également utilisées pour spécifier les accès aux documents partagés par les utilisateurs du réseau social Ipernity.com (un site français inspiré de FaceBook spécialisé sur le partage de photos), voir Erétéo et al. [154]. PrototyÒpée dans SweetWiki, la couche de gestion des accès basée sur AMO est en cours d'implémentations dans les différentes applications du projet ISICIL.

4.4 Conclusion

Nous concluons ici le chapitre consacré aux « wikis sémantiques d'applications » qui m'a occupé pendant plusieurs années. Pendant le développement de SweetWiki, le premier wiki sémantique que j'ai entièrement implémenté et co-conçu avec Fabien Gandon, nous avons bien senti que l'avenir n'était plus dans des outils « stand alone » mais bien dans leur intégration au sein d'un « réseau social d'entreprise ».

L'utilisation d'un wiki laisse des traces intéressantes sur l'activité de ses utilisateurs (co-édition, tagging social, commentaires, recherches, etc.) mais d'autres outils sont également concernés : gestionnaire de favoris partagés, moteur de recherche ; et plus généralement l'ensemble de ce que l'on peut appeler un « réseau social ».

Dans le cadre du projet ANR ISICIL, qui tourne autour de l'outillage logiciel de veilleurs en entreprise, nous avons développé un framework dans lequel des applications web (ou mobiles) clientes collaboratives nourriraient le réseau social avec des métadonnées, et où un serveur centralisé s'occuperait de la gestion de ces métadonnées sémantiques.

Sous ma direction, l'équipe Kewi du laboratoire I3S a développé ainsi un « wiki plus industriel », basé sur le moteur existant DekiWiki. Ce wiki d'applications, solide et déjà utilisé par de grandes organisations comme Microsoft, Amazon, etc. est open source et extensible. Nous avons donc développé un ensemble d'extensions et plugins pour apporter la dimension sémantique qui lui manquait et pour générer les traces qui nous intéressaient. Ce nouveau DekiWiki « boosté au web sémantique », a été intitulé SweetDeki (mélange de l'ancien wiki SweetWiki et de DekiWiki).

En outre, ce wiki étant pilotable par web services, nous l'avons « connecté » au serveur central d'ISICIL pour qu'il puisse à la demande générer des rapports d'activité, visualiser

dans des documents les relations entre utilisateurs partageant les mêmes intérêts, des documents contenant des bouts annotés de pages web « copiées et partagées » par des veilleurs etc. Dans l'autre sens, toute activité sur le wiki générait des métadonnées qui venaient enrichir le réseau social et la folksonomie.

J'ai également développé avec Catherine Faron, l'ontologie AMO (Access Management Ontology), qui a permis d'implémenter la gestion des droits d'accès aux ressources partagées par le réseau social du projet ISICIL sous forme d'annotations et de requêtes SPARQL. L'utilisation de cette ontologie a permis une grande souplesse dans la gestion de ces droits : utilisation du polymorphisme mais surtout de règles (supportées par le moteur CORESE). Ce système de gestion a été exploité par l'ensemble des logiciels clients développés (wikis, extensions de navigateur, moteur de recherche, outils d'exploration du réseau etc.)

La modélisation du réseau social ISICIL dans lequel le wiki sémantique DekiWiki s'intégrait est l'objet de la thèse de Guillaume Ereteo (que j'ai co-encadrée à 50%), présenté en détail dans le chapitre 5.

La gestion du tagging social a été entièrement repensée depuis SweetWiki (thèse de Freddy Limpens que j'ai co-encadrée à 50%). Ce travail est présenté dans le chapitre 6.

Vers la fin du projet ANR ISICIL, j'ai encadré à 100% la thèse de Pavel Arapov sur un wiki facilitant la programmation d'applications tirant partie du web de données et des frameworks du web sémantique. Ce wiki est un hybride entre les wikis sémantiques d'applications et permet de coder des applications directement dans le navigateur, avec les langages standards du web (HTML, CSS et JavaScript). C'est à la fois un wiki d'application classique mais aussi un IDE en ligne spécialisé pour tirer parti des technologies du web sémantique. WikiNEXT exploitait les technologies HTML5, server-side JavaScript, encore émergentes à l'époque. Il embarquait un moteur SPARQL complet codé en JS et assurait la persistance des métadonnées dans une base NoSQL.

De 2005 à 2014 j'ai donc travaillé sur les wikis au sens large, mais parallèlement à ce travail, je me suis intéressé au « web social » : analyse des réseaux sociaux, social tagging, détection de communautés, etc. J'ai notamment poussé (avec Fabien Gandon) l'équipe Edelweiss de l'INRIA à faire du « web social » un de ses thèmes de recherche prioritaires (l'équipe Kewi à laquelle j'appartenais fusionnera en 2012 avec l'équipe Edelweiss).

Avant de passer aux chapitres suivants, relatifs à mes travaux sur les réseaux socio-sémantiques, quelques réflexions sur ce que « nous n'avons pas fait » dans ce travail sur les wikis, et sur les pistes vers lesquelles nous aurions pu nous diriger…

Parmi les outils collaboratifs les plus connus aujourd'hui, en 2018, il est probable que l'on cite Google Docs, Trello, Slack, Mattermost etc. Ces outils de type Office, de gestion de projets agiles, de chat, de partage de documents ont tous une particularité : ils sont synchrones et autorisent l'interaction simultanée de leurs utilisateurs sur les mêmes ressources. Avec WikiNext nous avions expérimenté l'édition synchrone de documents

en intégrant ShareJS[72], une librairie JavaScript implémentant des algorithmes de « operational transformation » rendus populaires à l'époque par Google Wave[73]. Nous n'avons finalement pas poursuivi ce travail, ayant rencontré rapidement des limitations relatives aux outils choisis et aux fonctionnalités que nous avions dans le wiki (la fusion et le maintien de cohérence des annotations sémantiques n'était pas supporté par les formats attendus par la librairie ShareJS).

Ces problématiques appartiennent notamment à un domaine de recherche particulièrement intéressant, portant sur les « wikis distribués », qui regroupe diverses approches incluant les wikis pair-à-pair, les wikis mobiles, les wikis pouvant fonctionner offline (sans connexions ou avec connexions intermittente), les wikis fédérés etc.

Ces wikis distribués fonctionnent en plusieurs instances sur différents serveurs et distribuent leurs pages en assurant la disponibilité des données sur toutes les instances. Ainsi plusieurs utilisateurs peuvent travailler simultanément sur les mêmes documents qui seront versionnés et synchronisés. Si une instance s'arrête, les documents restent disponibles sur les autres, etc. Nous renvoyons ici le lecteur vers des articles de synthèse comme « Distributed Wikis : A Survey » par Davous et al. [217].

On retrouve également des wikis *sémantiques* distribués comme Swooki [218]. Swooki adresse le problème de la fusion des pages wikis (sujet classique dans les wikis pair à pair) mais contenant des annotations sémantiques. Les algorithmes classiques de fusion utilisés dans les wikis pair-à-pair ont été conçus pour des structures linéaires qui ne considèrent pas les annotations sémantiques. C'est le problème que nous avions rencontré lors de l'intégration de ShareJS dans WikiNext et que Swooki a adressé. Dans la thèse de Pavel Arapov sur WikiNext, nous avons finalement préféré nous spécialiser sur la thématique du « wiki sémantique d'application centralisé » et abandonner le côté « wiki distribué » qui aurait mérité un travail de recherche spécifique et qui s'éloignait trop du sujet initial.

[72] https://github.com/josephg/ShareJS. A noter que ShareJS est devenu ShareDB : https://github.com/share/sharedb

[73] Google Wave était une application web de messagerie et de travail collaboratif, créée par Google, dont le concept mélange les notions de services de courrier électronique, de messagerie instantanée, de wiki et de réseautage social, le tout associé à un correcteur orthographique et un traducteur instantané, et à de nombreuses autres possibilités développées par des éditeurs tiers. Il est basé sur des algorithmes de « operational transformation » (OT), une technologie permettant de prendre en charge une gamme de fonctionnalités de collaboration dans des systèmes logiciels collaboratifs avancés. OT a été inventé à l'origine pour la maintenance de la cohérence et le contrôle de la simultanéité dans l'édition collaborative de documents en texte brut.

5. Réseaux socio-sémantiques (mes travaux sur la période 2006-2012)

Contexte

Nous l'avons évoqué dans les chapitres précédents, les premières applications sociales du Web 2.0 de l'époque, les blogs, les wikis, ont montré qu'on pouvait analyser les actions et interactions de leurs utilisateurs. Deux personnes éditent la même page d'un wiki, elles s'intéressent à la même chose : ce dont parle la page. Une personne laisse un commentaire sur un article de blog, elle s'y intéresse, etc. On va rapidement considérer les wikis, les blogs et les sites de partage de média comme des réseaux sociaux implicites (je voudrais trouver toutes les personnes « qui contribuent le plus" à ce wiki », « qui s'intéressent à 'la moto' sur YouTube », qui « ont posté des vidéos de moto », « commenté des vidéos de moto », « taggué des vidéos de moto », ou plus généralement, « se sont intéressées à des vidéos qui sont plus ou moins proches du thème de la moto ».

En 2005, Facebook apparaît et systématise encore plus la notion de réseau social, en ajoutant à ces liaisons implicites des relations explicites (« je t'ajoute comme ami » et « je suis ton activité »).

L'analyse des réseaux sociaux (*Social Network Analysis* ou *SNA* en anglais) est une science assez ancienne, qui fournit des algorithmes pour caractériser la structure des réseaux sociaux, les positions stratégiques des membres dans ces réseaux (le plus central, le plus influent, le plus intermédiaire etc., elle permet d'identifier les sous-réseaux ou communautés en décomposant le graphe du réseau social en fonction du profils des personnes, de leurs activités, de leurs relations (communautés d'intérêt, de pratiques, etc.). Les plateformes comme Facebook forment aujourd'hui d'énormes réseaux sociaux, permettant aux gens de se connecter, d'interagir et de partager leurs activités en ligne.

Parallèlement à mes travaux sur les wikis, j'ai mené naturellement une activité de recherche sur le thème de l'analyse des réseaux sociaux sémantiques, les wikis étant des vecteurs de traces et d'interactions sociales à même de nourrir les interactions entre utilisateurs (coédition, tagging social, etc.)

Contributions

A l'époque, je me suis intéressé à l'analyse des réseaux sociaux (SNA) sous l'angle du Web, en exploitant les technologies du Web Sémantique permettant d'effectuer du raisonnement sur les graphes, notamment au travers d'extensions du langage SPARQL implémentées par le moteur sémantique CORESE. Nous avons en particulier étendu les opérateurs classiques de la SNA en utilisant des frameworks du web sémantique, pour inclure la sémantique des représentations à base de graphes lors de l'analyse et pour mieux faire face à la diversité des relations et interactions entre utilisateurs. Nous avons notamment validé nos contributions pour analyser un véritable réseau social comprenant

60 000 utilisateurs, et développé une ontologie proposant un modèle enrichi et extensible du réseau social et de ses utilisateurs (article ISWC 2009, rang A)

Collaboration, encadrement

J'ai co-encadré la thèse de Guillaume Ereteo ("*Semantic Social Network Analysis")* avec Fabien Gandon (encadrement 50%). Ce travail s'est intégré dans le projet ANR ISICIL dont j'étais responsable scientifique pour l'équipe KEWI du laboratoire I3S.

Publications principales ayant résulté de ces travaux

Guillaume Erétéo, Fabien Gandon, Michel Buffa, Olivier Corby*: Analysis of a Real Online Social Network Using Semantic Web Frameworks*. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 180–195. Springer, Heidelberg (2009). Accept rate 17%. Only paper from a french research group. PDF : http://www-sop.inria.fr/members/Fabien.Gandon/docs/ISWC2009_ereteo_et_al.pdf

Fabien Gandon, Talel Abdessalem, Michel Buffa and Al., *ISICIL: Information Semantic Integration through Communities of Intelligence online,* 10th IFIP Working Conference on Virtual Enterprises, Thessaloniki, Greece, 7-9 October 2009

Guillaume Erétéo, Fabien Gandon, Michel Buffa and Olivier Corby. *Semantic Social Network Analysis*. In Proc. Web Science WebSci'09, Athens, Greece, March 2009.

Guillaume Erétéo, Michel Buffa, Fabien Gandon, Mylène Leitzelman, and Freddy Limpens. *Leveraging Social Data with Semantics*, In W3C Workshop on the Future of Social Networking, Barcelona, Spain, 2009

Guillaume Erétéo, Michel Buffa, Fabien Gandon, Patrick Grohan, Mylène Leitzelman, Peter Sander. *A State of the Art on Social Network Analysis and its Applications on a Semantic Web*. SDoW2008, Workshop at ISWC2008, Karlsruhe, Germany (2008)

Résumé de cette section

Nous avons pu assister, entre 2000 et 2009, au déploiement d'un paysage de médias sociaux où « des outils d'expression en ligne ont permis aux utilisateurs de discuter et d'agréger leur vie sociale ». Ces « outils de partage » ont notamment donné la possibilité aux utilisateurs de publier et de partager du contenu de manière très intuitive, et les "outils de connexion sociales" ont facilité les interactions les uns avec les autres [4]. Des plateformes, comme Facebook, Orkut, Hi5, etc., ont été au centre de ce paysage à l'époque et sont à l'origine des réseaux sociaux tels qu'on les connaît aujourd'hui.

Il était possible, en 2006, de publier et partager des favoris web (ou "signets") sur le site del.icio.us (racheté par Yahoo, aujourd'hui disparu), des flux RSS ou des messages de microblog sur le fil d'actualité Facebook, mais aussi grâce à des applications Facebook dédiées. Cette intégration de différents moyens de publication et de socialisation a permis de partager, de recommander et de propager rapidement des informations sur le réseau social, de déclencher des réactions et enfin de l'enrichir.

L'analyse des réseaux sociaux, ou "Social Network Analysis (SNA)" en anglais, propose des algorithmes de graphes pour caractériser la structure d'un réseau social, les positions stratégiques, les sous-réseaux spécifiques et les activités de réseautage [22]. Les applications collaboratives capturent désormais de plus en plus d'aspects des réseaux

sociaux physiques et des interactions humaines de manière décentralisée. Ces données riches et diffuses ne peuvent pas être représentées en utilisant uniquement des graphes bruts comme dans les algorithmes SNA classiques, sans perte de connaissances.

Les frameworks du Web sémantique répondent à ce problème de représentation et d'échange de connaissances sur ces réseaux sociaux avec un modèle de graphe typé riche (RDF), un langage de requête (SPARQL) et des frameworks de définition de schéma (RDFS et OWL). En adaptant ces technologies sémantiques à la SNA sur les réseaux sociaux en ligne, nous avons montré dans [10], en 2008-2010, qu'il était possible d'appliquer et de paramétrer la SNA sur des représentations riches, typées des réseaux sociaux.

Nous avons proposé à l'époque un cadre pour exploiter directement des représentations RDF de réseaux sociaux, en utilisant des moteurs de recherche sémantiques exploitant les informations riches, typées et structurées des graphes étiquetés ainsi formés [1] [11].

De nombreux algorithmes sont disponibles pour détecter les structures sociales, les rôles et les positions. Mais un aspect de notre travail, qui nous a différencié des autres approches à l'époque, a consisté à aller au-delà de la simple application des algorithmes classiques aux réseaux sociaux décrits avec les technologies web sémantiques. Nous avons en effet étendu certains algorithmes existants et nous en avons créé de nouveaux, afin de gérer les cycles de vie des communautés, en tenant compte non seulement de la structure des graphes mais aussi de la sémantique des primitives ontologiques utilisées pour marquer les nœuds et les arcs du réseau social.

Enfin, nous avons mené une analyse de réseau social sémantique réel du service de partage de photographies Ipernity.com, et présenterons les résultats que nous avons obtenus.

5.1 Analyse des réseaux sociaux : extension des opérateurs classiques en utilisant des frameworks du Web sémantique

Entre 2005 et 2010 nous avons assisté au déploiement d'un paysage de médias sociaux où « les outils d'expression permettent aux utilisateurs de s'exprimer, de discuter et d'agréger leur vie sociale », « les outils de partage permettent aux utilisateurs de publier et de partager du contenu », et « les outils de réseautage permettent aux utilisateurs de chercher, de se connecter et d'interagir les uns avec les autres » [4]. Les plateformes sociales, comme Facebook, Orkut, Hi5, etc., sont déjà, en 2009, au centre de ce paysage car elles permettent d'héberger et d'agréger ces différentes applications sociales. Vous pouvez publier et partager vos favoris sur del.icio.us, vos flux RSS ou vos messages microblog via le flux d'actualités de Facebook, grâce à des applications Facebook dédiées. Cette intégration de divers moyens de publication et de socialisation nous permet de partager, de recommander et de propager rapidement l'information à notre réseau social, de déclencher des réactions et enfin de l'enrichir.

De plus en plus de solutions sociales (p. ex. SocialText[74]) sont déployées dans les intranets d'entreprise pour reproduire les réussites de partage d'information du Web dans les intranets des organisations (voir section 2.4 de ce manuscrit). Cependant, l'avantage de ces plates-formes est souvent entravé lorsque le réseau social devient si vaste que l'information pertinente est souvent perdue dans un flux écrasant de notifications d'activités. L'organisation de cette énorme quantité d'informations est l'un des défis majeurs du web 2.0[75] pour son acceptation dans les contextes d'entreprise et pour atteindre le plein potentiel de l'Entreprise 2.0, c'est-à-dire l'utilisation efficace des technologies du web 2.0 comme les blogs et les wikis au sein de l'Intranet [17].

L'Analyse des Réseaux Sociaux (*Social Network Analysis* ou « SNA » en anglais) propose des algorithmes de graphes pour caractériser la structure d'un réseau social, les positions stratégiques, les sous-réseaux spécifiques et les activités de réseautage [22]. Les applications collaboratives capturent de plus en plus d'aspects des réseaux sociaux physiques et des interactions humaines de manière décentralisée. De telles données riches et diffuses ne peuvent pas être représentées en utilisant uniquement des graphes bruts comme c'est le cas dans les algorithmes classiques de SNA, sans perte de connaissances importantes.

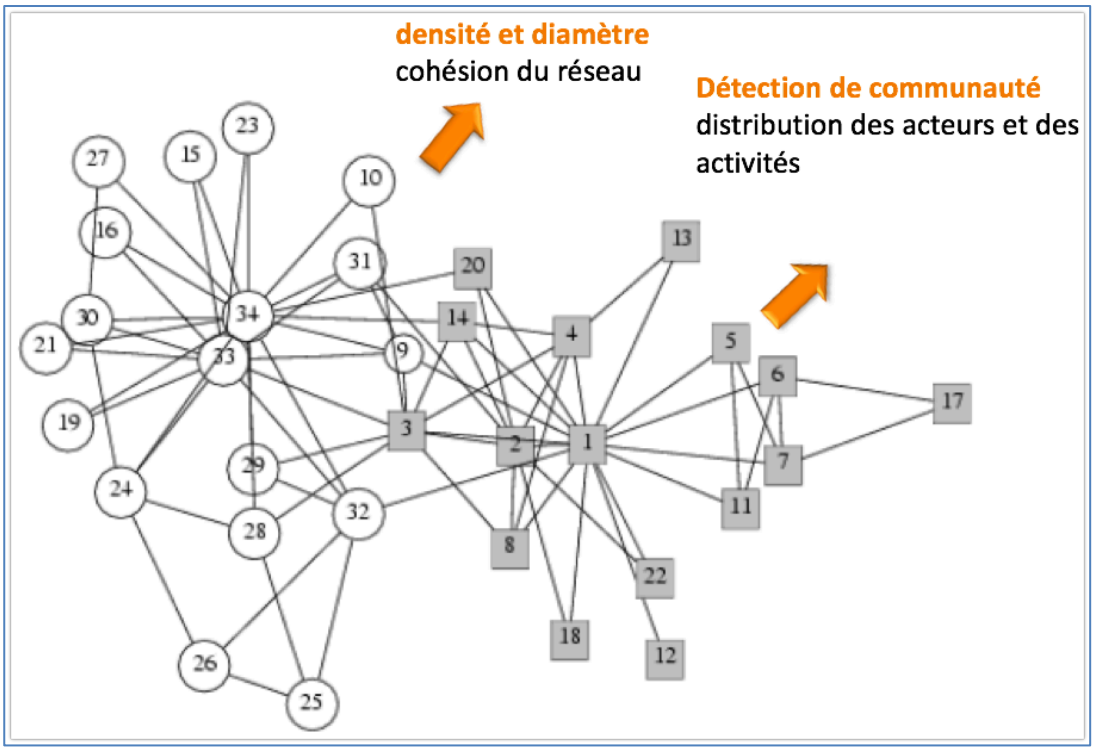


Figure 56. Quelques exemples d'indices typiques de l'analyse des réseaux sociaux. Ici densité et diamètre, détection de communautés.

Les frameworks du Web sémantique répondent à ce problème de représentation et d'échange de connaissances sur ces réseaux sociaux avec un riche modèle de graphe typé

[74] http://www.socialtext.com/
[75] http://oreilly.com/web2/archive/what-is-web-20.html

(RDF[76]), un langage de requête (SPARQL) et des frameworks de définition de schémas (RDFS et OWL). En faisant un état de l'art des travaux en SNA sur les réseaux sociaux en ligne, nous avons mis en évidence dans [10] le manque de techniques pour appliquer la SNA sur de riches représentations typées. Nous avons proposé un framework pour exploiter directement les représentations RDF des réseaux sociaux à l'aide des moteurs de recherche sémantique, afin de tirer profit de la richesse de l'information qu'ils détiennent et en particulier des relations typées qui forment ces graphes labellisés [1], [11].

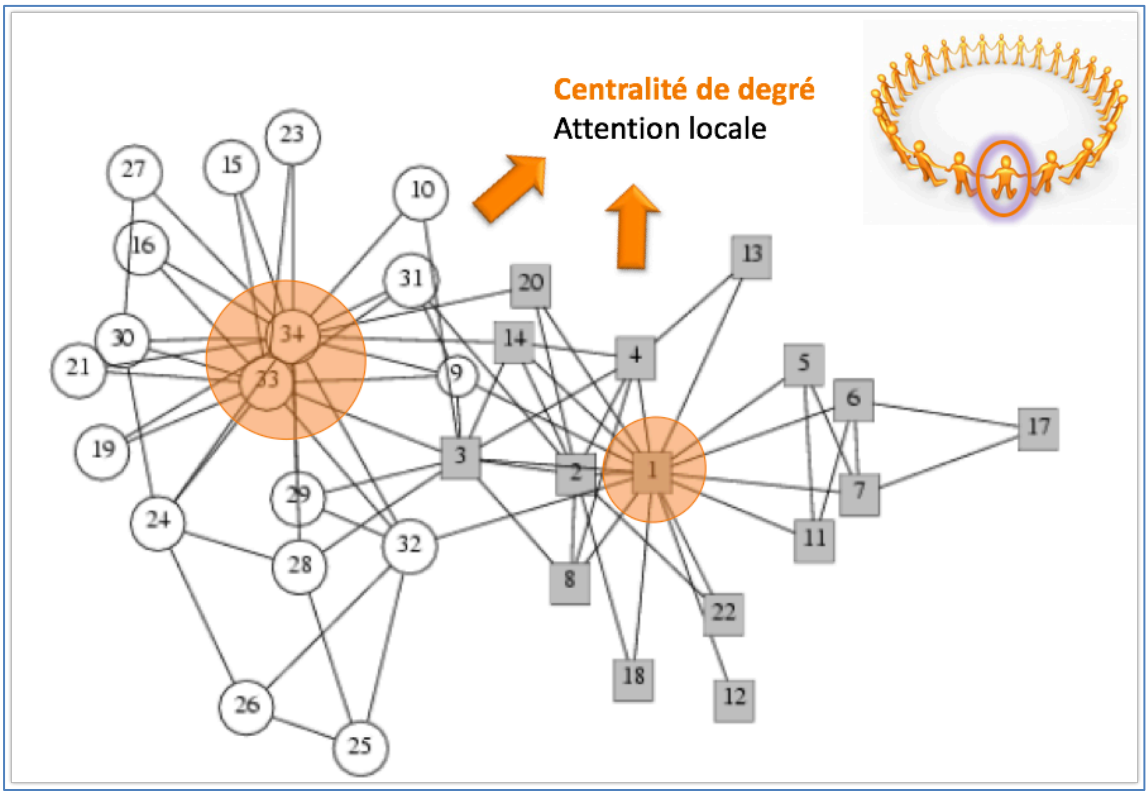


Figure 57. Quelques exemples d'indices typiques de l'analyse des réseaux sociaux. Ici la centralité d'un membre du réseau indique son importance locale.

Aujourd'hui, la plupart des implémentations de l'analyse des réseaux sociaux s'appuient directement sur la théorie des graphes ou sur des approches algébriques. Mika [18] a montré que les folksonomies peuvent être exploitées en utilisant la théorie des graphes afin d'identifier des groupes d'utilisateurs et l'émergence d'intérêts. Une approche par [19] utilise les profils FOAF afin d'identifier les communautés d'intérêt du réseau de LiveJournal.com. Goldbeck et al. [14] ont étudié la propagation de la confiance dans les réseaux sociaux en utilisant les frameworks du web sémantique. D'autres chercheurs [12] ont vérifié la loi de puissance des degrés et des structures communautaires dans les profils FOAF. Goldbeck et al. [15], encore, ont travaillé sur la fusion des profils FOAF et des identités utilisées sur différents sites. Par ailleurs, certains travaux comme ceux de Corby et al. [8], de Ereteo et al. [10] et [11], ont étendu des outils, comme par exemple le langage

[76] Semantic Web, W3C, http://www.w3.org/2001/sw/

de requête SPARQL, afin de trouver des chemins entre les ressources sémantiquement liées dans les graphiques basés sur RDF. Ces travaux nous ont servi de base pour travailler sur la représentation et l'analyse des réseaux sociaux basée sur des graphes et des ontologies.

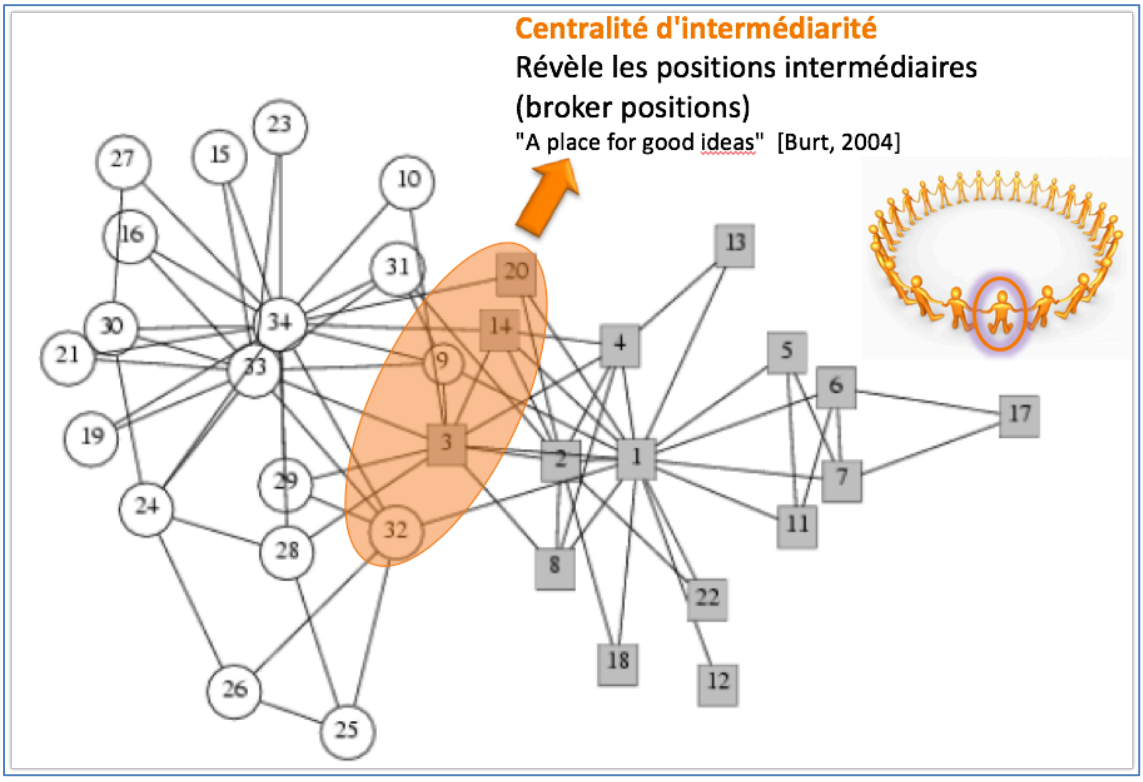


Figure 58. Quelques exemples d'indices typiques de l'analyse des réseaux sociaux. Ici la centralité d'intermédiarité d'un membre du réseau indique l'importance de son rôle de « passeur d'information » (broker).

De nombreux algorithmes sont disponibles pour détecter les structures sociales, les rôles et les positions. Mais un aspect de notre travail qui nous différencie des autres approches est d'aller au-delà de l'application des algorithmes classiques en travaillant sur des réseaux sociaux décrits avec les technologies du web sémantique. Nous avons ainsi proposé d'étendre ces algorithmes et d'en créer de nouveaux, afin de gérer les cycles de vie des communautés, en tenant compte non seulement de la structure du graphe social mais aussi de la sémantique des primitives ontologiques utilisées pour marquer ses nœuds et arcs.

Nous introduirons d'abord notre framework d'exploitation des modèles de graphes que sous-tendent les représentations RDF des réseaux sociaux. Nous fournirons des définitions formelles en SPARQL des opérateurs SNA paramétrés par les ontologies sous-jacentes à ces représentations. En outre, nous détaillerons SemSNA, une ontologie des caractéristiques de la SNA, utilisée pour annoter les réseaux sociaux. Enfin, nous présenterons comment nous avons effectué une analyse sémantique d'un réseau social sur un ensemble de données anonymisées fournies par la société Ipernity.com (un réseau social de partage de photos et blogs) et les résultats que nous avons obtenus.

5.2 Analyse sémantique des réseaux sociaux (SNA)

Nous utilisons les graphes RDF pour représenter les réseaux sociaux, et nous typons ceux qui utilisent des ontologies existantes avec des ontologies de domaines spécifiques si nécessaire.

En 2009, la plupart des données des réseaux sociaux populaires sont accessibles via des APIs REST mais ne sont pas accessibles directement en RDF, et les représentations utilisées par des sociétés comme Facebook ne se basent pas sur des modèles formalisés RDFs ou équivalents.

Les données sont accessibles par le biais d'API (flickr, Facebook, etc.) et parfois uniquement par l'exploration et copie de pages Web, qui doivent être converties. Pour annoter ces représentations de réseaux sociaux avec des indices SNA, nous avons conçu SemSNA (Figure 60), une ontologie qui décrit les notions de SNA, par exemple, la centralité, le diamètre, etc.

Avec cette ontologie, nous pouvons (1) abstraire des constructions de réseaux sociaux à partir d'ontologies de domaine pour appliquer nos outils sur des schémas existants en leur faisant étendre nos primitives ; et nous pouvons (2) enrichir les données sociales avec de nouvelles annotations (voir Figure 59) par exemple pour représenter les indices SNA qui seront calculés.

Ces annotations nous permettent de gérer plus efficacement le cycle de vie d'une analyse, en calculant au préalable les indices SNA pertinents et en les mettant à jour progressivement lorsque le réseau change au fil du temps. En plus de SemSNA, nous proposons des définitions formelles SPARQL des opérateurs SNA gérant la sémantique des représentations. Les tests actuels utilisent le moteur de recherche sémantique Corese [7] qui supporte les puissantes extensions SPARQL particulièrement bien adaptées aux fonctionnalités SNA telles que les calculs de chemins dans le graphe [8].

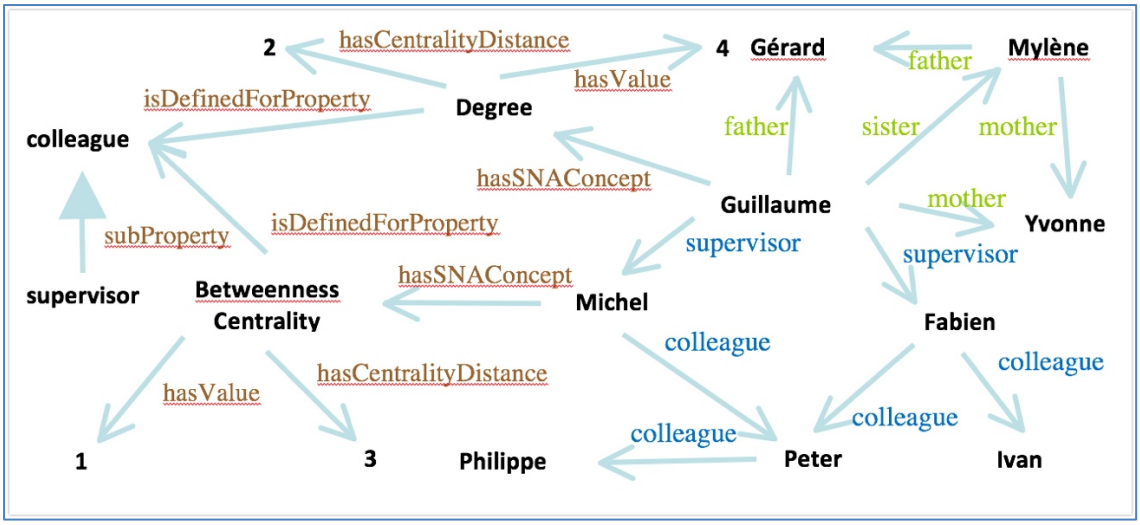


Figure 59. Un réseau social annoté avec des indices SemSNA (Degré, Intermédiarité).

En d'autres termes, on va associer au graphe RDF du réseau social des indicateurs utilisés par l'analyse des réseaux sociaux. Tel nœud représente une personne du réseau, on peut

calculer sa centralité (son importance dans le réseau, un paramètre lié au nombre de connexions du nœud avec d'autres membres) et la stocker sous la forme d'une annotation RDF. Si le réseau évolue (de nouvelles connexions se créent), la centralité pré-calculée pourra être mise à jour incrémentalement.

5.2.1 SemSNA : l'ontologie de l'analyse des réseaux sociaux.

Nous avons conçu SemSNA, une ontologie de l'analyse des réseaux sociaux. Une première version [11] était axée sur la position stratégique des acteurs, basée sur la définition de Freeman de la centralité [13]. Depuis lors, nous avons introduit de nombreuses nouvelles primitives pour annoter les données sociales avec différentes définitions de groupes et des indices utiles pour caractériser leurs propriétés (Figure 60).

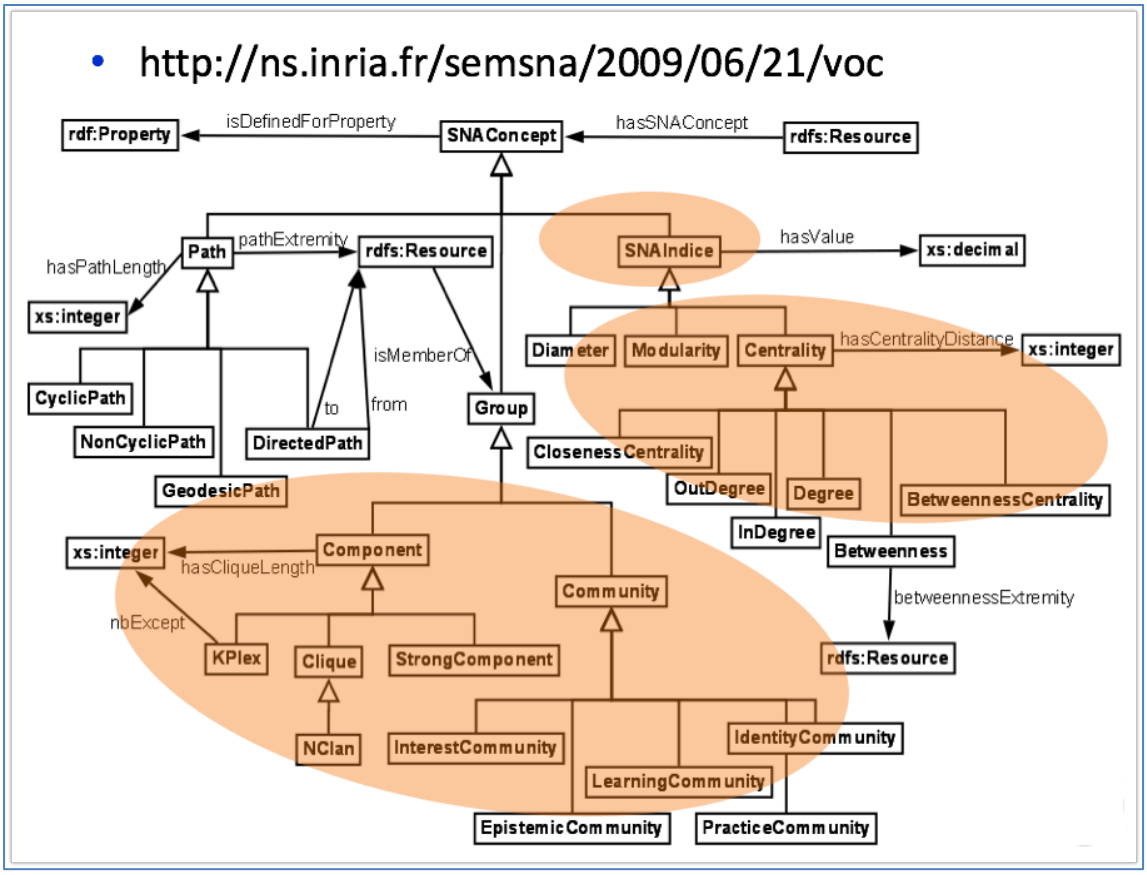


Figure 60. Schéma de SemSNA : l'ontologie de l'analyse des réseaux sociaux.

La classe principale `SNAConcept` est utilisée comme super classe pour tous les concepts SNA. La propriété `isDefinedForProperty` indique pour quelle relation, une instance du concept SNA est définie. Un concept de SNA est rattaché à une ressource sociale avec la propriété `hasSNAConcept`. La classe `SNAIndice` décrit des concepts tels que la centralité, et la valeur associée est définie avec la propriété `hasValue`.

Nous avons créé une super classe `Centrality` pour toutes les centralités définies par les classes `Degree`, `InDegree`, `OutDegree`, `Betweenness`, `BetweennessCentrality` et `ClosenessCentrality`. La propriété

`hasCentralityDistance` définit la distance du voisin prise en compte pour mesurer la centralité.

Nous proposons un ensemble de primitives pour définir et annoter des groupes de ressources liées par des propriétés particulières. La classe `Group` est une super classe pour toutes les classes représentant un concept alternatif de groupe de ressources. La classe `Component` représente un ensemble de ressources connectées. La classe `StrongComponent` définit un composant d'un graphe dirigé où les chemins reliant ses ressources ne contiennent aucun changement de direction.

La classe `Diameter`, sous-classe d'`Indice`, définit la longueur des géodésiques les plus longues (chemins les plus courts entre les ressources) d'un composant. La propriété `maximumDistance` nous permet de limiter l'appartenance d'un composant à une longueur de chemin maximum entre les membres du réseau. Une clique est un sous-graphe complet, pour une propriété donnée de notre modèle. Une n-clique étend cette définition avec une longueur maximale de chemin (n) entre les membres de la clique ; la classe `Clique` représente cette définition, et la longueur maximale du chemin est fixée par la propriété `maximumDistance`. Les ressources d'une clique peuvent être reliées par les chemins les plus courts passant par des membres ne faisant pas forcément partie de la clique.

Un `NClan` est une restriction d'une clique qui exclut ce cas particulier. Un `KPlex` relâche la définition de la clique pour permettre de se connecter à k membres avec un chemin plus long que la distance de la clique, k est déterminé par la propriété `nbExcept`.

Enfin, le concept `Communauty` regroupe différentes définitions de communautés : `InterestCommunity`, `LearningCommunity`, `GoalOrientedCommunity`, `PracticeCommunity` et `EpistemicCommunity` [16], [5]. Ces classes de définition de communautés sont liées à des ontologies plus détaillées comme [23] utilisées pour représenter les communautés de pratique.

5.2.2 Extraire des concepts de SNA avec SPARQL

Dans [21], les auteurs ont montré que SPARQL « *est assez expressif pour faire toutes les transformations sensibles des réseaux* ». Cependant, ce travail montre aussi que SPARQL n'est pas assez expressif pour répondre aux exigences de la SNA pour l'interrogation métrique globale (calculs de densité, centralité d'intermédiarité, etc.) des réseaux sociaux. De telles requêtes globales sont principalement basées sur l'agrégation de résultats et sur le calcul des chemins dans le graphe qui sont absents de la définition standard de SPARQL. Le moteur de recherche Corese [7] fournit de telles caractéristiques avec le groupement des résultats, des fonction d'agrégation comme `sum()` ou `avg()` et la récupération de chemins de [8][11]. Nous présentons ici comment effectuer une analyse de réseau social en combinant les caractéristiques structurelles et sémantiques du réseau social avec des requêtes effectuées avec Corese. Par exemple, la Figure 61 illustre le calcul d'un degré paramétré où seules les relations familiales sont prises en compte (*father*, *mother*, *sister*, *brother*).

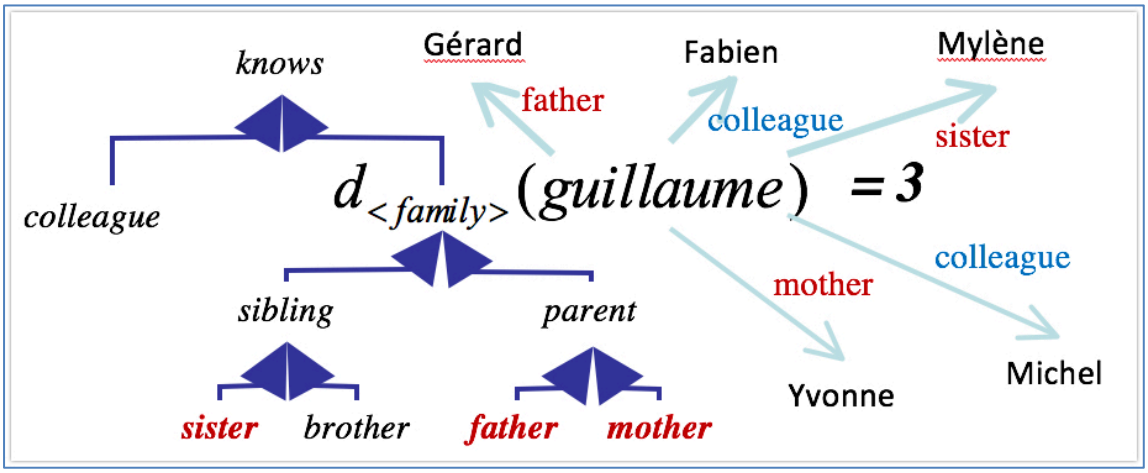


Figure 61. Fig. 3. Un degré paramétré qui tient compte d'une hiérarchie de relations. Guillaume est relié à Gérard, Fabien, Mylène, Muchel et Yvonne, mais son degré dans le réseau, paramétré par la relation « family » est de 3, alors que son degré « global » est 5. En typant les relations on peut affiner l'analyse.

Nous avons introduit une nouvelle convention syntaxique dans Corese pour l'extraction de chemins. Une expression régulière est maintenant utilisée à la place de la variable de propriété pour spécifier qu'un chemin est recherché et pour décrire ses caractéristiques. Les opérateurs d'expressions régulières sont : / (séquence), | (ou), * (0 ou plus), ? (optionel), ! (négation). Nous pouvons lier le chemin avec une variable spécifiée après l'expression régulière. L'exemple suivant récupère un chemin entre deux ressources :

```
?x foaf:knows*/rel:worksWith::$path ?y
    filter(pathLength($path) <= 3)
```

Les caractéristiques des chemins sont définies en ajoutant des options avant l'expression régulière : `'i'` pour les propriétés inverses, `'s'` pour récupérer un chemin le plus court, `'sa'` pour récupérer tous les chemins les plus courts, par exemple :

```
?x i sa foaf:knows*/worksWith ?y
```

La récupération des chemins nous permet d'exploiter la hiérarchie des relations, en prenant en compte les sous-propriétés à chaque étape. En conséquence, nous proposons des requêtes paramétrées qui acceptent comme argument une expression régulière de propriétés. La Figure 61 montre un degré paramétré qui ne tient compte que de la hiérarchie des relations familiales.

Nous proposons un ensemble de requêtes (Table 2) pour calculer les métriques de la SNA adaptées aux graphes orientés étiquetés décrits dans RDF (Table 1). Ces requêtes exploitent les fonctions de récupération des chemins ainsi que les fonctions de regroupement et d'agrégation dont nous venons de parler. Nous avons mis en place et testé tous les opérateurs que nous présentons.

Table 1. Définition des notions de SNA dans les graphes orientés étiquetés et notations utilisées.

Indices de la SNA et définition	**Notation**
Graphe : défini comme dans [1] and [11]	$G=(E_G, R_G, n_G, l_G)$ où : 1. E_G et R_G sont deux ensembles finis, respectivement de nœuds et de relations, 2. $n_G : R_G \rightarrow E_G^*$ associe à chaque relation un couple d'entités appelés les arguments de la relation. Si $n_G(r)=(e_1,e_2)$ on note $n_G^i(r)=e_i$ l' i^{ème} argument de r. 3. $l_G : E_G \cup R_G \rightarrow L$ est une fonction de labellisation des entités et des relations.
Nombre d'acteurs : le nombre d'acteurs d'un type donné.	$nb^{actor}_{<type>}(G)$
Nombre d'acteurs : le nombre d'acteurs impliqués dans une relation donnée en tant que sujet ou objet.	$nb^{actor}_{<rel>}(G)$
Nombre d'acteurs sujets : le nombre d'acteurs impliqués dans une relation donnée en tant que sujet.	$nb^{subject}_{<rel>}(G)$
N Nombre d'acteurs sujets : le nombre d'acteurs impliqués dans une relation donnée en tant qu'objet.	$nb^{object}_{<rel>}(G)$
Nombre de relations : nombre de paires de ressources liées par une *rel*.	$nb^{relation}_{<rel>}(G)$
Chemin : une liste de nœuds d'un graphe G, chacun relié au suivant par une relation appartenant à G.	$path\ p_{<rel>} =< x_0, x_1, x_2 ... x_n > \wedge \forall i < n \exists r \in R_G ; n_G(r) = (x_i, x_{i+1}) \wedge l_G(r) \leq rel$
Longueur : le nombre de relations / liens / arcs impliqués dans un chemin	$length(p =< x_0, x_1, x_2 ... x_n >) = n$
Densité : proportion du nombre maximal de relations. Représente la cohésion du réseau social.	$Den_{<rel>}(G) = \frac{nb^{relation}_{<rel>}(G)}{nb^{actor}_{<domain(rel)>}(G) * nb^{actor}_{<range(rel)>}(G)}$
Composant : un sous graphe connecté pour une propriété donnée (et ses sous-propriétés) sans lien avec des ressources en dehors du composant.	$Comp_{<rel>}(G) = (G_1, G_2, ..., G_k)$ Où G_k est un sous-graphe de G tel que pour chaque paire de nœuds n_i, n_j de G_k il existe un chemin entre n_i et n_j dans G_k.
Degrés : nombre de chemins de propriété *rel* (ou sous-type) ayant y à une extrémité et avec une longueur inférieure ou égale à *dist*. Le degré illustre la popularité locale d'un acteur.	$D_{<rel,dist>}(y) = \left\lvert \left\{ x_n ; \exists path\ p_{<rel>} =< x_0, x_1, x_2 ... x_n > \wedge ((x_0 = y) \vee (x_n = y)) \wedge n \leq dist \right\} \right\rvert$

Degré entrant : nombre de chemins de propriété *rel* (ou sous-type) finissant par *y* et avec une longueur inférieure ou égale à *dist*. Le degré entrant illustre l'influence que reçoit un acteur.	$D^{in}_{<rel,dist>}(y) = \left\vert \left\{ x_n ; \exists path\ p_{<rel>} =< x_0, x_1, x_2 ... x_n > \right. \right.$ $\left. \left. \wedge (x_n = y) \wedge n \leq dist \right\} \right\vert$
Degré sortant : nombre de chemins de propriété *rel* (ou sous-type) commençant par *y* et avec une longueur inférieure ou égale à *dist*. Le degré entrant illustre l'influence d'un acteur.	$D^{out}_{<rel,dist>}(y) = \left\vert \left\{ x_n ; \exists path\ p_{<rel>} =< x_0, x_1, x_2 ... x_n > \right. \right.$ $\left. \left. \wedge (x_0 = y) \wedge n \leq dist \right\} \right\vert$
Géodésique entre *from* et *to* : une géodésique est un plus court chemin entre deux ressources, un chemin de communication optionnel.	$g_{<rel>}(from,to) =< from, x_1, x_2 ..., x_n, to >$ tel que: $\forall p_{<rel>} =< from, y_1, y_2 ..., y_m, to >_{n \leq m}$
Diamètre : la longueur de la plus grande géodésique du réseau.	$Diam_{<rel>}(G)$
Nombre de géodésiques entre *from* et *to* : le nombre de géodésiques entre deux ressources montre la dépendance de leur connectivité avec leurs intermédiaires.	$nb^{g}_{<rel>}(from,to)$
Nombre de géodésiques entre *from* et *to* passant par le nœud *b* : le nombre de géodésiques entre deux ressources passant par un nœud intermédiaire ; il montre la dépendance de connectivité entre ces ressources, par rapport à *b*.	$nb^{g}_{<rel>}(b, from, to)$
Centralité de proximité : l'inverse de la somme de la distance géodésique entre chaque acteur du réseau social Représente la capacité d'un acteur d'être atteint ou d'atteindre un autre nœud du réseau : le degré auquel un individu est près de tous les autres individus d'un réseau social (directement ou pas).	$C^{c}_{<rel>}(k) = \left[\sum_{x \in E_G} length(g_{<rel>}(k,x)) \right]^{-1}$
Intermédiarité partielle : la probabilité pour *k* d'être situé sur un plus court chemin de propriété *rel* (ou sous-type) entre *x* et *y*. Représente la capacité de *k* d'être un intermédiaire entre *x* et *y* pour ce type (ou sous-type) de propriété.	$B_{<rel}(b,x,y) = \frac{nb^{g}_{<rel>}(b,x,y)}{nb^{g}_{<rel>}(x,y)}$
Centralité d'intermédiarité : la somme des intermédiarités partielles d'un nœud entre chaque autre paire de nœuds. Représente la capacité d'un nœud (d'un acteur) à être un intermédiaire dans l'ensemble du réseau social.	$C^{b}_{<rel>}(b) = \sum_{x,y \in E_G} B_{<rel>}(b,x,y)$

Table 2. Définition formelle dans SPARQL des indices SNA sémantiquement paramétrés.

SNA indices	Définition formelle en SPARQL

$nb^{actor}_{<type>}(G)$	`select merge`[77] `count(?x) as ?nbactor from <G>`
	`where{`
	`  ?x rdf:type param[type]`[78]
	`}`

$nb^{actor}_{<rel>}(G)$	`select merge count(?x) as ?nbactors from <G>`
	`where{`
	`  {?x param[rel] ?y}`
	`  UNION{?y param[rel] ?x}`
	`}`

$nb^{subject}_{<rel>}(G)$	`select merge count(?x) as ?nbsubj from <G>`
	`where{`
	`  ?x param[rel] ?y`
	`}`

$nb^{object}_{<rel>}(G)$	`select merge count(?y) as ?nbobj from <G>`
	`where{`
	`  ?x param[rel] ?y`
	`}`

$nb^{relation}_{<rel>}(G)$	`select cardinality(?p) as ?card from <G>`
	`where {`
	`{ ?p rdf:type rdf:Property`
	`  filter(?p ^ param[rel]) }`
	`UNION`
	`{ ?p rdfs:subPropertyOf ?parent`
	` filter(?parent ^ param[rel]) }`
	`}`

$Comp_{<rel>}(G)$	`select ?x ?y from <G> where {`
	` ?x param[rel] ?y`
	`}group by any`[79]

$D_{<rel,dist>}(y)$	`select ?y count(?x) as ?degree where {`
	`{?x (param[rel])*::$path ?y`
	`filter(pathLength($path) <=  param[dist])}`
	`UNION`
	`{?y param[rel]::$path ?x`
	`filter(pathLength($path) <=  param[dist])}`
	`}group by ?y`

$D^{in}_{<rel,dist>}(y)$	`select ?y count(?x) as ?indegree where{`
	`?x (param[rel])*::$path ?y`
	`filter(pathLength($path)`[80] `<=  param[dist])`
	`}group by ?y`

$D^{out}_{<rel,dist>}(y)$	`select ?x count(?y) as ?outdegree where {`
	`?x (param[rel])*::$path ?y`
	`filter(pathLength($path) <=  param[dist])`
	`}group by ?x`

$g_{<rel>}(from,to)$	`select ?from ?to $path pathLength($path) as ?length where{`
	`?from sa (param[rel])*::$path ?to`
	`}group by ?from ?to`

[77] Le mot clé *merge* fusionne les résultats avec des valeurs distinctes pour chaque variable.
[78] Corese accepte une liste de paramètres lors de l'exécution des requêtes.
[79] Le mot clé *any* active le regroupement de résultats ayant la même valeur pour n'importe quelle variable résultat.
[80] La fonction *pathLength()* renvoie la longueur du chemin donné en paramètre.

$Diam_{<rel>}(G)$	`select pathLength($path) as ?length from <G> where { ?y s (param[rel])*::$path ?to }order by desc(?length) limit 1`
$nb^{g}_{<rel>}(from,to)$	`select ?from ?to count($path) as ?count where{ ?from sa (param[rel])*::$path ?to }group by ?from ?to`
$nb^{g}_{<rel>}(b,from,to)$	`select ?from ?to ?b count($path) as ?count where{ ?from sa (param[rel])*::$path ?to graph $path{?b param[rel] ?j} filter(?from != ?b) optional { ?from param[rel]::$p ?to } filter(!bound($p)) }group by ?from ?to ?b`
$C^{c}_{<rel>}(y)$	`select distinct ?y ?to pathLength($path) as ?length (1/sum(?length)) as ?centrality where{ ?y s (param[rel])*::$path ?to }group by ?y`
$B_{<rel>}(b,from,to)$	`select ?from ?to ?b (count($path)/count($path2)) as ?betweenness where{ ?from sa (param[rel])*::$path ?to graph $path{?b param[rel] ?j} filter(?from != ?b) optional { ?from param[rel]::$p ?to } filter(!bound($p)) ?from sa (param[rel])*::$path2 ?to }group by ?from ?to ?b`
$C^{b}_{<rel>}(b)$	Non réalisé en SPARQL nécessite une étape de post-processing pour le calcul des plus courts chemins.

5.3 Lier les données d'interaction en ligne au Web sémantique

Ipernity.com, le réseau social que nous avons analysé, offre aux utilisateurs plusieurs options pour construire leur réseau social et partager du contenu multimédia (Figure 62).

Chaque utilisateur peut partager des photos, des vidéos, des fichiers musicaux, créer un blog, une page de profil personnel et commenter les ressources partagées. Chaque ressource peut être tagguée et partagée. Pour construire le réseau social, les utilisateurs peuvent spécifier le type de relation qu'ils ont avec les autres (leurs « amis ») : ami, famille, ou simple contact.

Les relations ne sont pas symétriques, Fabien peut déclarer une relation avec Michel mais Michel peut déclarer un autre type de relation avec Fabien ou ne pas l'avoir du tout dans sa liste de contacts ; nous avons donc un graphe dirigé étiqueté.

Les utilisateurs ont une page d'accueil contenant les informations de leur profil et des pointeurs vers les ressources qu'ils partagent. Les utilisateurs peuvent poster sur leur profil et les profils de leurs contacts en fonction des droits d'accès.

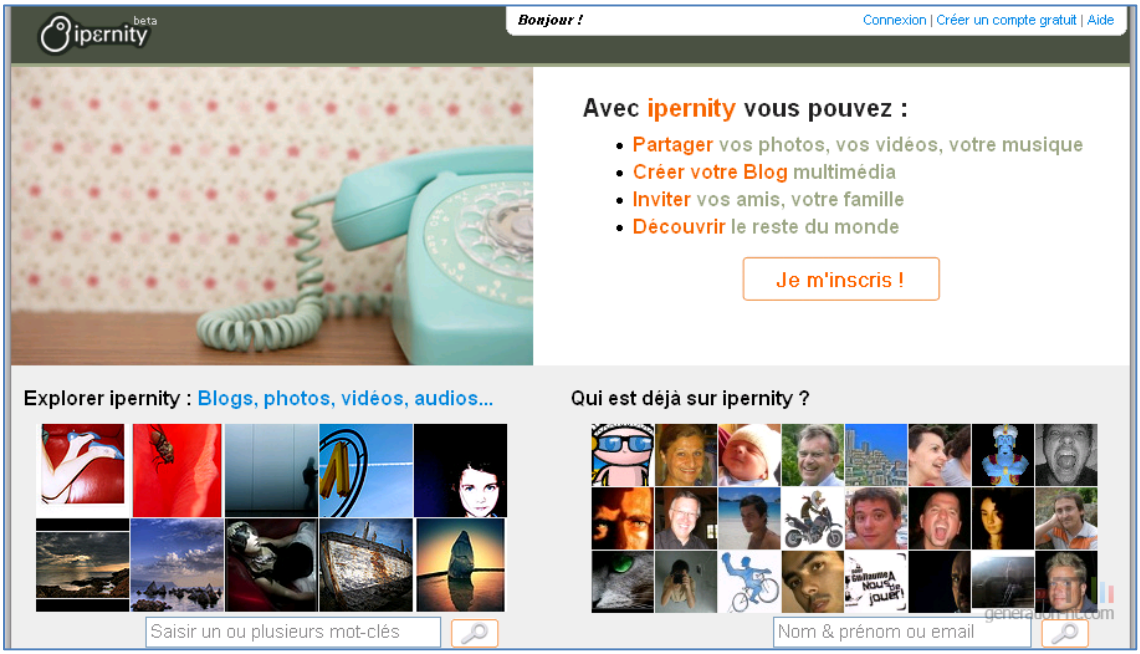


Figure 62. Ipernity.com un réseau social de partage de photos.

Toutes ces ressources peuvent être étiquetées, y compris la page d'accueil. Un éditeur peut configurer l'accès à une ressource pour la rendre publique, privée ou accessible uniquement pour un sous-ensemble de ses contacts, en fonction du type de relation (famille, ami ou simple contact), et peut surveiller qui l'a visitée. Les groupes peuvent également être créés avec des sujets de discussion avec trois types de visibilité : public (tous les utilisateurs peuvent le voir et s'y joindre), protégé (visible pour tous les utilisateurs, invitation requise pour s'y joindre) ou privé (invitation requise pour s'y joindre et consulter).

5.3.1 SemSNI : Extension de SIOC pour modéliser les interactions des réseaux sociaux.

Plusieurs ontologies existent déjà pour représenter les réseaux sociaux en ligne [10][11], et nous les utilisons comme base de notre modèle. Nous utilisons FOAF[81] pour représenter les profils des personnes et leurs relations en combinaison avec l'ontologie RELATIONSHIP[82] qui étend la propriété `foaf:knows` avec des sous-propriétés. RELATIONSHIP propose de nombreuses propriétés familiales (`parentOf`, `siblingOf`, etc.) mais n'a pas la super propriété dont nous avons besoin pour toutes ces relations : la famille.

Ainsi, nous avons étendu l'ontologie RELATIONSHIP avec la propriété `SemSNI:family` en tant que super propriété pour toutes ces relations familiales. Nous avons modélisé la relation « favori » et la relation « ami » respectivement avec la propriété `knowsInPassing` et `friendOf` de cette ontologie.

[81] http://www.foaf-project.org/
[82] http://vocab.org/relationship/

L'ontologie SIOC[83] [3] fournit la base pour définir un utilisateur (classe `User`), le contenu qu'il produit (classe `Item`, propriété `has_creator`) et les actions des autres utilisateurs sur ce contenu (propriété `has_reply`). L'ontologie SIOC types[84] étend `sioc:Item` pour spécifier les différents types de ressources produites en ligne. Nous utilisons `sioc:Post`, `sioct:ImageGallery`, `sioct:Weblog` et `sioct:Comment` pour modéliser respectivement les messages sur les pages d'accueil, les albums photos, les blogs et les commentaires sur les ressources. Nous utilisons SCOT[85] pour modéliser le tagging social.

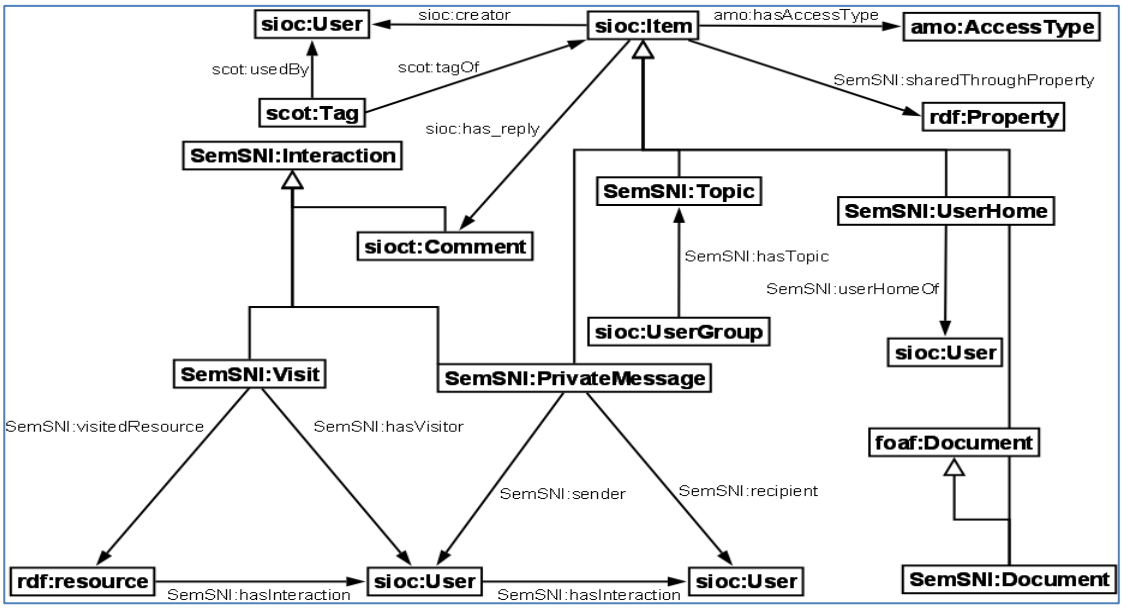


Figure 63. Schéma de SemSNI ; une ontologie des interactions au sein des réseaux sociaux.

Afin de modéliser les pages d'accueil, les messages privés, les sujets de discussion et les documents qui n'existent pas dans les types SIOC avec la sémantique requise, nous avons conçu l'ontologie SemSNI (Semantic Social Network Interactions, illustrée par la Figure 63).

SemSNI définit la classe `UserHome`, `PrivateMessage`, `Topic` et `Document` comme sous-classes de `sioc:Item` (`SemSNI:Document` étend également `foaf:Document`). La visite de classe et les propriétés `visitedResource` et `hasVisitor` nous permettent de décrire les visites d'un utilisateur à une ressource. Afin d'inférer de nouvelles relations entre les utilisateurs à partir de leurs interactions et du contenu qu'ils partagent, SemSNI définit la classe `Interaction` et la propriété `hasInteraction` (domain : `sioc:User`, range : `sioc:User`). Les classes représentant les échanges sur un contenu (`sioct:Comment`, `SemSNI:Visit` et `SemSNI:PrivateMessage`) sont définies comme des sous-classes de

[83] http://sioc-project.org/
[84] http://rdfs.org/sioc/types
[85] http://scot-project.org/

`SemSNI:Interaction` et nous pouvons déduire une relation `hasInteraction` entre le créateur de la ressource et son consommateur.

Nous n'avons pas typé plus précisément de telles relations, mais nous pouvons étendre cette propriété afin d'augmenter la granularité dans la description des interactions entre utilisateurs.

Nous utilisons les types d'accès (`Public`, `Private`, `Protected`) définis dans l'ontologie AMO [156], [157] que nous avons également développée dans le cadre du projet ISICIL en combinaison avec la propriété `SemSNI:sharedThroughProperty` pour modéliser le type de partage des ressources.

5.3.2 Génération de RDF à partir d'une base de données relationnelle avec Corese

Nous avons utilisé une autre fonctionnalité de Corese pour transformer la base de données relationnelle du réseau social d'Ipernity.com en RDF/XML. En effet, Corese a une extension qui nous permet d'imbriquer des requêtes SQL dans des requêtes SPARQL.

Ceci est fait au moyen de la fonction `sql()` qui retourne une séquence de résultats pour chaque variable dans un `SELECT` SQL. Corese possède une extension à la clause `SELECT` standard de SPARQL qui nous permet de lier ces résultats à une liste de variables [9]. Dans l'exemple suivant, nous montrons comment nous récupérons les relations « ami » de la base de données relationnelle, en utilisant cette fonction `sql()` ainsi qu'une autre intitulée `genIdUrl()` qui génère des URIs à partir des clés primaires (ids) de la base de données relationnelle :

Utilisation de requêtes SQL dans une requête SPARQL

```
construct { ?url_user1 rel:friendOf ?url_user2 }
select sql('jdbc:mysql://localhost:3306/mysql',
    'com.mysql.jdbc.Driver', 'user', 'pwd',
'SELECT user1_id, user2_id from relations where rel = 2 limit
    100000' ) as (?id1, ?id2)
fun:genIdUrl(?id1, 'http://semsni.fr/people/') as ?url_user1
fun:genIdUrl(?id2, 'http://semsni.fr/ people/') as ?url_user2
where {  }
```

Comme démontré dans [21], une fois exporté en RDF, on peut exploiter la représentation du réseau social avec une requête SPARQL en utilisant un bloc `CONSTRUCT`.

Corese peut également automatiser certaines transformations avec des règles d'inférence [6]. Par exemple, nous pouvons déduire une propriété `SemSNI:hasInteraction` entre deux acteurs lorsque l'un commente une ressource d'un l'autre en utilisant la règle suivante (ce travail date d'avant l'introduction de RIF[86] - Rule Interchange Format - comme standard) :

[86] https://www.w3.org/TR/rif-core/

Exemple de règle d'inférence

```
<cos:if>
{
      ?doc sioc:has_creator ?person1
      ?doc sioc:has_reply ?comment
      ?comment sioc:has_creator ?person2 `
}
</cos:if>
<cos:then>
  {
      ?person1 semsni:hasInteraction ?person2
}
</cos:then>
```

5.4 Résultats

En 2009, nous avons testé nos algorithmes et requêtes sur une configuration matérielle biprocesseurs quadri-cœurs Intel(R) Xeon(R) CPU X5482 3.19GHZ, avec 32.0Gb de RAM. Nous avons appliqué les requêtes définies sur les relations et interactions d'Ipernity.com. Nous avons analysé les trois types de relations séparément (*favorite*, *friend*,*family*) et nous avons également utilisé des requêtes polymorphes pour les analyser dans leur ensemble en utilisant leur super propriété : `foaf:knows`. Nous avons également analysé les interactions produites par les échanges de messages privés entre utilisateurs, ainsi que celles produites par quelqu'un qui commente les documents de quelqu'un d'autre.

Nous avons d'abord appliqué des métriques quantitatives pour obtenir des informations pertinentes sur la structure du réseau et sur les activités : le nombre de liens et d'acteurs, les composants et les diamètres.

Les résultats de l'analyse montrent que 61 937 acteurs sont impliqués dans un total de 494 510 relations. Ces relations sont décomposées en 18 771 relations *family* entre 8 047 acteurs, 136 311 relations *friend* impliquant 17 441 acteurs et 339 428 relations *favorite* pour 61 425 acteurs.

Ces premières mesures montrent que la sémantique des relations est globalement respectée, car les relations familiales sont moins utilisées que les relations amicales et favorites. 7 627 acteurs ont interagi à travers 2 874 170 commentaires et 22 500 ont communiqué à travers 795 949 messages.

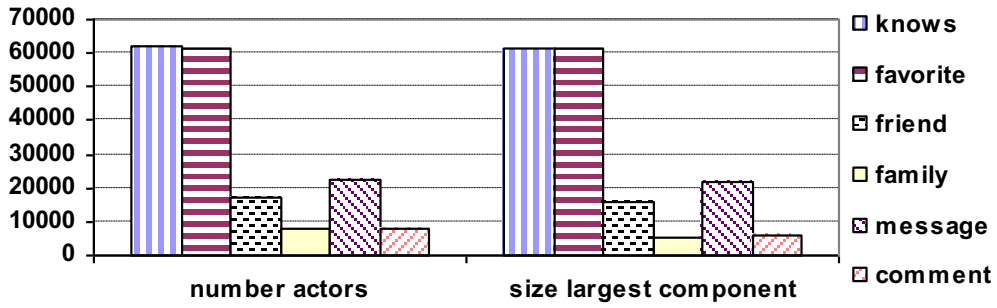


Figure 64. Nombre d'acteurs et taille du plus large composant des réseaux considérés

Tous ces réseaux possèdent un large composant (`Component`) contenant la plupart des acteurs (Figure 64) et de quelques très petits composants (moins de 100 acteurs) qui montrent « l'efficacité du réseau social à faire son travail » [20], c'est-à-dire « à connecter les gens ». Les sous-réseaux d'interaction ont un très petit diamètre (3 pour les commentaires et 2 pour les messages) en raison de leur densité élevée. Le réseau *family* a un diamètre élevé (19), ce qui correspond à sa faible densité. Cependant les réseaux *friend* et *favorite* ont une faible densité et un faible diamètre révélant la présence d'acteurs hautement intermédiaires.

L'intermédiarité et les centralités de degrés confirment cette dernière hypothèse. Le réseau *favorite* est très centralisé, avec cinq acteurs ayant une centralité d'intermédiarité supérieure à 0, et une valeur nettement supérieure pour un acteur : 1 999 858 alors que les 4 autres ont une valeur comprise entre 2,5 et 35. Cette valeur très élevée est en fait attribuée à l'animateur officiel du réseau social qui a une relation privilégiée avec la plupart des acteurs du réseau, lui donnant le degré le plus élevé : 59,301. Cette information a été confirmée par d'Ipernity, qui a confirmé la présence de « Léa d'Ipernity », l'animatrice officielle, dont le travail consiste à se lier avec de nombreux nouveaux arrivants et à animer le réseau.

Dans le réseau *friend*, 1 126 acteurs ont une centralité d'intermédiarité allant de 0 à 96 104 formant une longue queue, dont 12 seulement ont une valeur supérieure à 10 000.

Ces acteurs ne comprennent pas l'animateur, ce qui montre que le réseau *friend* a été bien adopté par les utilisateurs (le réseau d'« amis » doit être fortement connecté pour un réseau social qui fonctionne bien).

Le réseau *family* compte 862 acteurs avec une centralité entre 0 et 162 881 dont 5 valeurs supérieures à 10 000. Un seul acteur est hautement intermédiaire dans les réseaux d'amis et de famille.

La centralisation de ces trois réseaux présente des différences significatives montrant que la sémantique des relations a un impact sur la structure du réseau social. Les centralités d'intermédiarité de toutes les relations, calculées en utilisant le polymorphisme dans les requêtes SPARQL avec la propriété `knows`, soulignent à la fois l'importance de l'animateur qui a encore une fois la centralité significativement la plus élevée et l'adoption du réseau par les utilisateurs avec 186 acteurs jouant un rôle important d'intermédiaires.

Les employés d'Ipernity.com ont validé ces interprétations des métriques que nous avons calculées, démontrant l'efficacité d'une analyse de réseau social qui exploite la structure sémantique des relations.

Quelques notes relatives à l'implémentation.

Le moteur Corese fonctionne *en mémoire principale* et une telle quantité de données est consommatrice de mémoire. Les 494 510 relations déclarées entre les 61 937 acteurs utilisent un espace de 4,9 Go.

Les annotations de tous les messages utilisent 14.7 Go et la représentation des documents avec leurs commentaires utilise 27.2 Go.

D'autre part, travailler en mémoire centrale nous permet de traiter le réseau plus rapidement. Le calcul des chemins (extension *paths* de SPARQL) prend également beaucoup de temps et de mémoire et certaines requêtes ont dû être limitées à un nombre maximum de projections dans le graphe lorsque trop de chemins pouvaient être récupérés. Cependant, dans ce cas, les approximations obtenues sont suffisantes pour obtenir des métriques pertinentes (tableau 3) en particulier pour le calcul des centralités, comme Brandes l'a expliqué dans [2].

En outre, nous pouvons limiter la distance des chemins que nous recherchons en utilisant d'autres métriques. Par exemple, nous limitons la profondeur des chemins pour qu'elle soit inférieure ou égale au diamètre des composants lors du calcul des chemins les plus courts.

Table 3. Mesures de performance des requêtes / calculs d'indices SNA

Indice	Relation	Temps pour exécuter la requête	Nb Projections dans le graphe
$Diam_{<rel>}(G)$	Knows	1 m 41.93 s	10,000,000
	Favorite	1 m 51.37 s	10,000,000
	Friend	1 m 42.62 s	10,000,000
	Family	1 m 54.28 s	10,000,000
	Comment	35.06 s	1,000,000
	Message	1 m 50.84 s	10,000,000
$nb_{actors<rel>}(G)$	Knows	1 m 9.62 s	989,020
	Favorite	2 m 35.29 s	678,856
	Friend	11.67 s	272,622
	Family	0.68 s	37,542
	Message	17.62 s	1,448,225
	Comment	8 m 27.25 s	7,922,136
$Comp_{<rel>}(G)$	Knows	0.71 s	494,510
	Favorite	0.64 s	339,428
	Friend	0.31 s	136,311
	Family	0.03 s	18,771
	Message	1.98 s	795,949
	Comment	9.67 s	2,874,170
$D_{<rel,1>}(y)$	Knows	20.59 s	989,020
	Favorite	18.73 s	678,856

	Friend	1.31 s	272,622
	Family	0.42 s	37,542
	Message	16.03 s	1,591,898
	Comment	28.98 s	5,748,340
Plus courts Chemins utilisés pour calculer $C_{b<rel>}(b)$	Knows	Path length <= 2: 2h 56m 34.13s	1,000,000
	Favorite	Path length <= 2: 5h 33m 18.43s	2,000,000
	Friend	Path length <= 2: 1m 12.18 s	1,000,000
	Family	Path length <= 2 : 27.23 s Path length <= 3 : 1m 10.71 s	1,000,000 1,000,000

5.5 Conclusion

Les énormes quantités de contenu et d'interactions produits par les utilisateurs des plateformes sociales doivent être correctement organisées et filtrées pour préserver les avantages de la collaboration en ligne.

Alors que la SNA fournit des mesures pertinentes pour comprendre la structure des activités en ligne, les technologies du Web sémantique nous permettent de représenter, de combiner et d'interroger des données sociales à partir d'applications couvrant à la fois le web public ou les intranets.

La structure de graphe dirigée et étiquetée de RDF est bien adaptée pour représenter ces connaissances produites socialement. Notre framework permet d'analyser ces riches représentations typées des réseaux sociaux et de gérer la diversité des interactions et des relations avec des métriques SNA paramétrées.

La SNA classique ignore la sémantique des graphes richement typés et l'état de l'art en 2009 montrait que les modèles proposés s'appuiyaient presque toujours sur des bases de données relationnelles et n'offraient pas de mécanismes simples pour traiter la sémantique des réseaux typés.

Avec SPARQL et un moteur sémantique comme CORESE, les relations de subsumption sont prises en compte nativement lors de l'interrogation du graphe RDF. Les opérateurs paramétrés formellement définis dans SPARQL nous permettent d'ajuster la granularité de l'analyse des relations. De plus, des prétraitements peuvent être utilisés, par exemple en crawlant le réseau pour ajouter des types ou des relations chaque fois qu'un nouveau comportement typique est détecté, par exemple, un acteur qui commente fréquemment les messages d'un autre acteur peut être lié à lui par une relation « monitore » ou « surveille de près».

Les nouvelles requêtes associées à ces nouveaux opérateurs peuvent être définies à tout moment et SemSNA peut être étendu. L'*assortativité* du réseau (« *network assortativity"* tel que proposé par Newman dans [20]) est un exemple d'opérateur futur qui pourraient tirer parti à la fois de la sémantique des schémas (par exemple, la similarité entre deux nœuds) et des mécanismes d'extension de SPARQL (par exemple, compter le nombre de connexions partagées).

En outre, l'utilisation d'un schéma pour ajouter les résultats de nos requêtes (ou règles) au réseau nous permet également de décomposer les traitements complexes en deux étapes ou plus et de factoriser certains calculs entre différents opérateurs, par exemple, nous pouvons augmenter le réseau (enrichir le graphe RDF) avec des calculs de degrés entrants ou de centralité d'intermédiarité, puis exécuter ensuite une requête sur ces deux critères pour identifier les nœuds avec un degré > y et une centralité d'intermédiarité > x.

De plus, nous avons validé ce framework sur un véritable réseau social et révélé l'importance de prendre en compte la diversité des relations ainsi que leurs liens sémantiques.

Les sous-réseaux que nous avons analysés présentent des caractéristiques différentes qui mettent en évidence les acteurs stratégiques et le fractionnement des différentes activités.

L'approche est implémentée avec CORESE sur un triple store RDF de grande taille (CORESE est un logiciel gratuit qui gère des millions de nœuds, mais d'autres moteurs avec les mêmes extensions pourraient tout aussi bien être utilisés).

Par conséquent, nous annotons les données sociales avec les résultats de ces métriques SNA paramétrées en utilisant l'ontologie SemSNA pour fournir des services basés sur cette analyse (par exemple, filtrer les notifications d'activités sociales), pour les utiliser dans le calcul d'indices plus complexes ou (à l'avenir) pour soutenir des approches itératives ou parallèles dans les calculs.

Les temps de calcul sont toutefois importants et même si CORESE fonctionne en mémoire centrale, les expériences montrent que la manipulation d'un réseau avec des millions d'acteurs était hors de portée en 2009 sur les machines dont nous disposions. Nous avons commencé par la suite à étudier différentes approches pour résoudre ce problème : (1) identifier les techniques de calcul incrémentales, parallélisables, etc. (2) identifier les approximations qui peuvent être utilisées et dans quelles conditions elles fournissent des résultats de bonne qualité (3) identifier les caractéristiques des graphes (petits mondes, diamètres, etc.) qui peuvent nous aider à réduire l'espace et le temps de calcul pour les différents opérateurs.

Nous verrons dans le chapitre 7 que nous avons également exploité le travail présenté ici, notamment le modèle du graphe social et les ontologies associées semSNA et semSNI, conjointement au travail de la thèse de Freddy Limpens sur les folksonomies, pour identifier des communautés dans les réseaux sociaux.

6. Enrichissement sémantique collaboratif et assisté de folksonomies (social tagging, 2008-2011)

Contexte

Le "social tagging" ou "marquage social" est une pratique de classification par "mots clés" qui consiste à impliquer les utilisateurs dans le cycle de vie du contenu qu'ils échangent, lisent ou publient en ligne. On publie une vidéo sur YouTube, on va la « tagguer » en lui associant une série de mots clés qu'on appelle communément des « tags ».

Cette pratique est devenue populaire sur le Web en 2004 avec les premiers outils en ligne de classification de "bookmarks" tels que deli.cio.us ou de publications de photos et vidéos (Flicker, YouTube…), illustrée par la Figure 65.

L'ensemble des tags est généralement appelé une "folksonomie", il est classique dans l'analyse des réseaux sociaux de l'époque, de considérer le triumvirat *utilisateur-ressource-tag* pour structurer les folksonomies, notamment dans un graphe tripartite.

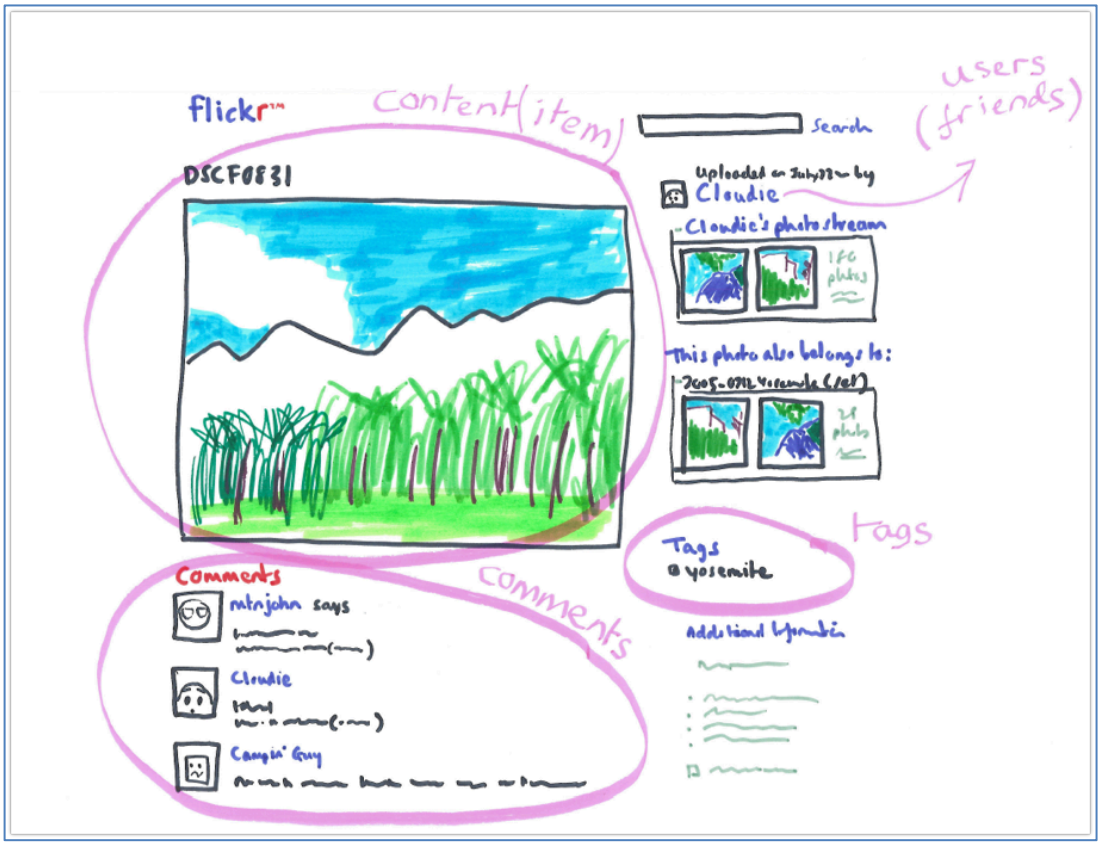


Figure 65. Social tagging typique réseau social de partage de photos.

Contributions

Dans le cadre de la thèse de Freddy Limpens nous avons conçu entre 2006 et 2010 un système basé sur le marquage/tagging qui intègre l'enrichissement sémantique collaboratif et assisté de la folksonomie de la communauté.

Nous avons proposé des modèles formels et des méthodes pour supporter des points de vue divergents concernant la sémantique des variables et les combiner efficacement dans une folksonomie cohérente et sémantiquement structurée, voir [28, 29 et 30] pour plus de détails. En effet, les folksonomies non contrôlées résultant du tagging social ont montré des limites (variations orthographiques, manque de relations sémantiques, etc.) qui entravent considérablement la navigation dans les corpus marqués.

Une façon de s'attaquer à ces limitations a été de structurer sémantiquement les folksonomies. Cela a aidé naviguer dans les corpus marqués en (1) enrichissant les résultats de recherche avec des variantes d'orthographe et des hyponymes, ou (2) en suggérant des balises pour étendre la recherche, ou (3) en organisant sémantiquement des balises pour guider les utilisateurs novices dans un domaine donné plus efficacement qu'avec des listes plates de balises ou de nuages de balises basés sur l'occurrence, ou (4) en aidant à la désambiguïsation.

Collaboration, encadrement

J'ai co-encadré la thèse de Freddy Limpens ("*Multi-points of View Semantic Enrichment of Folksonomies")* avec Fabien Gandon (encadrement 50%). J'étais responsable scientifique du projet ANR ISICIL dans lequel ces travaux se sont intégrés. J'ai supervisé le travail d'un ingénieur et coordonné les contributions des chercheurs de l'équipe KEWI dans ce projet, notamment en intégrant au wiki sémantique d'ISILC des outils pour exploiter les modèles de folksonomies présentés dans ce chapitre.

Publications principales ayant résulté de ces travaux

Guillaume Erétéo, Michel Buffa, Olivier Corby, Fabien Gandon, Mylène Leitzelman, Freddy Limpens, and Peter Sander. Handbook of Research on Methods and Techniques for Studying Virtual Communities, chapter *Semantic Social Network Analysis, a Concrete Case*. 2011, IGI Global

Pavel Arapov, Michel Buffa, Sébastien Comos, Olivier Corby, Nicolas Delaforge, Catherine Faron-Zucker, Fabien Gandon, Guillaume Husson and Freddy Limpens, *ISICIL : Intégration Sémantique d'Informations à travers des Communautés d'Intelligence en Ligne*, Démo paper, 11ème Conférence Internationale Francophone "Extraction et Gestion des Connaissances" EGC'10, Brest, France, 23-28 Janvier 2011.

Freddy Limpens, Fabien Gandon and Michel Buffa, *Helping Online Communities to Semantically Enrich Folksonomies,* Web Science Conference, April, 2010*,* Raleigh, NC, USA.

Freddy Limpens, Fabien Gandon, and Michel Buffa, *Collaborative Semantic Structuring of Folksonomies,* Conference IEEE Web Intelligence 2009*,* Milan September 15-18.

Fabien Gandon, Talel Abdessalem, Michel Buffa and Al., *ISICIL: Information Semantic Integration through Communities of Intelligence online,* 10th IFIP Working Conference on Virtual Enterprises, Thessaloniki, Greece, 7-9 October 2009.

Freddy Limpens, Fabien Gandon, and Michel Buffa. *Sémantique des folksonomies : structuration collaborative et assistée*. In Proc. Ingénierie des Connaissances IC'09, Hammamet, Tunisia, June 2009.

Guillaume Erétéo, Michel Buffa, Fabien Gandon, Mylène Leitzelman, and Freddy Limpens. *Leveraging Social Data with Semantics*, In W3C Workshop on the Future of Social Networking, Barcelona, Spain, 2009

Freddy Limpens, Fabien Gandon, Michel Buffa, *Rapprocher les ontologies et les folksonomies pour la gestion des connaissances partagées : un état de l'art*, Proc. 19èmes Journées Francophones d'Ingénierie des Connaissances, IC'08, June 18-20, Loria, Nancy, France.

Freddy Limpens, Fabien Gandon, and Michel Buffa, *Bridging Ontologies and Folksonomies to Leverage Knowledge Sharing on the Social Web: a Brief Survey*, in Proc. 1st International Workshop on Social Software Engineering and Applications (SoSEA), L'Aquila (IT), 15th September, 2008

Résumé de cette section

Un cycle de vie complet pour l'enrichissement sémantique des folksonomies

Dans cette section je présente une contribution qui a eu pour cadre la thèse de Freddy Limpens que j'ai co-encadrée à 50% avec Fabien Gandon, et qui a été intégrée au projet ANR ISICIL dont j'étais responsable dans l'équipe KEWI du laboratoire I3S.

Nous avons développé une approche pour l'enrichissement sémantique des folksonomies. Nous avons pour cela proposé un système sociotechnique dans lequel les utilisateurs sont assistés par des agents automatiques pour organiser leurs tags et maintenir leurs points de vue personnels tout en bénéficiant des contributions des autres. Une caractéristique notable est que les actions de chaque utilisateur sont collectées et aident les utilisateurs référents d'une communauté dans leur tâche de construction d'un point de vue consensuel.

Afin d'amorcer ce processus, et pour organiser initialement les tags de la folksonomie, nous avions développé un dispositif automatique, qui permettait d'extraire la sémantique émergente d'un ensemble de tags donnés.

À cet égard, l'article présenté dans [28] donne une évaluation des principales méthodes existantes et envisagées, basées sur des analyses de chaînes de caractères afin de: (a) motiver le choix des métriques les plus performantes et (b) évaluer la capacité de ces métriques à différencier les relations sémantiques typiquement utilisées dans un thésaurus, c'est-à-dire savoir (1) quand deux tags sont simplement liés, ou (2) lorsqu'un tag est plus général ou plus spécialisé qu'un autre, ou (3) lorsque deux tags sont des variantes orthographiques de la même notion.

En conséquence, nous avons proposé une métrique heuristique qui effectue cette tâche. Cette métrique heuristique était bien la plus performante pour détecter les variantes d'orthographe, comme prévu, mais a donné également des résultats intéressants pour la détection de relations de subsomption en découvrant automatiquement, par exemple, que « pollution » est plus général que « pollution du sol ».

Nous avons également montré quantitativement que des techniques analysant la structure globale des folksonomies sont nécessaires pour extraire certaines relations sémantiques, même si elles sont plus coûteuses et non incrémentales. Ceci est vrai en particulier,

lorsque des tags partageant les mêmes relations sémantiques ne sont pas morphologiquement similaires. Nous avons à l'époque obtenu des résultats expérimentaux montrant l'intérêt de notre approche, en calculant les trois types de relations entre tags précitées (tags simplement liés, plus généraux/spécialisés, variantes orthographiques), sur un ensemble de données du monde réel issues de l'ADEME.

Afin de capturer des points de vue divergents dans la structuration sémantique des folksonomies, nous avons également proposé une ontologie formelle qui utilise des graphes nommés pour décrire les relations sémantiques entre les tags.

Les points de vue des utilisateurs sont attachés aux relations identifiées. En outre, nous avons inclus les différentes classes d'agents qui proposent ou rejettent les relations identifiées (simple utilisateur, utilisateur référent, etc.). Par conséquent, nous avons été en mesure de modéliser le cycle de vie complet d'un enrichissement collaboratif et automatiquement assisté de folksonomies (Figure 66).

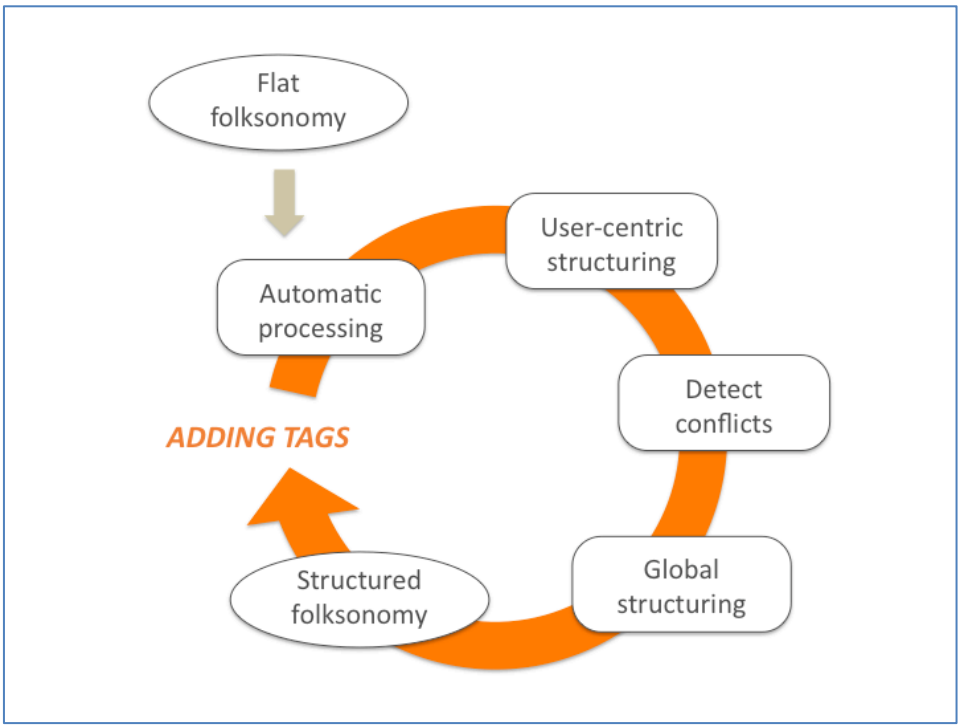


Figure 66. Cycle de vie d'un enrichissement collaboratif de folksonomies

Ce cycle commence par une folksonomie plate qui (1) est d'abord analysée par des agents automatiques proposant des relations sémantiques (tags simplement liés, plus généraux/spécialisés, variantes orthographiques). (2) Les utilisateurs peuvent alors contribuer et maintenir leur propre point de vue en validant, en rejetant ou en proposant les relations identifiées grâce à une interface conviviale intégrée dans un outil de navigation. (3) Les conflits émergeant de ces points de vue sont détectés et (4) utilisés pour aider un utilisateur référent à maintenir un point de vue global et consensuel. (5) Le résultat de ce processus est une folksonomie augmentée d'assertions sémantiques, chacune liée à différents points de vue coexistant avec un point de vue consensuel. (6)

Le cycle redémarre lorsque de nouveaux tags sont ajoutées ou lorsque des relations sont suggérées ou modifiées. Les assertions sémantiques sont utilisées pour suggérer des tags lors de la navigation dans la folksonomie et un ensemble de règles formelles permet de filtrer les assertions sémantiques afin de présenter une expérience cohérente aux utilisateurs tout en leur permettant de bénéficier des contributions des autres.

Notre approche a été testée à l'agence ADEME pour améliorer la navigation dans son corpus documentaire en ligne, disponible aux membres de l'agence et au grand public (Figure 67). Dans ce contexte, les ingénieurs experts de ADEME ont pu maintenir leurs points de vue afin de maintenir leur expertise dans un domaine donné. En même temps, les archivistes (nos utilisateurs référents) sont aidés dans l'enrichissement de la folksonomie par la suggestion de nouveaux tags et structurent sémantiquement leur point de vue global en profitant de l'enrichissement collaboratif de la folksonomie.

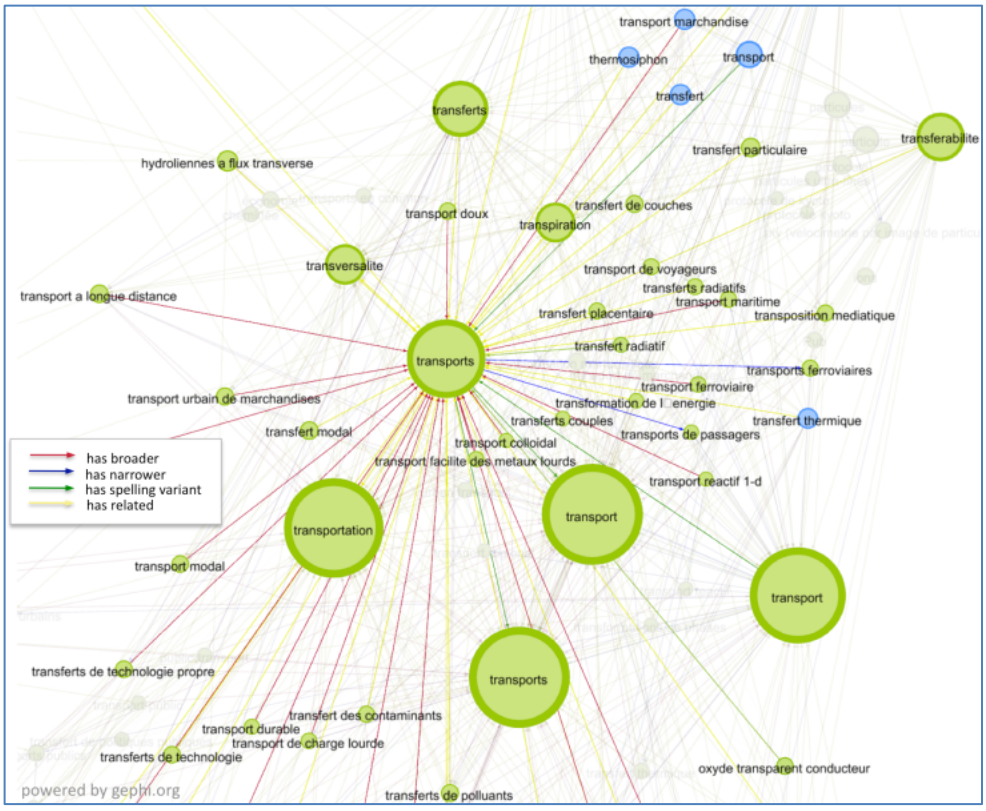


Figure 67. Exemple de résultats du traitement automatique des tags liés au mot clé "transports". La taille des nœuds indique le nombre d'arêtes entrant (en degrés).

Dans cette section nous donnerons le détail de la conception de ce système, des ontologies et opérateurs développés pour la manipuler, des évaluations des différentes métriques envisagées et détaillerons celles qui ont été choisies. Enfin, nous décrirons en détail les résultats expérimentaux obtenus.

6.1 Introduction

En 2009, avec les dernières évolutions des applications web (delicious, digg, Facebook, Google+, LinkedIn, etc.), des communautés d'intérêt émergent et construisent des

répertoires de références dans leurs domaines à une vitesse impressionnante et avec des réponses très agiles aux changements. L'une des principales forces des outils permettant à ces communautés de s'organiser est leur capacité à transformer des utilisateurs habituellement passifs en participants actifs, en producteurs. La diversité et la masse des utilisateurs sont utilisées pour s'attaquer à la diversité et à la masse des sources d'information. Pouvoir favoriser l'émergence de ces communautés et catalyser leur activité devient un atout décisif pour de nombreux acteurs du web.

Cependant, d'une part, les outils Web 2.0 présentent des limites lorsqu'il s'agit d'automatiser des tâches ou de contrôler des processus et, d'autre part, les systèmes d'information plus structurés souffrent souvent de problèmes de facilité d'utilisation et de capture des connaissances.

Le projet ANR ISICIL (**I**nformation **S**emantic **I**ntegration through **C**ommunities of **I**ntelligence on**L**ine) a été lancé en 2008 pour réconcilier les nouvelles applications web virales avec les représentations et les processus formels. Plus précisément, nous avons étudié et expérimenté l'utilisation de nouveaux outils d'aide à l'intelligence d'entreprise (Business Intelligence) et à la veille technologique. Ces outils s'appuient sur les interfaces avancées du Web 2.0 pour les interactions et sur les technologies du Web sémantique pour l'interopérabilité et le traitement de l'information.

La plate-forme ISICIL intègre toutes les approches détaillées dans le reste de chapitre dans une architecture web supportant le déploiement d'outils web sociaux sémantiques interconnectés sur un intranet pour soutenir les activités de veille, de recherche, de notification et de reporting. Cette plate-forme open source gratuite propose trois fonctionnalités : (1) un réseau social sémantique associé à des outils d'analyse de réseaux sociaux, (2) un système d'enrichissement sémantique de folksonomies liées à des terminologies d'entreprise et (3) des interfaces utilisateur sémantiquement augmentées, des outils de suivi d'activité et de reporting.

Dans une première partie, nous présentons notre approche de l'enrichissement sémantique semi-automatique des folksonomies.

Depuis 2009, nous avons développé un système socio-technique dans lequel les agents automatiques aident les utilisateurs à maintenir leur point de vue personnel tout en bénéficiant des contributions des autres, et aident également les utilisateurs référents (tels que les administrateurs système ou les gestionnaires de communauté) dans leur tâche de construction d'un point de vue consensuel. Notre approche est fondée sur une analyse minutieuse des usages de nos communautés cibles qui nous permet d'intégrer notre processus en tant qu'effet secondaire de leur activité quotidienne. Le résultat est une folksonomie sémantiquement enrichie qui soutient de multiples points de vue, dans lesquels les balises sont liées par des relations telles que "variante orthographique", "apparentée", "plus général que" ou "plus spécialisé que" qui sont ensuite utilisées pour améliorer la recherche, la navigation et la notification.

En parallèle, nous avons développé « une pile logicielle complète » pour l'analyse des réseaux sociaux sémantiques. Nous utilisons des graphes riches et des opérateurs

SPARQL pour étendre et paramétrer les indicateurs classiques de l'analyse des réseaux sociaux (SNA) tels que le diamètre du réseau, la centralité des acteurs, etc. C'est le travail qui a été présenté dans la section 5.

Sur la base de ces résultats, nous présenterons dans le chapitre suivant de ce manuscrit, un algorithme de détection de communautés, SemTagP qui offre non seulement de détecter mais aussi d'étiqueter les communautés en exploitant (en plus de la structure du réseau social) les relations sémantiques déduites entre les balises. Ce faisant, nous sommes en mesure d'affiner le partitionnement du réseau social par un traitement sémantique et d'étiqueter l'activité des communautés détectées.

Nous avons testé et évalué ces deux contributions sur le réseau social construit à partir de thèses de doctorat financées par l'ADEME. Nous avons montré comment cette approche nous permet de détecter et d'étiqueter les communautés d'intérêt, de contrôler le niveau d'abstraction dans les tags utilisés pour l'étiquetage et donc la granularité des communautés.

6.2 Un cycle de vie complet pour l'enrichissement sémantique des folksonomies.

Les tags librement fournis par les utilisateurs de services de marquage social ne sont pas explicitement liés sémantiquement, ce qui entrave considérablement les possibilités de navigation et d'exploration de ces données. D'autre part, les folksonomies offrent de grandes opportunités pour *bootstraper* la construction de thésaurus.

Dans cette section, nous détaillons une approche visant à enrichir sémantiquement les folksonomies en s'appuyant à la fois sur un traitement automatique mais aussi sur les actions de restructuration menées par les utilisateurs, tout en soutenant formellement des points de vue multiples (Figure 68

Nous prenons en compte la structure sociale de nos communautés cibles pour intégrer le processus d'enrichissement de la folksonomie dans les tâches quotidiennes.

Notre système permet aux utilisateurs individuels de naviguer plus efficacement au sein des folksonomies, mais aussi de maintenir leur propre structure de tags tout en bénéficiant des contributions des autres.

Notre approche apporte également des solutions au problème du goulot d'étranglement de l'acquisition des connaissances en aidant les communautés à construire des thésaurus intégrant les multiples contributions de tous leurs membres. Le résultat est une véritable solution socio-sémantique légère et non contraignante à l'enrichissement et à la construction de thésaurus.

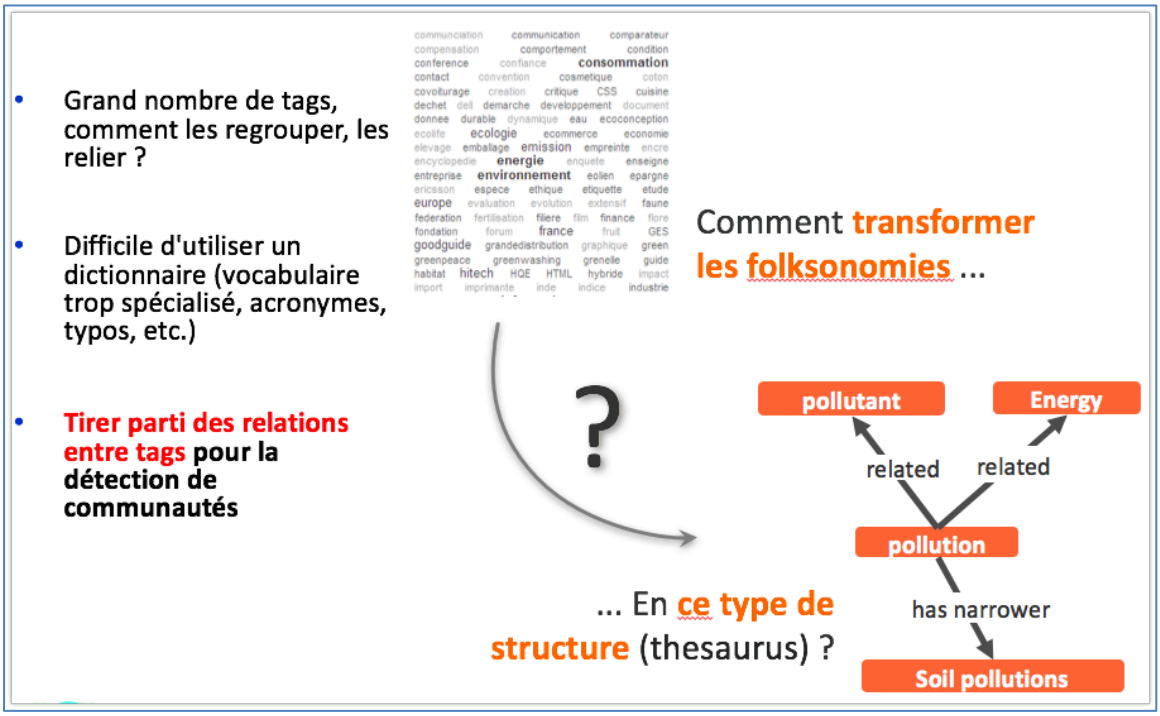


Figure 68. Enjeu : structurer les tags par enrichissement sémantique.

6.2.1 Contexte

Le marquage social (« social tagging ») est un moyen efficace d'impliquer les utilisateurs dans la gestion du contenu qu'ils échangent, lisent ou publient en ligne. Cependant, les folksonomies résultant de cette pratique ont montré des limites, en particulier lorsqu'on rencontre des variations orthographiques de balises similaires ou lorsque l'absence de relations sémantiques entre les balises entrave significativement la navigation dans les corpus balisés.

Une façon de s'attaquer à ces limitations est de structurer sémantiquement les folksonomies. Cela peut aider à naviguer dans les corpus balisés en (1) enrichissant les résultats de recherche basés sur les balises avec des variantes orthographiques et des hyponymes, ou (2) en suggérant des balises connexes pour étendre la recherche, ou (3) en organisant sémantiquement les balises pour guider les utilisateurs novices dans un domaine donné plus efficacement qu'avec des listes plates de balises ou des nuages d'occurrences, ou (4) en aidant à la désambiguïsation.

La conception de notre système d'étiquetage intègre l'enrichissement sémantique collaboratif et assisté de la folksonomie de la communauté. Nous proposons des modèles formels et des méthodes pour soutenir les points de vue divergents concernant la sémantique des tags et pour les combiner efficacement en une folksonomie cohérente et sémantiquement structurée.

Notre utilisateur final est l'agence ADEME qui cherche à élargir le public de ses travaux scientifiques dans le domaine du développement durable et de l'environnement. Dans ce scénario, on peut distinguer trois types d'acteurs : (1) les ingénieurs experts travaillant à ADEME qui sont des spécialistes d'un domaine donné, (2) les archivistes qui s'occupent

de l'indexation des documents de ADEME et qui ont une connaissance transversale de la thématique couverte à l'agence, et (3) le public qui a accès aux documents de ADEME à partir de son site web.

Les archivistes cherchent à la fois à enrichir leur base d'indexation, qui peut être considérée comme une folksonomie contrôlée, et à la faire évoluer vers une structure de type thésaurus. La difficulté vient ici des différents points de vue qui peuvent surgir de la communauté des ingénieurs experts (et peut-être aussi du public), et qui doivent être transformés en une structure cohérente par les archivistes.

Dans la section 6.2.2, nous présentons les travaux en cours sur l'enrichissement sémantique des folksonomies, et positionnons notre contribution. Dans la section 6.2.3, nous donnons une présentation générale de notre approche. Dans la section 6.2.4, nous présentons les résultats du traitement automatique des données des balises et détaillons notre méthode d'extraction de la sémantique émergente avec une combinaison de mesures de distances. La section 6.2.5 traite quant à elle de la saisie et de l'exploitation de la contribution des utilisateurs afin de fournir une folksonomie sémantiquement enrichie qui soutient de multiples points de vue.

6.2.2 Travaux connexes sur l'enrichissement de folksonomies

L'enrichissement de folksonomies a fait l'objet de nombreux travaux de recherche couvrant une grande variété d'approches.

6.2.2.1 Extraction de la sémantique émergente

Une première catégorie de travaux vise à extraire la sémantique émergente des tags des folksonomies en mesurant la similitude sémantique entre les tags. Les études de [131] et [132] proposent une analyse des différents types de mesures de similarité et des relations sémantiques qu'elles tendent à véhiculer.

L'approche la plus simple consiste à compter la cooccurrence des tags dans différents contextes (utilisateurs ou ressources). Cattuto et al [132] ont montré que ce type de mesure fournissait des relations de subsomption mais que les résultats obtenus n'étaient pas suffisamment précis.

Des méthodes plus élaborées exploitent la structure en réseau des folksonomies en utilisant l'hypothèse de distribution selon laquelle les mots utilisés dans des contextes similaires ont tendance à être sémantiquement liés. Pour appliquer cette hypothèse sur les tags, Cattuto et al. [132] ont calculé la mesure de similarité avec la méthode des cosinus dans les espaces vectoriels obtenus en pliant la structure tripartite de folksonomie/utilisateur/tags sur des agrégations couvrant les associations de tags avec soit : les autres tags (contexte tag-tag), soit les utilisateurs (contexte tag-user), soit les ressources (tag-ressources).

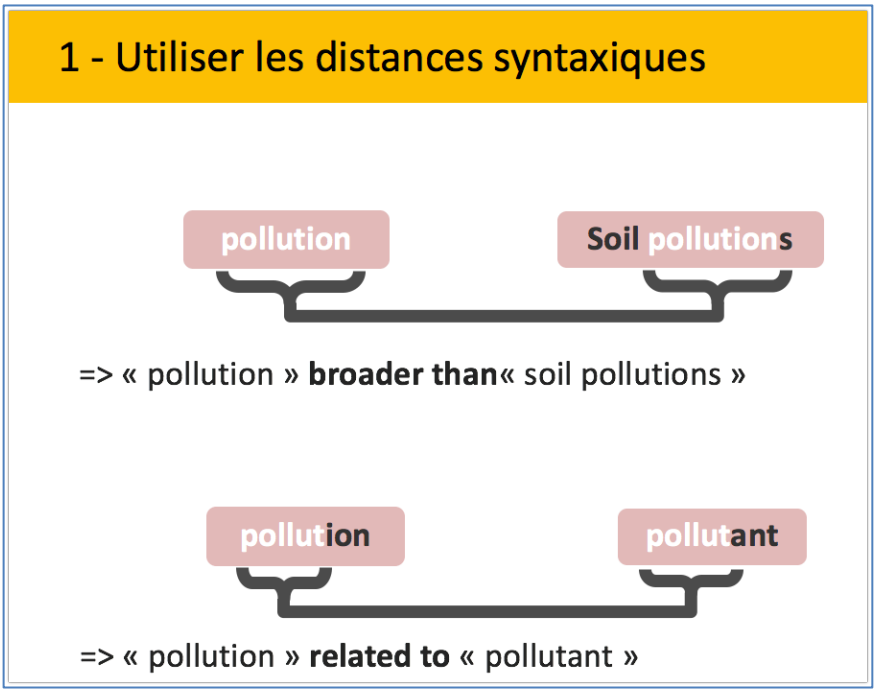


Figure 69. Utilisation de distances syntaxiques pour inférer des relations entre tags

Leur étude montre que le contexte tag-tag est le plus performant à un coût raisonnable et que la relation sémantique véhiculée par cette mesure est de type "*related*" (deux tags sont reliés entre eux, sans plus de précision sur la nature de la relation).

Mika [133] a également appliqué et évalué différents aspects de la structure tripartite des folksonomies. Il est intéressant de noter qu'il a montré, selon une évaluation qualitative, que l'exploitation des associations de tags basées sur l'utilisateur permettait d'obtenir des relations taxonomiques plus représentatives. Le principe de cette association est que si, par exemple, la communauté des utilisateurs du tag "agriculture biologique" est incluse dans la communauté des utilisateurs du tag "agriculture", alors le tag "agriculture" est plus général que le tag "agriculture biologique".

Heyman et al [134] ont proposé un algorithme qui construit une taxonomie à partir de tags en parcourant le graphe de similarité calculé à partir de la distance (méthode des cosinus) basée sur le contexte Tag-Resource. La hiérarchie des tags est construite à partir du tag ayant la centralité la plus élevée, et chaque tag, pris dans l'ordre de centralité, est ajouté soit comme un enfant de l'un des nœuds, soit comme un enfant du nœud racine en fonction d'une valeur seuil.

6.2.2.2 Modèles et outils pour structurer les tags

Un autre type d'approche consiste à laisser les utilisateurs structurer sémantiquement les tags par différents moyens. Tanasescu et Streibel ont proposé de *marquer les tags* [135], Huynh-Kim Bang et al. ont proposé une syntaxe simple pour spécifier les relations de subsumption (avec ">" ou "<") ou de synonymie (avec "=") entre les balises [136].

Certains outils disponibles en ligne disposent également de capacités de structuration sémantique telles que Gnizr[87] et Semanlink[88], et même Flickr.com avec ce qu'ils appellent des « machine tags », des tags destinés non pas aux utilisateurs mais au système d'indexation sous-jacent du site[89].

Dans la même tendance, la communauté Linked Data cherche à tisser ensemble le contenu des sites web sociaux grâce à un ensemble d'ontologies formelles ne visant pas à décrire les connaissances des communautés mais plutôt la structure de leurs plateformes d'échange de connaissances. Par exemple, SCOT[90] décrit les tags comme faisant partie de nuages de tags partageables, et SIOC[91] décrit le contenu des communautés en ligne.

MOAT [137] est une ontologie visant à lier chaque action de balisage avec une URI représentant *la signification de cette action de balisage*. Ces URIs peuvent se relier à des concepts formels d'ontologies ou à n'importe quelle page web contenant la description d'une notion. Une fois que les actions de tags sont formellement liées à des concepts, il est possible de désambiguïser les tags lors de la recherche, mais aussi d'exploiter les mécanismes d'inférence via les concepts formels et d'obtenir une expérience de navigation plus riche.

NiceTag est un modèle qui cherche à rendre compte des usages des tags à travers une modélisation plus fine des relations entre les tags et les ressources balisées/tagguées [138]. Sa flexibilité et l'utilisation de graphes nommés permettent à ce modèle de servir de pivot pour tous les autres modèles de balises.

Enfin, comme nous proposons de soutenir des points de vue divergents, rappelons brièvement quelques approches multipoints de vue comme [139] qui proposait des représentations de connaissances multi points de vue fondées sur des graphes conceptuels dans lesquels les liens entre les concepts peuvent être liés à un point de vue donné.

Bouquet et al. ne proposent pas exactement de représenter des concepts selon des points de vue multiples, mais suggèrent plutôt de contextualiser les ontologies grâce à C-OWL, une extension de OWL [140]. L'idée de C-OWL est de fournir un ensemble de primitives pour décrire les correspondances (*mappings*) entre des 'ontologies locales qui peuvent être chacune associée à un point de vue.

D'autres travaux cherchent à intégrer une ou plusieurs des approches précédentes. Par exemple, Angletou et al. [141] et Specia et al. [142] utilisent des mesures de similarité pour trouver des tags connexes, puis associent ces tags à des concepts à partir d'ontologies en ligne disponibles afin de structurer sémantiquement ces tags avec des propriétés formelles. Van Damme et al. ont proposé une approche intégrée de l'enrichissement de folksonomies incluant autant de ressources que possible, en utilisant chacune d'entre elles d'une manière adaptée en plus de la validation des inférences par les utilisateurs [143].

[87] http://code.google.com/p/gnizr/
[88] http://www.semanlink.net/
[89] http://www.flickr.com/groups/mtags/
[90] http://scot-project.org/
[91] http://sioc-project.org/

Enfin, notre approche peut être liée à la notion de « construction et maturation d''ontologies ». En effet, notre tentative de structurer les balises à la manière d'un thésaurus peut être liée aux méthodes de construction d'ontologies formelles à partir de textes [144] ou de bases de données [145]. Plus récemment, Braun et al [146] se sont penchés sur le problème de l'édition collaborative de l'ontologie et ont souligné les limites des outils actuels d'ingénierie ontologique à cet égard. Ils ont proposé d'intégrer la maturation de l'ontologie dans des tâches communes telles que la recherche d'information, et ils ont développé un service de bookmarking avec la possibilité pour tous les utilisateurs d'ajouter ou d'éditer de nouveaux tags "sémantiques" formellement structurées avec SKOS[92].

6.2.2.3 Discussions sur les approches actuelles

L'automatisation complète de l'enrichissement sémantique de folksonomies est difficile. Premièrement, les mesures de similarité utilisées par ([132],[131],[142]) ou d'autres méthodes de récupération de structures taxonomiques à partir de folksonomies ([133], [134]) sont utiles pour amorcer le processus, mais leur précision pour identifier les connaissances des communautés est limitée.

L'ancrage sémantique des mesures proposées par [132] peut cependant aider à évaluer leur exactitude. Cependant, cette évaluation exige que les tags soient tous présents dans la liste des mots synonymes du dictionnaire Wordnet[93] ou dans d'autres ressources ontologiques : la validité de ces mesures ne pourra être évaluée que pour des tags étant des « mots communs » et non pour des tags qui sont des termes spécifiques. Ces « tags spécifiques », inventés par les utilisateurs, constituent l'un des avantages les plus précieux des folksonomies. Le même argument peut être utilisé pour d'autres approches comme [141] qui utilisent des ressources ontologiques partagées (dictionnaires etc.) pour structurer formellement les folksonomies.

D'autre part, les approches qui reposent sur des entrées de l'utilisateur (pour « tagger les tags », ou pour lier un tag à un concept non ambigu) peuvent induire, sans interfaces utilisateur conviviales adaptées aux usages, une surcharge cognitive que les utilisateurs réguliers du social tagging ne sont pas prêts à supporter.

Les approches intégrées tentent de surmonter cette limite en combinant le traitement automatique et la validation de l'utilisateur.

Cependant, aucune de ces deux approches ne prend formellement en compte la multiplicité des points de vue au sein d'une communauté, une caractéristique au cœur de notre approche.

6.2.3 Enrichissement sémantique de folksonomies

Une méthode automatique et générique pour enrichir sémantiquement tous types de folksonomies semble hors de portée aujourd'hui.

[92] http://www.w3.org/2004/02/skos/core
[93] https://wordnet.princeton.edu/

Notre approche de l'enrichissement sémantique des folksonomies consiste à combiner la manipulation automatique, l'amorçage du processus et les contributions des utilisateurs au coût le plus bas possible grâce à des interfaces conviviales.

Nous proposons un système qui supporte les points de vue contradictoires sur l'organisation sémantique des tags, mais qui aide aussi les communautés en ligne à construire un point de vue consensuel émergeant des contributions individuelles.

6.2.3.1 *L'ontologie SRTag : utiliser des graphes nommés pour suivre les points de vue divergents.*

Afin de modéliser la structuration sémantique des folksonomies tout en soutenant des vues contradictoires, nous proposons un schéma RDF intitulé SRTag[94], qui utilise des mécanismes de graphes nommés ([147], [148]). Les graphes nommés permettent de réifier la relation sémantique entre deux balises ou deux concepts (modélisés avec SKOS) sans le fardeau de la réification RDF classique[95] (voir Figure 70).

Les avantages et les raisons d'utiliser des graphes nommés pour capturer des points de vue sont détaillés dans [138]. Nous encapsulons les déclarations sur les tags dans un graphe nommé, puis les graphes nommés sont typés avec notre classe `srtag:TagSemanticStatement` ou avec des sous-classes plus précises.

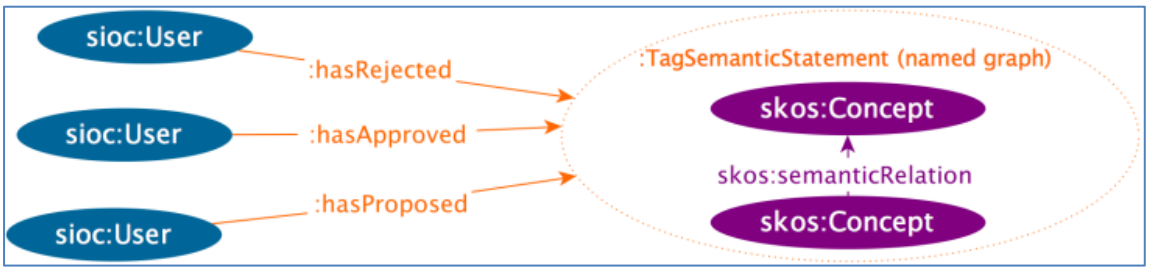


Figure 70. Schéma RDF de SRTag

Nous avons modélisé les relations classiques (hyponymie, synonymie, etc.) que l'on trouve dans des thésaurus (et qui peuvent être modélisées en SKOS). Ensuite, nous avons modélisé une série limitée d'actions sémantiques qui peuvent être exécutées par les utilisateurs (représentés par la classe `sioc:User`) sur les relations entre tags proposées par l'interface utilisateur, lors de la saisie. Ces actions sur les relations sont : « approuvée » (le système suggère une relation entre deux tags et on l'approuve), « proposée » (c'est l'utilisateur qui suggère la relation manuellement), « rejetée » (l'utilisateur n'approuve pas la relation proposée), représentées par les relations : `srtag:hasApproved`, `srtag:hasProposed`, et `srtag:hasRejected`.

Nous sommes alors en mesure de capturer et de suivre les opinions des utilisateurs (rejeter ou approuver) sur les relations affirmées, et donc de recueillir des points de vue divergents.

[94] http://ns.inria.fr/srtag/2009/01/09/srtag.html
[95] http://www.w3.org/TR/rdf-mt/ - Reif

6.2.3.2 Cycle de vie de l'enrichissement de folksonomies

Notre approche de la prise en compte de points de vue multiples dans la structuration d'une folksonomie permet la factorisation des contributions individuelles ainsi qu'une vision consensuelle liée à un utilisateur référent.

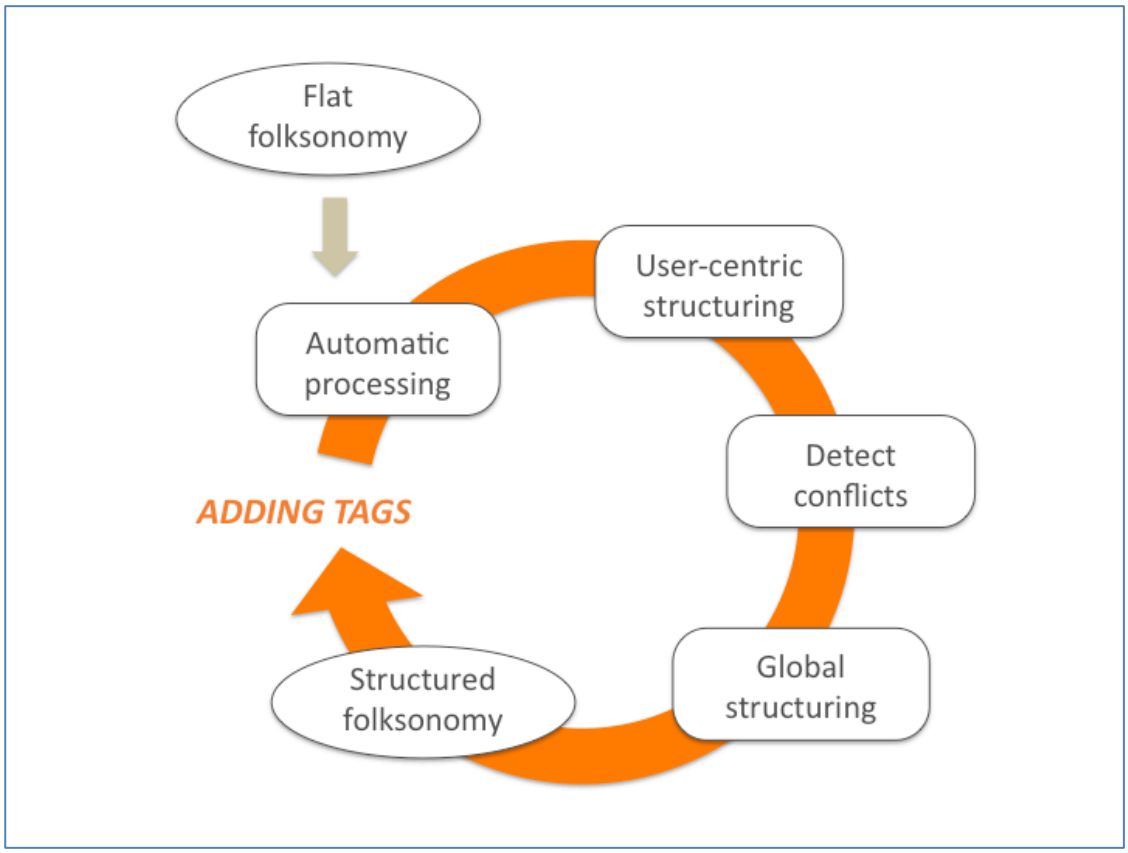


Figure 71. Cycle de vie de l'enrichissement de la folksonomie

De plus, en incluant dans notre modèle différents types d'agents traitant des relations entre tags, nous sommes en mesure de mettre en place un cycle de vie complet de folksonomies enrichies (voir Figure 71). Ce cycle de vie commence avec une folksonomie "plate" (c'est-à-dire sans relations sémantiques entre les tags) et peut être décomposé comme suit :

1. Le traitement automatique est effectué sur les tags à l'aide de méthodes basées sur une analyse des « labels sur les tags » et sur la structure du réseau de la folksonomie. Des agents automatique, de type `srtag:TagStructureComputer` vont alors ajouter des assertions dans le triple store énonçant les relations sémantiques pouvant être extraites entre les tags. Ces calculs sont effectués du jour au lendemain en raison de leur complexité algorithmique. Ils peuvent prendre plusieurs heures sur un ordinateur de 2010.
2. Les agents humains, modélisés comme instances de `srtag:SingleUser`, contribuent à l'aide d'interfaces utilisateur conviviales intégrées dans les outils qu'ils utilisent quotidiennement, en suggérant, corrigeant ou validant les

relations entre tags. Chaque utilisateur maintient ainsi son point de vue, tout en bénéficiant des points de vue des autres utilisateurs.

3. Comme des incohérences logiques peuvent apparaître entre les points de vue de tous les utilisateurs, un autre type d'agent automatique (`srtag:ConflictSolver`) détecte ces conflits et propose des résolutions. Les énoncés proposés sont utilisés pour réduire le bruit qui peut entraver l'utilisation de notre système lorsque, par exemple, différentes relations sont énoncées au sujet d'une même paire de tags.
4. Les déclarations de l'agent de résolution de conflits sont également utilisées pour aider l'utilisateur référent dans sa tâche qui consiste à maintenir une vision globale et consensuelle, sans conflits. Cette vue peut ensuite être utilisée pour filtrer les suggestions de tags connexes en donnant la priorité aux balises validées par cette vue de référence, par rapport aux autres balises suggérées automatiquement.
5. A ce point du cycle de vie, nous avons une folksonomie sémantiquement structurée dans laquelle le point de vue de chaque utilisateur coexiste avec le point de vue consensuel. Ensuite, un ensemble de règles est appliqué pour exploiter ces points de vue afin d'offrir une navigation cohérente à tous les utilisateurs.
6. Un autre cycle redémarre avec des manipulations automatiques pour prendre en compte les nouvelles balises ajoutées à la folksonomie.

6.2.4 Extraction automatique de la sémantique émergente

Plusieurs types de méthodes peuvent être appliquées aux folksonomies afin de calculer automatiquement des relations sémantiques entre les balises.

Nous présentons d'abord l'expérience que nous avons menée pour évaluer la performance des méthodes de calcul de similarité entre chaines de caractères (strings) et présentons notre proposition pour combiner efficacement les plus pertinentes.

Puis nous présentons notre implémentation utilisant des algorithmes représentant l'état de l'art pour analyser la structure des folksonomies ([14], [15]).

6.2.4.1 Évaluation de la performance des métriques basées sur les chaînes de caractères

Vue d'ensemble des métriques existantes basées sur les chaînes de caractères

Les mesures de distance basées sur des chaînes de caractères considèrent les labels des tags à comparer. Par exemple, la distance de Levenshtein [149] a été utilisée dans [142] pour regrouper des variantes orthographiques telles que "new_york" et "newyork".

Pour aller plus loin dans l'utilisation de ces méthodes, nous avons effectué un benchmark pour évaluer la capacité de ces métriques à récupérer d'autres types de relations sémantiques telles que la relation *connexe*, ou relation *plus spécialisé* ou *plus général*, également appelée relation *hyponyme*. Les relations hyponymiques reflètent le degré

relatif de généralité entre deux notions, par exemple : "*La pollution*" est plus « large » que "*la pollution des sols*". Dans d'autres cas, deux notions peuvent simplement être liées, comme par exemple "*énergie*" et "*électricité*".

Nous avons comparé l'ensemble des métriques de similarité implémentées dans le package SimMetrics[96] qui donnent, pour une paire de chaînes *(s1, s2),* une valeur normalisée entre 0 et 1, avec une valeur de 1 signifiant que les deux chaînes comparées sont les plus similaires.

Les mesures de similarité que nous avons comparées se répartissent en plusieurs catégories : (a) les méthodes basées sur la distance d'édition, qui considèrent l'ensemble des opérations nécessaires pour transformer la chaîne *s1* en chaîne *s2* ; (b) les méthodes basées sur des *tokens* -par exemple le coefficient de chevauchement-, qui décomposent les chaînes en tokens séparés par un espace; (c) les méthodes utilisant des représentations vectorielles de chaînes, par exemple celle calculant la similarité avec la méthode des cosinus ; et enfin (d) d'autres types de métriques telles que QGram ou Soundex.

Benchmarking

Nous avons construit manuellement un échantillon de test à partir des tags utilisés à l'ADEME pour indexer leurs documents et ressources. Cet échantillon, qui mélange des tags librement choisis et des tags choisis par les archivistes, a été divisé en 4 ensembles de 22 paires de tags (t_1,t_2), chaque ensemble contenant des paires de tags qui correspondent à une relation sémantique, à savoir : *variante orthographique*, *hyponyme*, *relié* et *non relié*. Ces relations ont été validées par un membre de l'équipe d'archivistes de l'ADEME afin qu'elles reflètent la connaissance du domaine de l'utilisateur.

La métrique Monge-Elkan est une métrique hybride basée sur les distances d'édition qui décompose également les chaînes en tokens, et utilise une seconde métrique pour comparer chaque token avec tous les autres. Pour notre expérience, nous avons utilisé une série de 15 métriques et la combinaison de ces 15 métriques avec la méthode Monge-Elkan, ce qui fait un total de 30 métriques différentes.

Pour évaluer la performance de chaque métrique dans la récupération des bonnes relations de notre échantillon, nous avons calculé pour chaque sous-ensemble le rappel, la précision et la moyenne harmonique pondérée F_1 (pour donner autant d'importance au rappel qu'à la précision).

Ces valeurs ont été calculées en faisant varier le seuil au-dessus duquel une paire d'étiquettes donnée est récupérée ou non. Ensuite, pour compter les paires positives fausses et les paires positives vraies qui ont été appariés, nous avons appliqué les règles suivantes : (a) pour le cas *relié*, les vrais positifs sont comptés à partir de tous les sous-ensembles à l'exception du sous-ensemble non apparenté, puisque les paires *variante orthographique* et *plus large/étroite (hyponyme)* peuvent également être considérées comme "apparentées" ; (b) pour la *variante orthographique* et le cas *hyponyme*, les vrais

[96] http://www.dcs.shef.ac.uk/~sam/stringmetrics.html

positifs n'étaient que ceux de leur sous-ensemble correspondant, et les paires de tous les autres sous-ensembles ont été comptées comme faux positifs.

Nous avons ensuite examiné la meilleure métrique pour chaque type de relation en les classant en fonction de la valeur moyenne et de l'écart type de F_1.

Le résultat de cette première évaluation est que la méthode Monge-Elkan_Soundex a surpassé d'autres mesures dans le cas *relié*.

Le meilleur dans le cas de la *variante orthographique* est la métrique Jaro-Winkler, et le meilleur pour le cas *hyponymique* est la métrique MongeElkan_NeedlemanWunch.

Dans ce dernier cas, toutefois, aucune des mesures les plus performantes n'a nettement surclassé les autres. Nous pouvons également remarquer un plus grand écart type dans le cas *relié* que dans les deux autres cas, et ce résultat était attendu puisque le fait que les deux notions soient liées se traduit rarement par des similitudes terminologiques, par exemple, "voiture" et "roue" sont liés mais ne partagent pas de lettres communes.

Identification des différents types de relations

Maintenant, nous nous intéressons à trouver un moyen, en utilisant ces métriques, de différencier les 3 types de relations sémantiques.

Tout d'abord, nous utilisons la métrique MongeElkan Soundex pour récupérer toutes les paires de tags liées, c'est-à-dire toutes les paires partageant une relation qui est *au moins* de type "*relié*", ce qui signifie que dans cette catégorie, nous récupérerons également les *variantes orthographiques* et les cas *hyponymes*. Pour ce faire, nous devons déterminer un seuil pour la valeur de similarité à partir de la métrique MongeElkan_Soundex au-dessus de laquelle une paire est considérée comme apparentée.

Pour déterminer ce seuil, nous avons examiné la valeur moyenne de similarité pour tous les cas *reliés* (variante orthographique, hyponyme, relié) et pour tous les cas *non reliés* dans l'ensemble de l'échantillon. Les résultats sont présentés à la Figure 72 (à gauche). Nous pouvons voir que, compte tenu des écarts, si nous choisissons une valeur seuil de 0.9, nous sommes en mesure d'éviter les paires non apparentées.

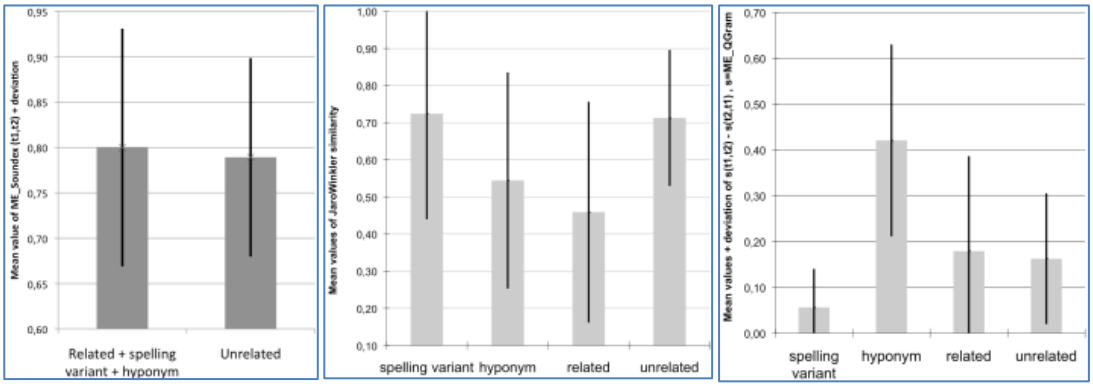


Figure 72. A gauche : comparaison de la valeur moyenne de la métrique MongeElkan_Soundex pour tous les cas connexes (variantes orthographiques, hyponymes et simples cas connexes) et pour les cas non connexes.
Milieu : comparaison de la valeur moyenne de la métrique JaroWinkler pour chaque type de relation sémantique.
A droite : valeur moyenne de la différence $\delta = s(t_1,t_2) - s(t_2,t_1)$ avec s étant la métrique Monge-Elkan_QGram pour chaque ensemble de paires de tags.

Pour distinguer la *variante orthographique* des paires *reliées*, nous examinons la valeur moyenne et l'écart type de la meilleure métrique dans le cas de la *variante orthographique*.

Dans la Figure 72 (au milieu), nous montrons la valeur moyenne de la métrique Jaro-Winkler pour les quatre types de relations sémantiques. Nous voyons que, compte tenu de l'écart, si nous choisissons un seuil supérieur à 0.9, nous avons plus de chances de récupérer les paires de *variantes orthographiques*. Ce résultat est confirmé lorsque l'on regarde la valeur seuil pour laquelle F_1 est maximum pour la mesure JaroWinkler dans le cas de la variante orthographique.

Ensuite, nous voulons trouver un moyen de différencier les paires d'*hyponymes* des paires *reliées*. Les métriques de MongeElkan ne sont pas symétriques, et nous avons calculé, pour chaque paire de tags (t_1, t_2), la différence $\delta = s(t_1, t_2) - s(t_2, t_1)$, s étant l'une des 15 combinaisons de MongeElkan avec une autre métrique. Dans la Figure 72 (à droite), nous donnons la valeur moyenne et l'écart type δ pour chaque ensemble de paires de tags selon la métrique MongeElkan_QGram, qui a donné les meilleurs résultats à cet égard. Nous n'avons inclus dans ce calcul que les paires de tags qui ont été récupérées grâce à la métrique MongeElkan_Soundex et ont été identifiés "*reliés*". Nous pouvons voir que si nous choisissons un seuil supérieur à 0.39 (la valeur la plus élevée pour δ en incluant l'écart), nous sommes en mesure de récupérer les tags partageant une relation *hyponymique*. En tenant compte du signe de la différence, nous sommes capables de dire la direction de cette relation, ce qui signifie que si nous avons δ négatif et au-dessus d'un certain seuil, alors t_1 peut être considéré comme plus spécialisé que t_2.

Méthodes heuristiques basées sur des chaînes de caractères.

En conséquence, nous sommes en mesure de proposer une heuristique (voir Algorithme 1) qui combine les meilleures métriques pour récupérer différentes relations sémantiques entre les tags.

Algorithme 1 : Métrique heuristique sur les chaînes de caractères pour récupérer les relations sémantiques entre les tags.

---

for all distinct pair of tags (t_i,t_j) from $S = \{t_1,t_2,...,t_n\}$ **do**
if *MongeElkanSoundex*$(t_i,t_j) > \tau_a$
then
if *JaroWinkler*$(t_i,t_j) > \tau_b$
then t_i has spelling variant t_j
else
if *MongeElkanQGram*(t_i,t_j) - *MongeElkanQGram*$(t_j,t_i) \leq -\tau_c$
then t_i has broader t_j
else
if *MongeElkanQGram*(t_i,t_j) - *MongeElkanQGram*$(t_j,t_i) \geq \tau_c$
then t_j has broader t_i
else t_i has related t_j
end if
end if
end if
end if
end for

Nous recherchons d'abord des paires de tags *reliés* (t_1, t_2) en utilisant Monge-Elkan_Soundex avec un premier seuil τ_a de sorte que nous avons $s(t_1, t_2) \geq \tau_a$. Ce premier seuil est choisi comme indiqué dans la section précédente, c'est-à-dire en prenant $\tau_a = 0,9$ dans notre cas.

Ensuite, nous comparons la similarité de JaroWinkler avec un deuxième seuil τ_b pour voir si les tags sont des variantes orthographiques, telles que $s(t_1, t_2) \geq \tau b$. Dans ce cas, le seuil est choisi comme indiqué dans la section précédente, c'est-à-dire, dans notre cas, $\tau b = 0,94$.

Si ce n'est pas le cas, nous utilisons un troisième seuil τ_c et nous calculons la différence δ de la métrique MongeElkan_QGram $\delta = s(t_1, t_2)$ - $s(t_1, t_2)$,et si δ est tel que $\delta \leq -\tau_c$, alors nous pouvons inférer que t_1 est plus spécialisé que t_2, ou si $\delta \geq \tau_c$ alors t_1 est considéré comme plus général que t_2.

Le troisième seuil est choisi d'après les résultats présentés Figure 72, en choisissant une valeur supérieure à 0.39. Dans ce processus, nous donnons la priorité à la détection des *variantes orthographiques* puisque les méthodes basées sur des chaînes de caractères sont mieux adaptées à ce type de relation, et en vérifiant d'abord ce cas, nous nous assurons de récupérer autant de cas de variantes orthographiques que possible, car ceux qui ont été récupérés ont statistiquement plus de chances d'être de vrais positifs.

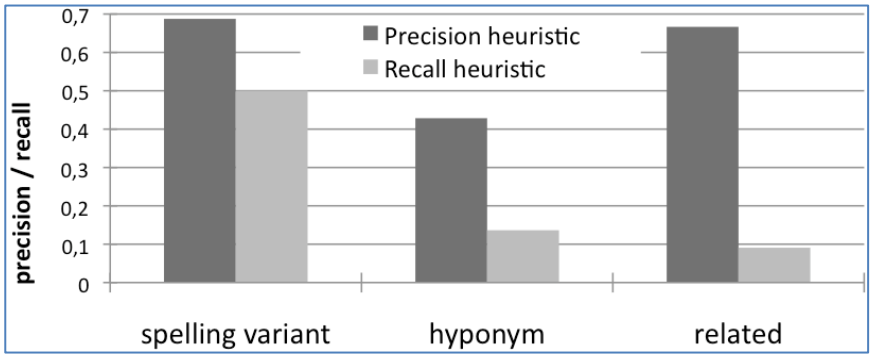


Figure 73. Performance de la métrique heuristique basée sur les chaînes de caractères (Algorithme 1).

Nous avons appliqué cette méthode heuristique au même échantillon de test. Cependant, cette heuristique n'est pas directement comparable aux autres métriques car elle combine différentes méthodes et récupère 3 types de relations sémantiques à la fois, alors que dans l'expérience de comparaison globale, chaque métrique traitait un seul type de relation sémantique à la fois.

Ainsi, afin d'évaluer quantitativement la performance globale de cette métrique heuristique à base de chaines de caractères, nous montrons dans la Figure 73 les valeurs de précision et de rappel pour les 3 types de relations.

Nous pouvons clairement voir dans cette figure que les métriques basées sur les chaînes de caractères sont les plus performantes dans le cas de la variante orthographique, ce qui confirme une intuition naturelle puisque les méthodes basées sur les chaînes de caractères ont été conçues à l'origine pour identifier des chaînes de caractères similaires.

Néanmoins, la performance perceptible dans le cas de *l'hyponymie* s'explique par la capacité des métriques à base de chaines de caractères à détecter facilement les tokens communs que l'on retrouve par exemple dans "*pollution*" et "*pollution du sol*" et ces cas correspondent souvent à une relation *hyponymique*.

Le cas *relié* est plus difficile (d'où la faible précision) car cette relation est la plus floue et probablement la moins perceptible dans l'orthographe réelle des balises ("*soleil*" et "*énergie*", par exemple).

Cependant, nous avons montré ici que différents types de similitudes morphologiques peuvent être mis en correspondance avec des types spécifiques de relations sémantiques, avec des niveaux de précision variables, la *variante orthographique* et *l'hyponyme* étant les plus précis.

Nous présentons maintenant d'autres méthodes pour extraire les relations sémantiques entre les balises.

6.2.4.2 Analyse de la structure de la folksonomie

Dans cette section, nous détaillons notre mise en œuvre de deux méthodes (Algorithme 2 et Algorithme 3 dans le texte qui suit) basées sur l'analyse tripartite de la folksonomie (users-resources-tags) afin d'extraire la sémantique émergente entre tags.

Algorithme 2 : afin d'identifier les tags « *reliés* », nous utilisons la mesure de similarité basée sur l'agrégation distributive dans le contexte tag-tag proposée par Cattuto et al. [6] (Figure 74).

Ces derniers ont comparé les différents contextes dans lesquels les mesures de similarité peuvent être calculées, et ont étudié les différents types de relations sémantiques qu'elles apportent en utilisant la structure hiérarchique de Wordnet.

Cette expérience montre que les tags associés par des mesures de similarité basées sur de simples cooccurrences tendent à partager des relations de subsomption, alors que les tags associés par des mesures de similarité de distribution dans le contexte tag-tag tendent à se situer au même niveau d'une hiérarchie sémantique, en ayant les mêmes parents et grands-parents dans la hiérarchie.

Cattuto et al. font l'hypothèse qu'en utilisant deux tags sur une seule ressource (cooccurrence) l'utilisateur peut avoir tendance à couvrir différents niveaux de généralité.

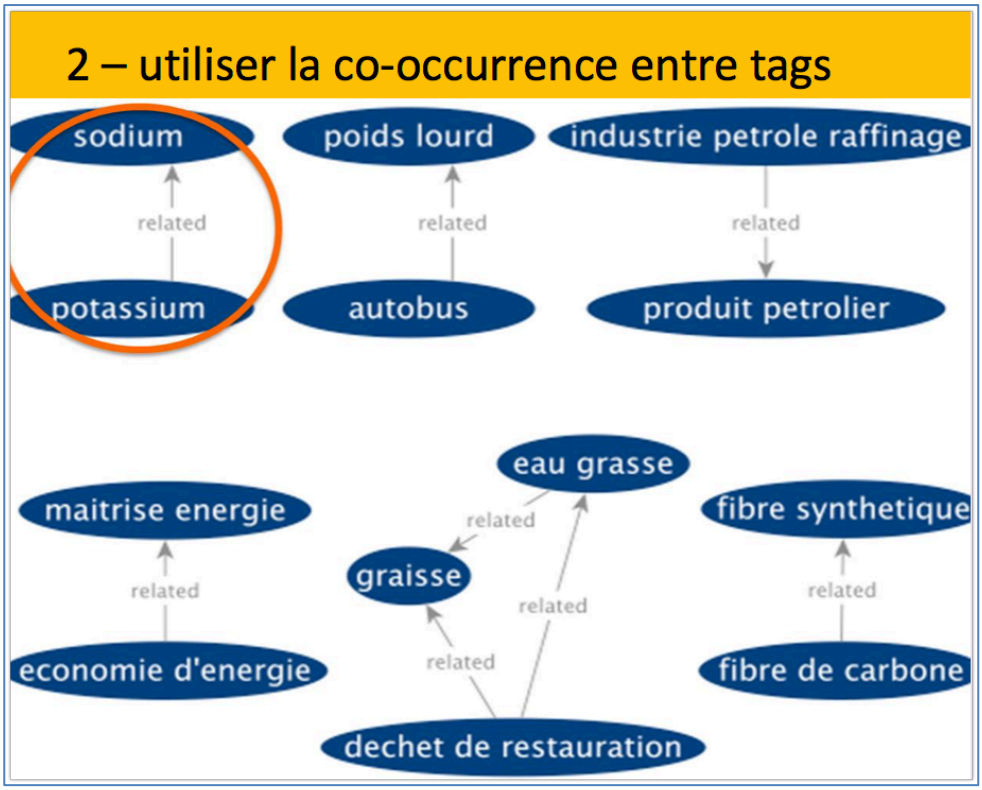


Figure 74. Utilisation de la cooccurrence de tags pour inférer des relations.

Par exemple, les tags "java" et "programmation" sont susceptibles d'être utilisées simultanément, et on peut supposer qu'ils ont, dans l'esprit de l'utilisateur, différents niveaux de généralité.

La relation identifiée par la mesure de distribution basée sur le contexte tag-tag associe des tags qui partagent des schémas de cooccurrence similaires, mais qui ne sont rarement utilisés ensemble. C'est le cas par exemple des tags "java" et "python" qui sont rarement utilisés ensemble, mais on les trouve souvent utilisée avec le tag "programmation".

Pour calculer la similarité du contexte tag-tag, nous considérons d'abord la représentation vectorielle v_i de chaque tag t_i dans ce contexte. Chaque entrée de ce vecteur v_i est donnée par $v_{ti}t_j = w(t_i,t_j)$ pour $t_i \neq t_j$ où $w(t_i, t_j)$ correspond à la valeur de cooccurrence pour les tags (t_i, t_j), et $v_{ti}t_i = 0$.

Nous mettons à zéro la valeur d'un tag relié avec lui-même de sorte que nous considérons les tags comme étant liées lorsqu'ils se trouvent dans un contexte similaire, mais pas lorsqu'ils sont directement co-occurrents. La valeur de similarité pour une paire de balises (t_i,t_j) dans le contexte tag-tag est alors donnée par la distance cosinus entre les vecteurs v_i et v_j :

$$\cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|^2 \cdot \|v_j\|^2}$$

Lorsque cette valeur est supérieure à un seuil donné, nous créons une annotation disant que la balise t_i est liée à la balise t_j.

Algorithme 3 Pour extraire les relations de subsomption, nous avons utilisé la méthode décrite par P. Mika [133] qui consiste à regarder les inclusions des ensembles d'utilisateurs associés à un tag (Figure 75). Soit S_i l'ensemble des utilisateurs utilisant le tag t_i, et S_j l'ensemble des utilisateurs utilisant le tag t_j. Si l'ensemble S_i est inclus dans l'ensemble S_j, de sorte que nous avons $S_i \subset S_j$, avec $card(S_i) > 1$ et $card(S_j) > card(S_i)$, nous pouvons en déduire que le tag t_j est plus général que le tag t_i.

Prenons un exemple : si l'ensemble des utilisateurs qui a utilisé « *pollution des sols* » est compris dans l'ensemble des utilisateurs qui ont utilisé « *pollution* », alors « pollution » est plus général que « pollution des sols ».

Notez que ces deux algorithmes ne sont pas incrémentaux puisque nous devrons analyser l'ensemble de la folksonomie pour calculer la similarité des tags nouvellement ajoutées.

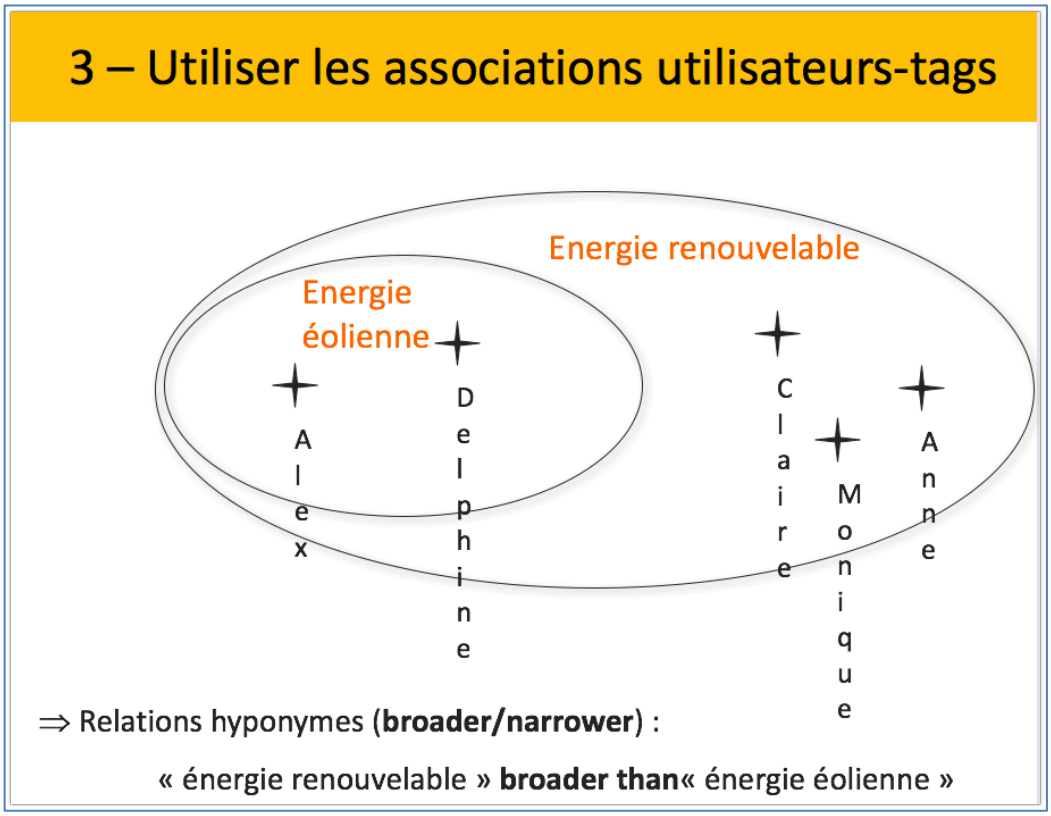


Figure 75. En regardant l'usage des tags par des groupes d'utilisateur, on peut inférer des relations hyponymes (ici, si l'ensemble des utilisateurs ayant utilisé « Energie renouvelable » inclut les utilisateurs ayant utilisé « Energie Eolienne », alors ce dernier est une spécialisation (narrower) du tag « Energie Renouvelable »

Traitement automatique sur un dataset du monde réel

Nous avons effectué les trois types de calcul décrits précédemment sur un ensemble de données du monde réel (voir Table 1) composé des éléments suivants : (a) le *delicious* dataset provient de delicious.com et se compose du tagging des utilisateurs qui ont tagué au moins un de leurs favori avec l'étiquette "ademe" du 1er octobre 2009 à mi-2010. (b) le *thesenet* dataset provient d'une base de données de l'ADEME qui répertorie tous les projets de doctorat financés par l'agence. Chaque mot-clé utilisé pour catégoriser les projets/thèses archivés a été considéré comme un tag, chaque projet identifié comme une ressource étiquetée et chaque doctorant comme le tagger. (c) le *caddic* dataset est constitué de l'ensemble des entrées des cinq dernières années (1005-2010) de la base d'indexation des documents des archivistes de l'ADEME.

Chaque document correspond à une ressource tagguée. Dans le tableau 1, nous détaillons, pour chaque dataset : le nombre de tags distincts ; le nombre de taggings (c'est-à-dire le nombre de liens tripartites entre une ressource, un tag et un utilisateur) ; le nombre de ressources distinctes tagguées ; et le nombre d'utilisateurs.

Table 1: Description des datasets utilisés.

	Delicious	Thesenet	Caddic	Full Dataset
Nb. distinct Tags	1015	6583	1439	9037
Nb. Restricted Tagging (1R - 1T - 1U)	3015	10160	25515	38690
Nb. distinct Resources	196	1425	4765	6386
Nb. distinct Users	812	1425	1	2238

Dans la Table 2, nous donnons quelques détails sur les résultats obtenus pour chacune des trois méthodes de calcul (Algorithmes 1, 2 et 3) lorsqu'elles sont appliquées aux différents datasets. La première chose à noter est que l'algorithme 1 donne beaucoup plus de résultats (71034 résultats) que l'algorithme 2 (8377 résultats au total, avec 97% de données issues du *delicious* dataset).

Les résultats de l'algorithme 2 peuvent être expliqués parce que cette méthode examine le modèle de cooccurrence des tags, et *delicious* est le dataset dans lequel deux tags sont plus susceptibles d'avoir des modèles similaires de cooccurrence puisque, si nous regardons le rapport entre le nombre de taggings et le nombre de ressources distinctes, nous obtenons 15.38 pour *delicious*, 7.13 pour *thesenet*, et 5.35 pour *caddic*.

Table 2 : Description des résultats du traitement automatique sur les tags

	Algo1	**Algo 2**			**Algo3**		**Total**
	Full Dataset	Delicious	Thesenet	caddic	Delicious	thesenet	
related	59889	8141	206	30	-	-	68633
Hyponym	10952	-	-	-	106	196	11254
Spelling variants	3193	-	-	-	-	-	3193
Comp. time (s)	20952	4200	180	300	5	10	25647
Total number of statements							83080
Nb. of pairs with overlapping statements between different methods							31
Nb. of pairs with conflicting statements between different methods							22
Total number of statements on distinct pairs							83027

Cela signifie qu'il y a plus du double d'utilisateurs distincts qui ont taggué la même ressource dans *delicious* que dans *thèsenet* ou *caddic*. De plus, dans *delicious*, un plus grand nombre d'utilisateurs ont taggué la même ressource à l'aide d'un plus petit ensemble de tags distincts, d'où une plus grande probabilité que deux tags aient des modèles similaires de cooccurrence.

Pour l'algorithme 3, nous avons obtenu un plus grand nombre de relations dans l'ensemble de données *thèsenet* que dans l'ensemble de données délicieuses puisque le dataset *thèsenet* compte environ 75 % d'utilisateurs de plus et encore plus de tags distincts (environ 6 fois plus), d'où une plus grande probabilité d'avoir des ensembles imbriqués d'utilisateurs ayant utilisé les mêmes tags.

Dans la partie inférieure du tableau 2, nous voyons qu'au total, nous avons obtenu 83080 relations à partir des 3 types de calcul appliqués à nos différents datasets. Dans 31 cas, différentes méthodes ont déduit les mêmes relations, et dans 22 cas, certaines méthodes sont contradictoires. Après avoir éliminé les relations redondantes et qui sont contradictoires, nous obtenons un total de 83027 relations (Figure 77).

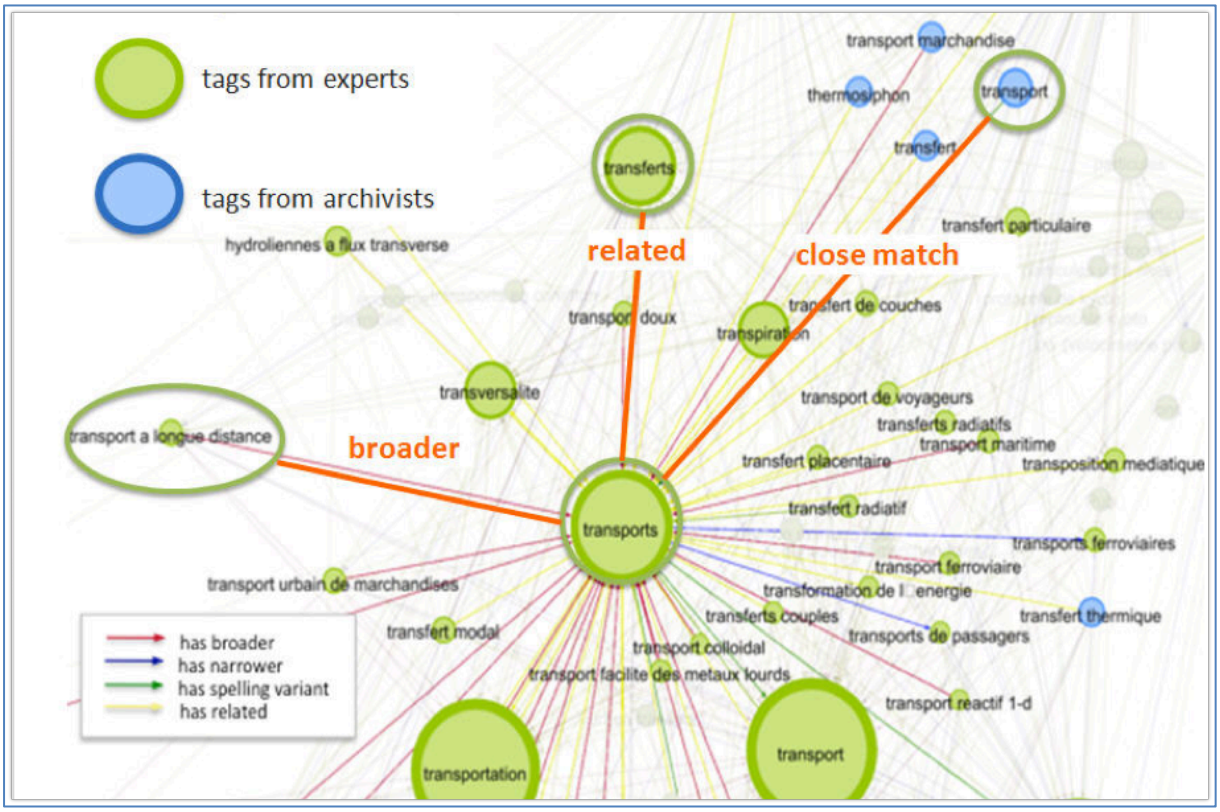


Figure 76. Exemple des résultats issus du traitement automatique montrant les tags liés au tag "transports". La taille des nœuds indique le nombre de relations entrantes (en degrés). Les nœuds verts correspondent aux tags des datasets *delicious* et *thesenet* (d'où la présence de deux nœuds "transport"), et les nœuds bleus correspondent aux tags du dataset *caddic*.

Ce traitement automatique est effectuée pendant les périodes de faible activité en raison de sa complexité algorithmique, et chaque relation résultante est liée au type d'agent correspondant (ici traitement automatique et non une action de l'utilisateur) : chaque instruction étant modélisée comme une sous-classe de `srtag:AutomaticAgent`.

De plus, les algorithmes 2 et 3 ne sont pas incrémentaux puisque lorsque de nouveaux tags sont ajoutées, la structure de l'ensemble de la folksonomie est modifiée.

Ce n'est pas le cas pour l'algorithme 1 qui ne compare que les labels des tags nouvellement ajoutées avec tous les autres tags.

Pour donner un exemple du temps de calcul, le temps total pour appliquer ces 3 méthodes sur l'ensemble des données est de 25647s dans notre configuration de 2010, avec une machine équipée d'un processeur Intel Core2 Duo 4 cœurs fonctionnant à 3.00 GHz avec 8Go de RAM. La Figure 76, présente les résultats d'une analyse obtenue pour l'extraction automatique de relations autours du tag "transports".

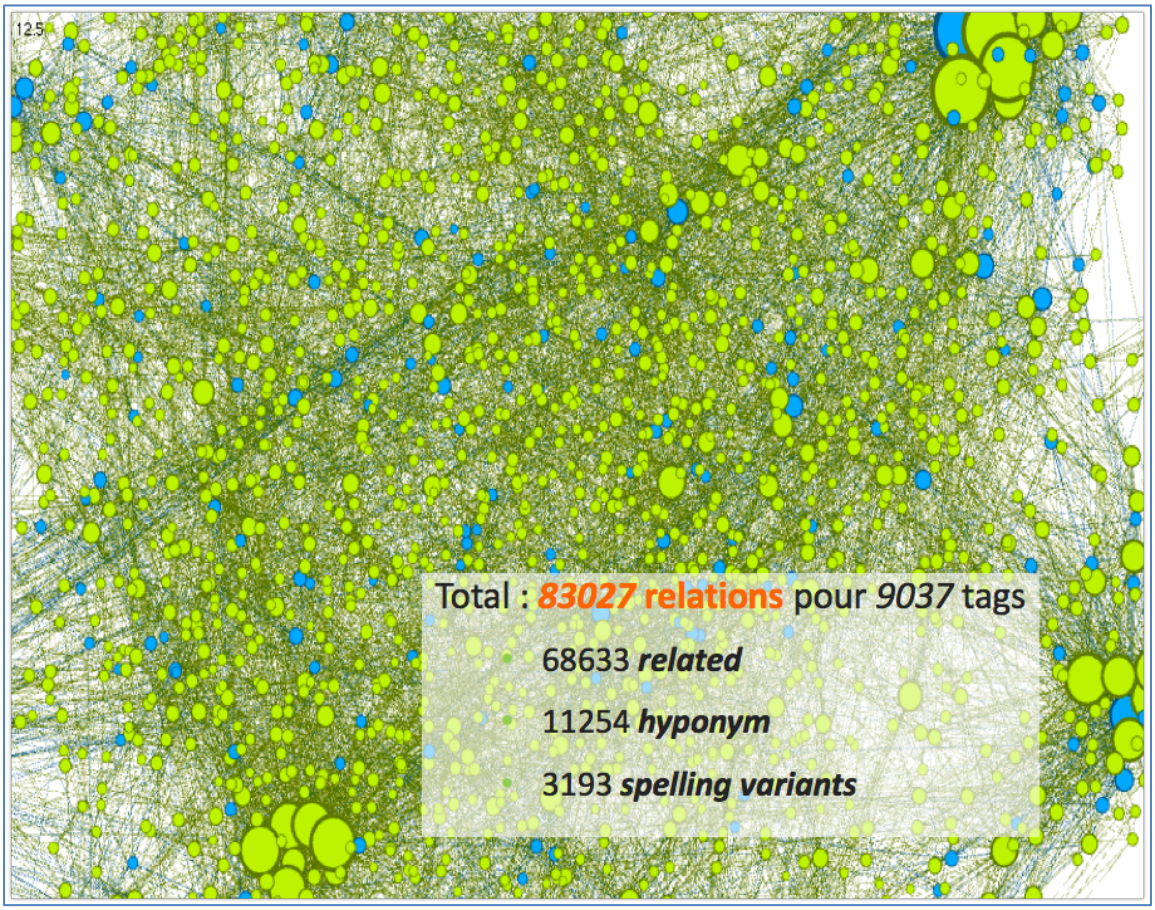


Figure 77. Résultats sur le jeu de données complet.

6.2.5 Capture et exploitation des contributions individuelles

Jusqu'à présent, nous avons présenté les différentes méthodes de calcul des relations entre tags et le modèle, SRTag, pour capturer les points de vue divergents de tous les utilisateurs. Maintenant, nous allons voir comment ces points de vue sont d'abord capturés, puis triés et arrangés ensemble dans une vue cohérente.

6.2.5.1 Capturer les contributions des utilisateurs

Une fois que nous sommes capables de supporter des points de vue divergents, nous désirons permettre aux utilisateurs de contribuer à la structuration sémantique de la folksonomie tout en maintenant « la friction cognitive » (les efforts à fournir) aussi basse que possible.

Pour atteindre cet objectif, nous proposons d'intégrer des fonctionnalités structurantes simples et non intrusives dans les tâches quotidiennes de l'utilisateur. Par exemple, dans notre communauté cible à l'ADEME, nous voulons pouvoir capturer l'expertise des ingénieurs lorsqu'ils parcourent le corpus des ressources de l'agence.

Notre proposition consiste en une interface pour expliciter la structure calculée de la folksonomie dans laquelle les tags sont suggérées et ordonnées en fonction de leurs relations sémantiques avec le tag recherchée (voir Figure 78).

Les balises *reliées* (« related » dans la figure) et les *variantes orthographiques* (« close match » dans la figure) sont positionnées sur le côté droit (respectivement dans le coin supérieur et inférieur) et les balises plus générales (« broader ») et plus spécialisées (narrower ») sont positionnées sur le côté gauche (respectivement dans le coin supérieur et inférieur).

La Figure 78 et la Figure 79 montrent un exemple d'interface utilisateur de recherche de ressource par tag. Ici l'utilisateur a cherché des ressources tagguées avec « energie». On propose les résultats de recherche (à droite) mais aussi les tags reliés dans la partie gauche. L'utilisateur peut drag'n'dropper les différents tags proposés dans une des catégories pour établir une relation manuelle, ou ajouter le tag recherché dans une catégorie s'il n'existe pas encore, ou supprimer de stags. Ceci ne sera effectué que pour son « propre point de vue » dans un premier temps.

Nous recommandons au lecteur de visionner une vidéo en ligne qui montre l'utilisation de cette interface de manière interactive, afin de mieux appréhender l'ergonomie de l'application[97]

[97] https://youtu.be/-ZKhmx5wpj0

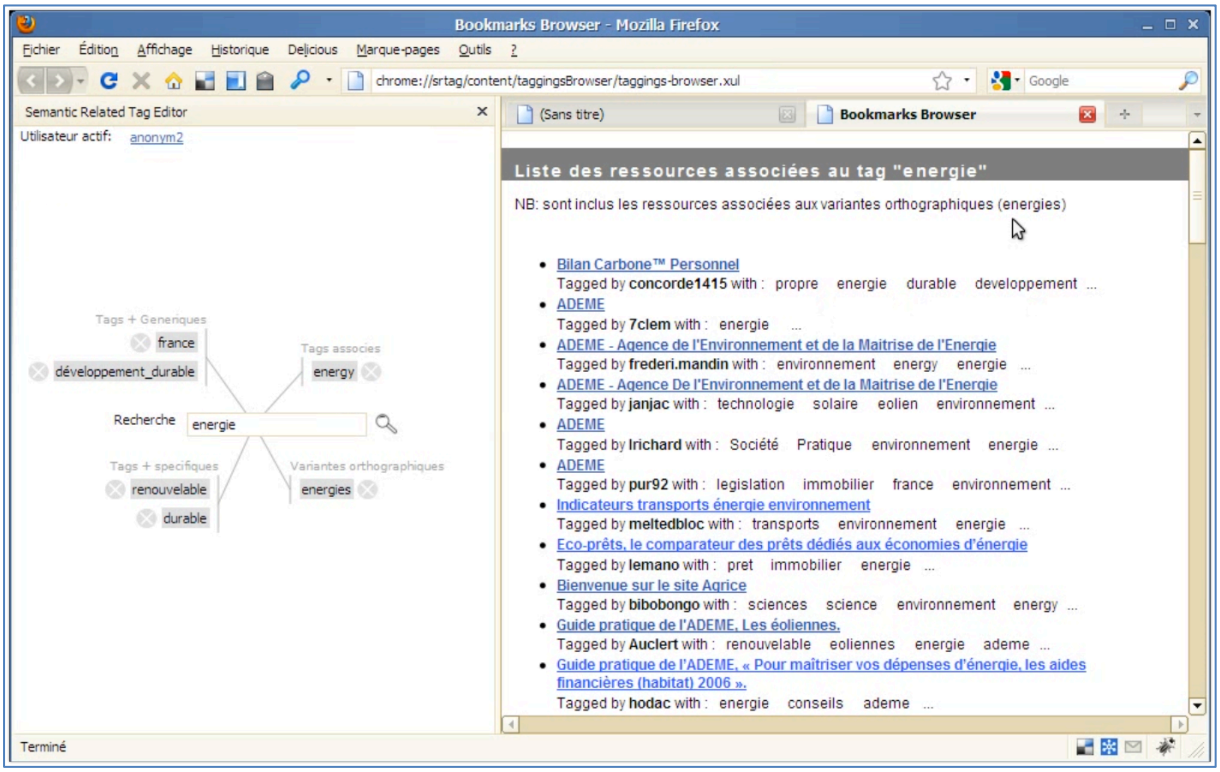


Figure 78. Extension Firefox intégrant de manière transparente des outils interactifs pour la structuration des balises (partie gauche).

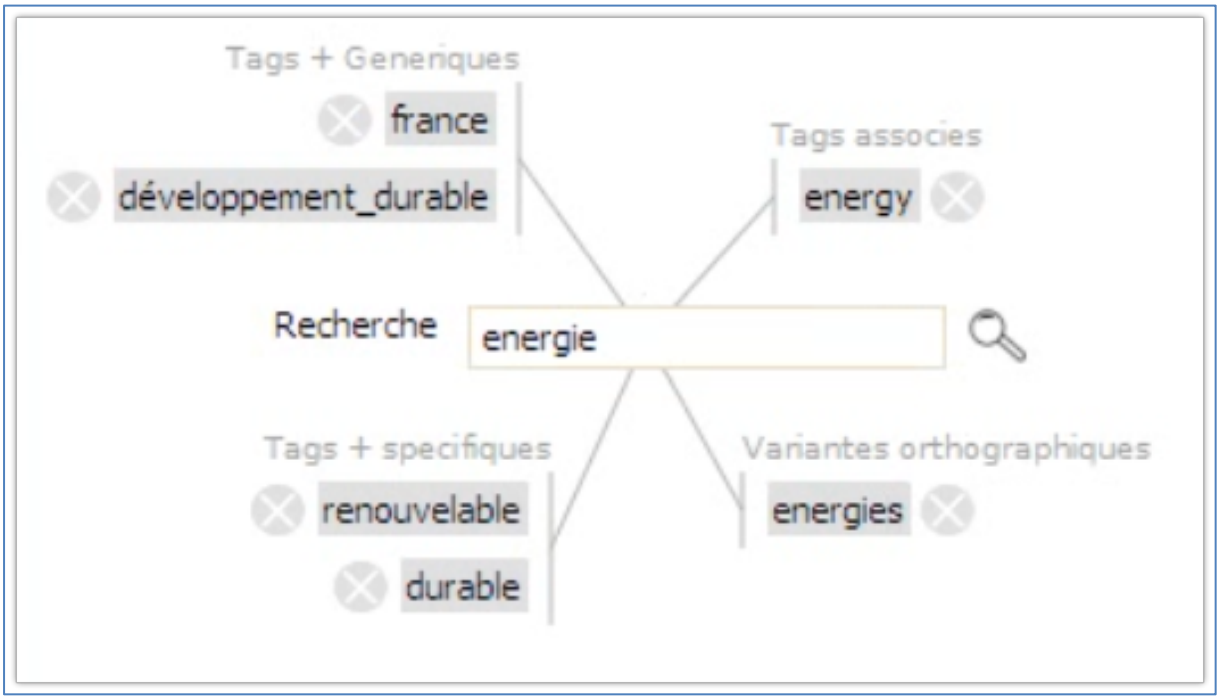


Figure 79. Zoom sur la partie de structuration interactive « selon le point de vue de l'utilisateur courant ».

6.2.5.2 Détection et résolution des conflits

Mécanisme de résolution des conflits

Un troisième type d'agent est introduit, modélisé avec une sous-classe de `srtag:AutomaticAgent` nommé `srtag:ConflictSolver` et qui recherche les conflits émergeant entre les points de vue de tous les utilisateurs.

Un conflit dans la folksonomie structurée apparaît lorsque différentes relations ont été proposées ou approuvées par différents utilisateurs sur la même paire de tags (si un utilisateur change d'avis, nous mettons simplement à jour son point de vue).

Par exemple, le tag "pollution" est *plus spécialisé* que "co2" pour un nombre n_1 d'utilisateurs, mais pour un nombre n_2 d'utilisateurs "pollution" est *plus général* que "co2". En outre, d'autres utilisateurs peuvent dire que la "pollution" est *relié* au "co2".

Dans ce cas, le résolveur de conflits compte d'abord le nombre d'approbation $nbApp_i$ pour chaque énoncé conflictuel $s_i \in \{s_i\}_n$ avec n étant le nombre total de relations existants sur une paire donnée de tags. Ensuite, il récupère le maximum $max\{nbApp_i\}_{i\in[1,n]} = nbApp_{max}$, et compare le rapport $r = \frac{nbApp_{max}}{\sum_i nbApp_i}$ avec un seuil donné τ_{cs}.

Si ce ratio est supérieur à τ_{cs}, alors le résolveur de conflits approuve la relation correspondante. Sinon, si r est en dessous, cela signifie qu'aucun consensus fort n'a encore été atteint, et le résolveur de conflits dit simplement que les deux tags sont *reliés* puisque cette relation (la plus vague) représente un compromis entre chaque point de vue divergent. Dans ce cas, le résolveur approuve la relation « *relié* » si elle existe, ou la rajoute si elle n'existe pas.

Expérimentation

Protocole : nous avons mené une expérience auprès de 5 employés de l'ADEME. Nous leur avons présenté une liste de 94 paires de tags (t_1,t_2) et leur avons demandé de choisir une relation sémantique entre t_1 et t_2 parmi les suivantes :

1. t_1 est une *variante orthographique* de t_2,
2. t_1 est *plus général* que t_2,
3. t_1 est *plus spécialisé* que t_2,
4. t_1 est *relié* à t_2,
5. t_1 *n'est pas lié* à t_2.

Nous avons ensuite appliqué le résolveur de conflits sur l'ensemble des relations et des points de vue. Lorsqu'un utilisateur a choisi la cinquième possibilité, c'est-à-dire que t_1 n'est pas lié à t_2, nous avons appliqué une règle SPARQL pour traduire ce choix en rejetant toutes les relations déclarées sur la même paire de tags.

Cela nous permet de considérer des relations *discutables*, en ce sens que certains utilisateurs les ont approuvées et que d'autres les ont rejetées, mais qu'aucun n'a proposé ou approuvé une autre relation à la place.

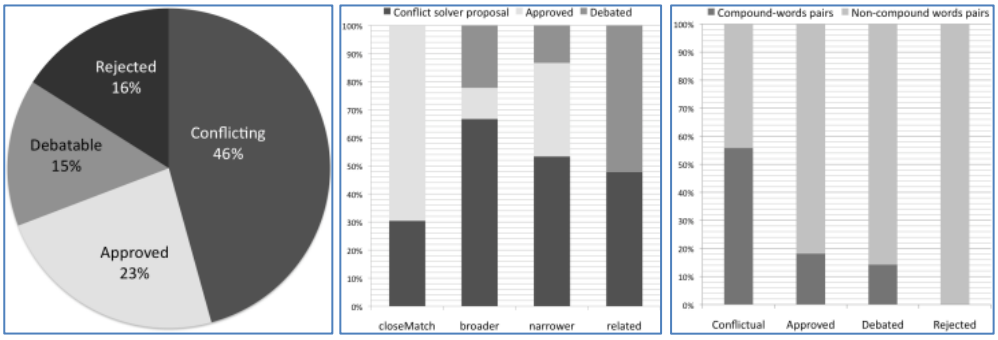


Figure 80. Résultat de la résolution de conflits. De gauche à droite : a) répartition des différents cas de résolution de conflits pour toutes les paires d'étiquettes. (b) répartition des différents cas de résolution de conflits pour chaque type de relations sémantiques. c) répartition des paires de tags qui sont chacun des mots composés.

Après avoir appliqué le résolveur de conflits, nous sommes en mesure de distinguer 4 cas :

1. **Déclarations approuvées** : lorsqu'une relation a seulement été approuvée.
2. **Déclarations contradictoires** : lorsque certains utilisateurs ont proposé une relation et que d'autres utilisateurs ont approuvé une autre relation sur la même paire de tags, par exemple, certains utilisateurs ont approuvé que "pollution" est plus général que "polluant" e, et d'autres utilisateurs ont approuvé que "pollution" est une variante orthographique de "polluant".
3. **Déclarations discutables** : lorsqu'une seule relation est indiquée sur une paire de tags donnée, mais que cette relation a été approuvée par certains utilisateurs et rejetée par d'autres.
4. **Déclarations rejetées** : lorsqu'une relation a seulement été rejetée.

Analyse des résultats : dans la Figure 80, nous montrons les résultats détaillés du résolveur de conflits appliqué sur le dataset auprès des 5 utilisateurs qui ont choisi une relation pour chacune des 94 paires de tags de l'ensemble de données.

Le premier graphique à gauche de la figure (a) illustre la répartition des différents cas de résolution de conflits sur les 94 paires de tags. Nous constatons que près de la moitié des paires (46 %) sont en conflit, c'est-à-dire que dans un cas sur deux, au moins un utilisateur a choisi une relation différente.

Puis, dans le deuxième graphique (b), nous avons examiné la distribution des cas de résolution de conflits pour chaque type de relation sémantique. Puisque plusieurs relations sont énoncées dans le cas conflictuel, nous n'avons gardé dans ce tableau que les relations qui ont été proposées par le résolveur de conflits, c'est-à-dire les relations qui ont été soutenues par une majorité claire ou proposées comme compromis.

Nous voyons dans ce tableau que *seulement* 70 % des relations de type *variante orthographique* (« closeMatch » dans la figure) ont été approuvées par les utilisateurs alors que 30 % ont été proposées par le résolveur de conflits. Si l'on considère le cas des relations *plus général* et *plus spécialisé* (puisque ces deux relations sont l'inverse l'une

de l'autre), on constate qu'elles sont impliquées dans des conflits dans plus de 50% des cas.

Cette tendance au conflit est encore plus forte dans le cas *relié* (« Related » dans la figure) : la relation *relié* n'a jamais été approuvée *uniquement* par les utilisateurs et est soit a) impliquée dans des conflits (48% des relations), soit b) discutable (52% des déclarations). Il convient de noter ici que *relié* a été proposé comme compromis sans avoir été approuvé par aucun utilisateur et a obtenu une majorité claire 3 fois sur les 43 cas de paires avec conflits.

Cela signifie que dans la plupart des cas où les relations sont proposées par le résolveur de conflit, cette relation sert de compromis entre les propositions d'autres relations.

Ainsi, le graphique (b) montre que la *variante orthographique* est la relation la plus apte à apporter un consensus explicite, et il est clair qu'il est plus facile de s'accorder sur le fait que " écologie " et " écologie collective" se réfèrent à la même notion, que de s'entendre sur le fait que " action collective " est plus spécialisée que " collectivité ".

En effet, dans de nombreuses situations, les deux balises tombant dans ce dernier cas de figure peuvent ne pas être reliée et cela dépend fortement du niveau d'expertise des utilisateurs.

Dans le troisième graphique (c), nous avons examiné l'influence d'une autre caractéristique notable qui peut distinguer différents types de paires de tags. Certaines paires de tags se composent d'un mot pour le premier tag et d'un mot composé pour le second, comprenant le premier tag (comme dans "pollution" et "pollution du sol") ou d'un de ses dérivés (comme dans "pollution" et "détection des polluants"), et cela concerne 30 paires de tags sur 94.

Dans ce graphique, nous avons tracé la distribution entre deux types de paires de tags : les paires avec des mots composés et les paires sans mots composés, pour chaque cas de résolution de conflit.

Le résultat montre que les paires en conflit sont des paires avec des mots composés dans la majorité des cas (56 %). De même, seulement 18% des relations approuvées et 14% des déclarations discutables (nous rappelons que dans ce cas, une seule relation est mentionnée, bien qu'elle puisse être approuvée et rejetée) impliquaient des paires avec des mots composés, et ce type de paires n'a jamais été à l'origine de déclarations rejetées.

Cela suggère que les paires avec des mots composés sont plus susceptibles de causer des conflits et mènent rarement à des consensus clairs.

6.2.5.3 Créer un point de vue consensuel

Le quatrième type d'agent que nous avons introduit est le `ReferentUser`. L'utilisateur référent pourra approuver, rejeter ou corriger toutes les relations déjà existantes dans la folksonomie structurée afin de maintenir *son propre point de vue consensuel*.

Le mécanisme de résolution de conflits aidera l'utilisateur référent dans sa tâche en soulignant les conflits qui existent déjà dans la folksonomie structurée de manière

collaborative. Ensuite, toutes les déclarations que l'utilisateur référent a déjà traitées seront ignorées dans d'autres passages du `ConflictSolver`. Le point de vue consensuel peut être utilisé pour générer un nuage de tags hiérarchisé à partir de la folksonomie, où les tags plus généraux sont affichés avec des polices de caractères plus grandes que les tags plus spécialisés. Ce type de nuage de tags peut être utile pour guider l'utilisateur en lui donnant une vue panoramique du contenu de la folksonomie et peut être présenté comme un point de départ de la navigation, en indiquant les tags les plus généraux, puis, pendant la recherche, en donnant l'environnement sémantique du tag courant en montrant des tags de plus en plus généraux et de plus en plus spécialisés (à la manière de la Figure 78).

6.2.5.4 Exploitation et filtrage des points de vue

A ce stade du processus, on obtient une folksonomie sémantique structurée par plusieurs points de vue, parmi lesquels un point de vue global et consensuel émerge.

Nous présentons dans cette section les stratégies que nous proposons pour exploiter ces points de vue afin de présenter une expérience cohérente à tous les utilisateurs du système.

En gardant la trace du type d'agent associé à chaque relation obtenue, nous pouvons donner par exemple priorité à certaines relations suggérées lorsqu'un utilisateur *u* recherche une tag *t*. (ex : priorité aux relations correspondant à « mon point de vue »).

Le système émet 5 requêtes SPARQL pour la recherche de relations associées au tag recherché et chaque fois approuvées par différents types d'utilisateurs mais en s'assurant que ces relations n'entrent pas en conflit avec les résultats précédents.

Tous les résultats seront ensuite fusionnés et utilisés pour suggérer des balises sémantiquement liées à *t*. L'ordre de priorité suivi est donné ci-dessous :

1. Toutes les relations S_u approuvées par l'utilisateur *u*.
2. Toutes les relations S_{ru} approuvées par le `ReferentUser`, sauf si elles entrent en conflit avec une relation de S_u.
3. Les relations S_{cs} approuvées par le `ConflictSolver`, sauf si elles entrent en conflit avec une une relation de S_u ou de S_{ru}.
4. Toutes les relations S_{ou} approuvées par d'autres utilisateurs, sauf si elles entrent en conflit avec une relation de S_u, S_{ru} ou S_{cs}.
5. Toutes les relations S_{tc} approuvées par le `TagStructureComputer`, sauf si elles entrent en conflit avec une relation de S_u, S_{ru}, S_{cs} ou S_{ou}.

Cet ensemble de règles permet, lors de la suggestion de tags à un utilisateur lors d'une recherche, de filtrer les points de vue conflictuels ou plus généraux des autres contributions provenant d'humains ou de machines.

Par exemple, si l'utilisateur recherche le tag "énergie", le système suggérera d'abord des étiquettes provenant d'assertions que l'utilisateur a approuvées, par exemple, si l'utilisateur actuel a approuvé que "énergies" est une *variante orthographique* de "énergie", il suggérera "énergies".

Nous donnons un exemple dans le Listing 1 de la seconde requête qui est émise sur des graphes nommés et qui recherche les relations approuvées par le `ReferentUser` (ligne 1 à 4) et qui (i) ne sont pas rejetées par l'utilisateur courant (ligne 5 à 8) et (ii) ne sont pas en conflit avec celles approuvées par l'utilisateur courant (ligne 9 à 13).

```
1 SELECT * {
2   GRAPH ?g {?search-tag ?p ?suggested-tag}
3   FILTER(?search-tag = <http://ex.org/tag/energy>)
4   ?g rdf:type srtag:ReferentValidatedStatement
5   OPTIONAL {
6     ?u srtag:hasRejected ?g
7     FILTER(?u  = <http://ex.org/users/me>)}
8   FILTER(!bound(?u))
9   OPTIONAL{
10    GRAPH ?g2 {?search-tag ?p2 ?suggested-tag}
11    ?g2 srtag:approvedBy <http://ex.org/users/me>
12    ?p srtag:incompatibleWith ?p2}
13  FILTER (!bound(?g2))}
```

Listing 1 : **Requête SPARQL utilisée pour récupérer les relations concernant la balise "energy" et approuvée par le `ReferentUser` mais pas directement rejetée par l'utilisateur actuel ou contradictoire avec les relations qu'il a approuvées.**

Par exemple, si le `ReferentUser` avait approuvé que "énergies" est *plus général* que "énergie", cette relation ne sera pas incluse dans les résultats puisque, dans l'ontologie SRTag, la propriété `skos:closeMatch` (c'est la propriété que nous utilisons pour les variantes orthographiques) est déclarée être `srtag:incompatibleWith` avec la propriété `skos:broader`. Le système procèdera alors aux requêtes suivantes, en suivant l'ordre de priorité décrit précédemment.

En conséquence, cet algorithme il permet à chaque utilisateur de bénéficier des contributions des autres utilisateurs tout en préservant une expérience cohérente en utilisant un point de vue référent ou, en cas d'absence, en utilisant le point de vue du résolveur de conflits.

6.3 Conclusion et discussion

Le projet ISICIL a réconcilié les applications virales du web social avec les frameworks du web sémantique pour offrir une plate-forme open source gratuite proposant trois fonctionnalités principales : (1) un réseau social sémantique associé à des outils d'analyse de réseaux sociaux, (2) un système d'enrichissement sémantique de folksonomies liées à des terminologies d'entreprise et (3) des interfaces utilisateur sémantiquement augmentées pour les outils de surveillance et de reporting.

Nous avons d'abord proposé une approche semi-automatique pour enrichir sémantiquement les folksonomies. Les utilisateurs conservent leur point de vue personnel tout en bénéficiant des contributions des autres et en construisant collectivement un point de vue consensuel. Le résultat est une folksonomie

sémantiquement enrichie qui supporte de multiples points de vue, dans lesquels les balises sont liées par des relations SKOS.

Pour amorcer le processus, nous utilisons la manipulation automatique des folksonomies pour extraire la sémantique émergente. A cet égard, nous avons proposé une évaluation des principales méthodes basées sur les chaînes de caractères afin de : (a) motiver le choix des métriques les plus performantes dans notre contexte ; et (b) évaluer la capacité de ces métriques à différencier les relations sémantiques généralement utilisées dans les thésaurus, c'est-à-dire être capable de dire quand deux tags sont simplement *reliées*, ou quand un tag est *plus général* ou *plus spécialisé* qu'un autre tag, ou quand deux tags sont des *variantes orthographiques* de la même notion.

En conséquence, nous avons proposé une métrique heuristique qui exécute cette tâche. Cette métrique heuristique est la plus performante pour détecter les variantes orthographiques, comme prévu, mais donne aussi des résultats intéressants pour les relations de subsomption dans des cas tels que la "pollution" qui est plus général que la "pollution du sol".

Nous avons également montré quantitativement que les approches analysant la structure des folksonomies sont nécessaires pour récupérer les relations sémantiques lorsque les balises partageant des relations sémantiques ne sont pas morphologiquement similaires, même si elles sont plus coûteuses et non incrémentales, et nous avons présenté les résultats de ces trois types de méthodes que nous avons obtenus sur un ensemble de données du monde réel.

Afin de capturer les points de vue divergents dans la structuration sémantique des folksonomies, nous avons proposé une ontologie formelle qui utilise des graphes nommés pour décrire les relations sémantiques entre les balises. Les points de vue des utilisateurs sont alors attachés à ces relations affirmées. En décrivant les différentes classes d'agents qui proposent ou rejettent les relations affirmées, nous sommes capables de modéliser un cycle de vie complet pour un enrichissement collaboratif et automatiquement assisté des folksonomies. (1) Ce cycle commence par une folksonomie plate qui est d'abord analysée par des agents automatiques qui proposent des relations sémantiques. (2) Les utilisateurs peuvent apporter et maintenir leur propre point de vue en validant, rejetant ou proposant des relations sémantiques grâce à une interface conviviale intégrée dans un outil de navigation qui entre dans leur utilisation quotidienne. (3) Les conflits émergeant de ces points de vue sont détectés et (4) utilisés pour aider un utilisateur référent à maintenir un point de vue global et consensuel. (5) Le résultat de ce processus est une folksonomie augmentée de relations sémantiques, chacune liée à des points de vue différents coexistant avec un point de vue consensuel. (6) Le cycle redémarre lorsque de nouveaux tags sont ajoutées ou lorsque des relations sont suggérées ou modifiées. Les assertions sémantiques sont utilisées pour suggérer des tags lors de la navigation dans la folksonomie et un ensemble de règles formelles permet de filtrer les assertions sémantiques afin de présenter une expérience cohérente aux utilisateurs tout en leur permettant de bénéficier des contributions des autres.

La prise en charge des points de vue divergents permet aux premiers utilisateurs d'organiser les tags comme ils le souhaitent. Cette caractéristique est importante car les études empiriques sur l'utilisation des tags montrent que les tags sont souvent perçus par les utilisateurs d'abord comme un outil pour organiser leur propre base de connaissances. En permettant à chaque utilisateur de maintenir son point de vue, nous prenons en compte une des caractéristique essentielle et fondamentale de la pratique du tagging social. De plus, notre système comporte plusieurs mécanismes pour construire une structuration consensuelle des tags et permettre à chaque utilisateur de bénéficier des contributions des autres utilisateurs. Par exemple, dans le populaire service de bookmarking social delicious.com, des fonctionnalités similaires permettent aux utilisateurs de regrouper les tags en bundles. Cependant, delicious.com ne fournit aucun mécanisme pour partager ces groupes de tags entre les utilisateurs. Notre approche vise à permettre le partage d'une telle structuration des tags. En les gardant ensemble dans un système cohérent, ces multiples points de vue permettent également de conserver les différents niveaux d'expertise que l'on trouve dans de nombreuses communautés. Dans notre exemple, si nous prenons les tags "pollution" et "polluant", les utilisateurs pourraient ne pas être tous d'accord sur la relation sémantique qui devrait lier ces deux balises et il n'y a probablement pas de relation unique à conserver qui soit valable pour tous les utilisateurs. Dans notre approche, nous laissons aux utilisateurs le choix entre quatre relations sémantiques de type thésaurus, à savoir les relations connexes (deux tags sont juste *reliés*), les relations *plus général* ou *plus spécialisé* et la *variante orthographique*. Certains utilisateurs ayant un haut niveau d'expertise dans le domaine correspondant seront capables de distinguer quelle relation choisir, peut-être en optant pour des relations plus générales ou plus spécialisées, tandis que d'autres utilisateurs moins experts seront simplement disposés à tenir compte du fait qu'il existe une relation entre deux tags, optant pour des relations de type *relié*, ou certains utilisateurs moins experts seront même prêts à fusionner deux tags, optant pour une *variante orthographique*, parce qu'ils ne sont pas trop préoccupés par les distinctions précises qui peuvent être faites.

De plus, la capture de points de vue divergents permet de détecter des tags à significations multiples. En effet, ces tags sont susceptibles d'être liés à des tags très différents par les utilisateurs. Par exemple, le tag "RDF" peut être déclaré plus spécialisé que le tag "web sémantique" pour certains utilisateurs, pour qui il signifie Resource Description Framework, ou il pourra être placée par d'autres utilisateurs comme étant plus spécialisé que la balise "African politics", auquel cas RDF signifie "Rwanda Defense Force". En permettant à chaque utilisateur de maintenir son point de vue, nous lui permettons de se concentrer sur sa propre structuration et compréhension des tags, favorisant ainsi l'émergence des multiples significations.

Notre approche permet à un utilisateur référent de construire une structuration globale de la folksonomie qui se nourrit de toutes ces contributions individuelles. Cette structuration globale peut être exploitée dans la construction d'un thésaurus interne dans lequel les concepts sont définis et structurés plus précisément que les tags / mots-clés. Par exemple, dans notre exemple du tag « RDF », ce tag sera relié à des tags plus généraux (« semantic web » et « Rwanda Defense Force ») dans la vue globale de la folksonomie structurée, ce

qui aidera l'utilisateur référent à réaliser que cette balise peut avoir plusieurs significations qui peuvent être transformées en différents concepts du thésaurus interne. Ainsi, notre approche multipoints de vue peut grandement aider à la construction de représentations de la connaissance plus élaborées (comme les thésaurus par exemple) qui bénéficieront des contributions de tous les membres de la communauté.

Ainsi, notre approche apporte des solutions au problème classique du goulot d'étranglement dans l'acquisition des connaissances, puisque nous donnons une chance à tous les utilisateurs de contribuer à l'élaboration d'une représentation des connaissances partagées.

Enfin, l'expérience que nous avons menée a montré que, même parmi un petit nombre d'utilisateurs, les utilisateurs n'étaient pas tous d'accord sur la relation sémantique qui devrait lier les balises. Le résultat global de cette expérience montre que pour près de la moitié (46 %) des paires de balises de l'ensemble de données, les utilisateurs ont proposé différentes relations sémantiques. Ainsi, l'utilité de la prise en compte de points de vue multiples peut être observée même dans de petits groupes d'utilisateurs.

Nous allons voir dans le chapitre suivant, suite logique du travail présenté ici sur les tags, qu’il est également possible d’exploiter le graphe tripartite de la ressources-tags-utilisateurs pour extraire cette fois-ci des sous communautés d’utilisateurs.

7. Identification de sous communautés en exploitant le réseau socio-sémantique et la folksonomie (2010-2012)

Contexte

Il est possible, en appliquant des algorithmes d'étiquetage et de clustering, d'exploiter les relations utilisateurs-tags pour identifier des sous-communautés (de pratique, d'intérêt, etc.)

Contributions

Sur la base de nos résultats d'analyse de réseaux sociaux sémantiques, nous avons également présenté un algorithme de détection de communautés, SemTagP, qui tire parti des données sémantiques capturées lors de la structuration des graphes RDF des réseaux sociaux. SemTagP propose non seulement de détecter mais aussi d'étiqueter les communautés en exploitant (en plus de la structure du graphe social) les tags utilisés par les personnes pendant le processus de marquage social ainsi que les relations sémantiques déduites entre ces balises. Ce faisant, nous avons été en mesure d'affiner le partitionnement du graphe social avec un traitement sémantique et d'étiqueter l'activité des communautés détectées. Nous avons testé et évalué cet algorithme sur le réseau social construit à partir de la liste des thèses de doctorat financées par l'ADEME, l'Agence française de l'environnement et de la maîtrise de l'énergie, qui contiennent de nombreux tags du domaine de l'énergie ou reliés. Nous avons montré comment cette approche nous permet de détecter et d'étiqueter les communautés d'intérêt et de contrôler la précision des étiquetages.

Collaboration, encadrement

La détection de communautés est un des résultats conjoint des deux thèses que j'ai co-encadrées, déjà citées : celle de Guillaume Ereteo sur l'analyse des réseaux sociaux-sémantiques et celle de Freddy Limpens sur les folksonomies.

Publications principales ayant résulté de ces travaux

Michel Buffa, Nicolas Delaforge, Guillaume Ereteo, Fabien Gandon, Alain Giboin and Freddy Limpens, *"ISICIL: Semantics and Social Networks for Business Intelligence"*, conference SOFSEM 2013, 39th International Conference on Current Trends in Theory and Practice of Computer Science. January 26–31, 2013 Špindlerův Mlýn, Czech Republic.

Guillaume Erétéo, Fabien Gandon, and Michel Buffa, *SemTagP: Semantic Community Detection in Folksonomies*, IEEE/WIC/ACM International Conference on Web Intelligence, August 2011, Lyon.

Guillaume Erétéo, Michel Buffa, Olivier Corby, Fabien Gandon, Mylène Leitzelman, Freddy Limpens, and Peter Sander. Handbook of Research on Methods and Techniques for Studying Virtual Communities, chapter *Semantic Social Network Analysis, a Concrete Case*. 2011, IGI Global.

Résumé de cette section

La détection de communautés aide à comprendre la répartition des acteurs et des activités dans un réseau social. De nombreuses tâches peuvent bénéficier de l'identification de communautés d'intérêts, par ex. dans le domaine de l'intelligence économique, de la création d'équipes de projet, de la veille technologique, du conseil, pour des notifications ciblées dans les systèmes d'information, etc. Les algorithmes qui abordent ce problème sont soit hiérarchiques, soit basés sur des heuristiques [32]. Les algorithmes hiérarchiques produisent un arbre de partitions en divisant itérativement le réseau en sous-communautés (approche *top-down*) ou en fusionnant les communautés en plus grandes (approche *bottom-up*). Les algorithmes basés sur des heuristiques, par exemple ceux utilisant le principe de marche aléatoire ou exploitant les analogies avec les réseaux électriques, exploitent les caractéristiques du réseau pour déterminer les groupes de nœuds fortement connectés. Parmi les algorithmes basés sur l'heuristique, on en trouve aussi basés sur la propagation d'étiquettes (labels). L'un des algorithmes les plus rapides proposés à l'époque était l'algorithme de propagation d'étiquettes de Raghavan et al. [31], que nous appellerons l'algorithme RAK. Il permettait de détecter les communautés en propageant des étiquettes dans le réseau social comme suit : (1) l'algorithme attribue une étiquette aléatoire unique à chaque nœud, chaque nœud *n* remplace son étiquette par l'étiquette la plus utilisée par ses nœuds adjacents dans le graphe, si son propre label est différent. Dans le cas où plusieurs étiquettes sont les plus utilisées, une étiquette est choisie au hasard. (3) Si au moins un nœud a changé d'étiquette, on revient à l'étape 2. (4) Sinon, les nœuds qui partagent une même étiquette forment une communauté.

La figure suivante présente cet algorithme sur un exemple jouet.

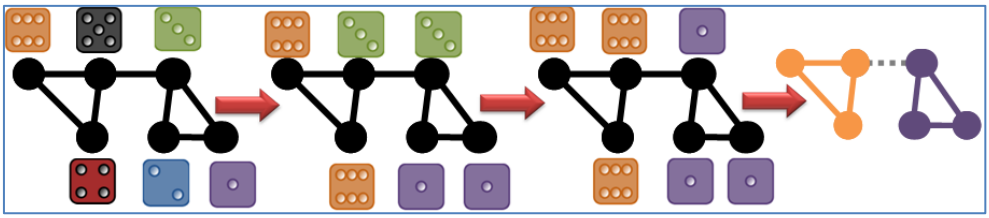


Figure 81. Exemple de propagation d'étiquettes

Les applications Web sociales ont popularisé le marquage social (*social tagging*) : les utilisateurs catégorisent les ressources (par exemple, les médias, les billets de blog, etc.) avec des mots-clés librement choisis, appelés tags. Ce processus génère une folksonomie : un ensemble d'acteurs décrivant un ensemble d'objets avec un ensemble de balises. Un travail pionnier de Peter Mika [18] a étudié les folksonomies comme des ontologies légères émergeant des usages des communautés. Chaque tag peut représenter une communauté d'intérêt composée de tous les acteurs utilisant cette balise. Les tags permettent aux utilisateurs de classer facilement les ressources en ligne pour leur usage personnel ou pour des communautés ciblées, et de rejoindre librement les interactions en ligne. Les balises partagées par plusieurs utilisateurs forment une nouvelle source de liens entre les utilisateurs : "*l'interaction produit une similarité, alors que la similarité produit une interaction*" dit P.Mika. Par exemple, pendant les élections en Iran, les gens ont

surmonté la censure des médias avec le réseau social Twitter en annotant leurs messages avec le même tag, `#iranelection`, afin d'interagir, recueillir et partager leurs informations sur ces élections. Les balises-tags ont permis de lier les utilisateurs et d'étiqueter leur communauté émergente : "*ceux qui s'intéressent aux élections en Iran*". Dans [33], les auteurs ont amélioré la détection de communauté en appliquant un algorithme de clustering à un graphe traitant à la fois les tags et les ressources marquées avec ces tags.

Par ailleurs, nous avons vu dans la section précédente que certains tags peuvent être liés sémantiquement (hyponymes, synonymes, plus général/spécialisé, variante orthographique etc.) et qu'un ensemble de tags liés peut également être vu comme un vocabulaire partagé par les membres d'une même communauté. Nous avons défini en 2009-2010 une méthode originale pour combiner le traitement automatique et les contributions manuelles des utilisateurs pour aider les communautés en ligne à enrichir sémantiquement les folksonomies et à structurer leur propre vocabulaire. Une fois que les folksonomies sont typées et structurées, les relations entre les tags et entre les tags et les utilisateurs fournissent une nouvelle source de réseaux d'affiliation, ce qui nous permet d'affiner le processus d'étiquetage des communautés.

Notre principale contribution dans le domaine de la détection de communautés dans un réseau social sémantiquement annoté, a consisté à fusionner les trois approches présentées (algorithme RAK, étiquetage à base de tags et structuration de la folksonomie).

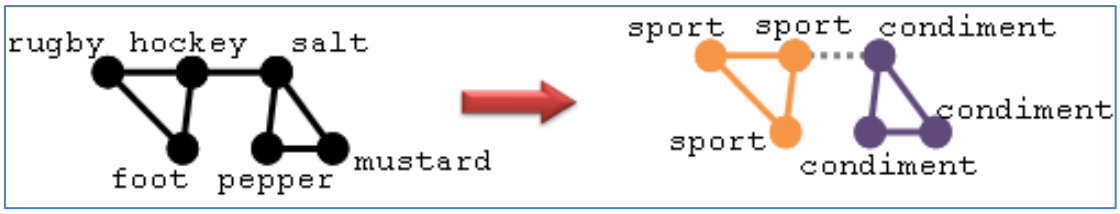


Figure 82. Propagation sémantique d'étiquettes.

Ainsi, nous avons amélioré l'état de l'art de l'époque en réalisant des détections de communautés qui tirent parti non seulement de la structure de liens du réseau social, mais aussi de la sémantique émergente des folksonomies. Nous avons appelé l'algorithme principal SemTagP, un algorithme qui transforme la propagation de l'étiquette aléatoire RAK en une propagation de tags sémantiques afin de détecter les communautés et de les étiqueter de manière intelligible. Nous présenterons en détail cet algorithme et nous le comparerons avec l'état de l'art qui lui est antérieur.

Par la suite nous verrons comment nous avons implémenté cet algorithme avec des frameworks du web sémantique afin de tirer profit des primitives ontologiques utilisées pour typer les graphes RDF. Enfin, nous présenterons les résultats que nous avions obtenus à l'époque avec le réseau social construit à partir des thèses de doctorat financées par l'ADEME, l'Agence française de l'environnement et de la maîtrise de l'énergie. Il comprenait 1853 agents, 1597 superviseurs académiques et 256 ingénieurs de l'ADEME, 13982 relations entre personnes et 6583 tags.

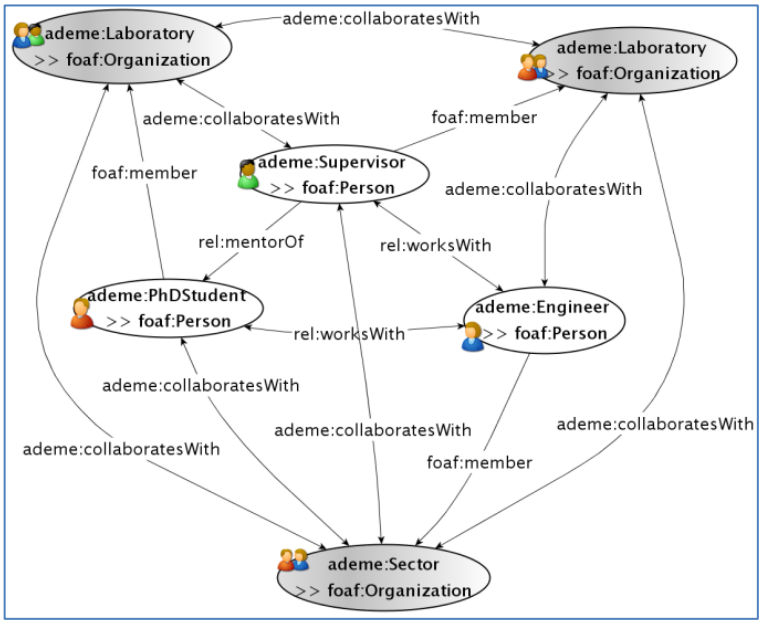


Figure 83. Réseau social des thèses financées par l'ADEME

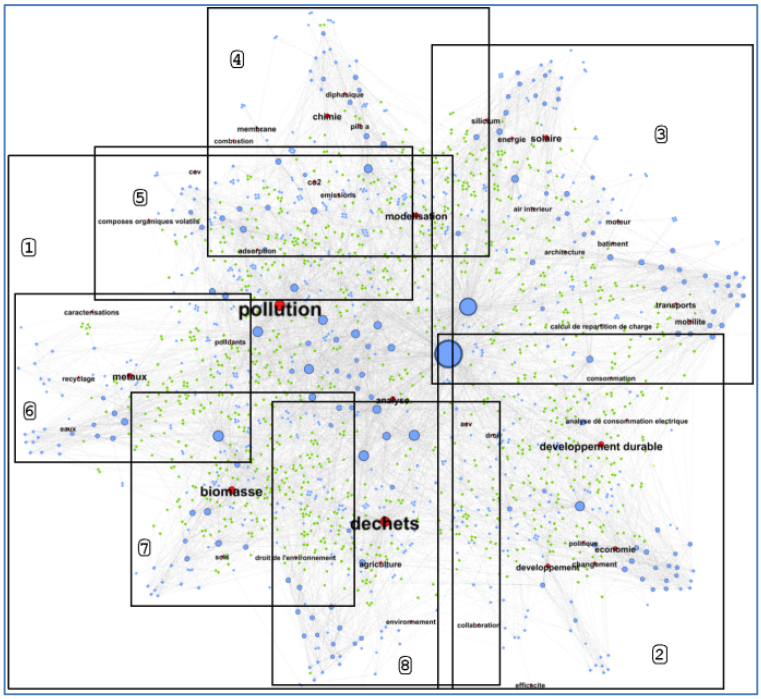


Figure 84. Réseau social des thèses financées par l'ADEME avec les tags identifiant les communautés obtenues avec l'algorithme SemTagP. Les nœuds rouges, bleus et verts sont respectivement les tags, les ingénieurs de l'ADEME et les superviseurs académiques. Les zones encadrées contiennent des communautés

7.1 SemTagP : détection sémantique des communautés dans les folksonomies.

En nous appuyant sur nos résultats de l'analyse sémantique des réseaux sociaux [150], nous présentons un algorithme de détection de communautés, SemTagP, qui tire profit

des données sémantiques qui ont été capturées tout en structurant le graphe RDF du réseau social.

SemTagP offre non seulement de détecter mais aussi d'étiqueter (labelliser, identifier…) les communautés en exploitant la structure tripartite du graphe social mais aussi les tags utilisés par les personnes pendant le processus de tagging social ainsi que les relations sémantiques déduites entre ces tags (avec les méthodes présentées dans le chapitre précédent de ce manuscrit).

Ainsi, nous sommes en mesure d'affiner le partitionnement du graphe social par un traitement sémantique nous proposons d'étiqueter l'activité des communautés détectées.

Nous avons testé et évalué cet algorithme sur le réseau social construit à partir de thèses de doctorat financées par l'ADEME et avons montré comment cette approche nous a permis de détecter et d'étiqueter les communautés d'intérêt, en contrôlant la précision des de cet étiquetage.

7.1.1 Introduction et travaux connexes

La détection de communautés aide à comprendre la répartition des acteurs et des activités. De nombreuses tâches peuvent bénéficier de l'identification de communautés d'intérêts, comme la veille économique (*business intelligence*), la création d'équipes de projet, la veille technologique, le consulting, les notifications ciblées dans les systèmes d'information, etc. Les algorithmes qui s'attaquent à ce problème sont soit hiérarchiques, soit basés sur l'heuristique [10].

Les algorithmes hiérarchiques produisent un arbre de partitions en divisant itérativement le réseau en sous-communautés (*top-down*) ou en fusionnant des communautés en communautés plus grandes (*bottom-up*).

Les algorithmes basés sur l'heuristique, par exemple la marche aléatoire (*random walk*) ou exploitant les analogies avec les réseaux électriques pour déterminer des groupes de nœuds densément connectés. Parmi les algorithmes heuristiques, celui de propagation de labels [31] (également connu sous le nom de RAK) propose de détecter les communautés en propageant les labels dans le réseau social comme suit :

1. L'algorithme attribue une étiquette aléatoire unique à chaque nœud,
2. Chaque nœud *n* remplace son étiquette par l'étiquette la plus utilisée par ses nœuds adjacents dans le graphique, si sa propre étiquette est différente. Dans le cas où plusieurs étiquettes sont les plus utilisées, l'une est choisie au hasard.
3. Si au moins un nœud a changé d'étiquette, passer à l'étape 2.
4. Sinon, les nœuds qui partagent le même label forment une communauté.

La Figure 85 présente cet algorithme sur un exemple jouet.

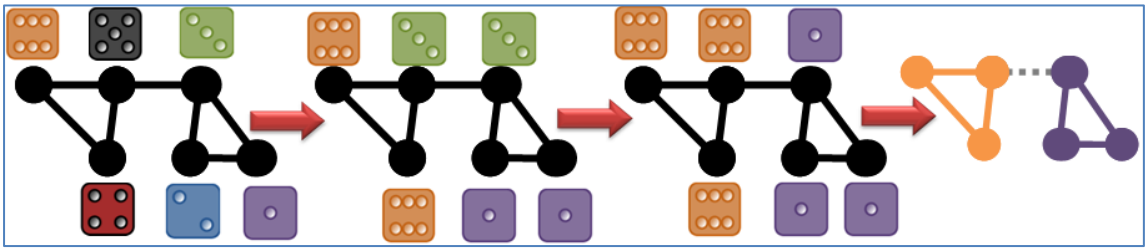


Figure 85. Exemple de propagation d'étiquettes.

Les applications du Web social ont rendu le tagging social populaire : les utilisateurs catégorisent les ressources (par exemple les médias, les articles de blog, etc.) avec des mots-clés librement choisis appelés *tags*. Ce processus génère une folksonomie : un ensemble d'acteurs décrivant un ensemble d'objets avec un ensemble de balises. Un travail pionnier de Peter Mika [133] a étudié les folksonomies en tant qu'ontologies légères émergeant des usages des communautés.

Chaque tag peut représenter une communauté d'intérêt composée de tous les acteurs utilisant ce tag.

Les tags permettent aux gens de classer facilement les ressources en ligne pour leur usage personnel ou pour des communautés ciblées, et de participer librement aux interactions en ligne. Les tags partagés par plusieurs utilisateurs forment une nouvelle source de liens entre les utilisateurs : P. Mika dit "*L'interaction produit la similitude, tandis que la similitude produit l'interaction*" [133]. Par exemple, pendant les élections en Iran, les gens ont surmonté la censure des médias avec le réseau social Twitter en annotant leurs messages avec le même tag, `#iranelection`, afin d'interagir et de rassembler leurs informations. Les tags permettent de lier les utilisateurs et d'étiqueter leur communauté émergente. Dans [33], les auteurs améliorent la détection des communautés en appliquant un algorithme de clustering à un graphe considérant *de manière égale* les tags et les ressources.

Certains tags sont sémantiquement liés (hyponymes, variantes orthographiques, etc.) et un ensemble de tags liés peut également être considéré comme un vocabulaire partagé par les membres d'une communauté.

Comme nous l'avons montré dans le chapitre précédent de ce manuscrit, nous avons défini une méthode pour combiner le traitement automatique et les contributions manuelles des utilisateurs afin d'aider les communautés en ligne à enrichir sémantiquement les folksonomies et à structurer leur propre vocabulaire. Une fois les folksonomies typées et structurées, les relations existant entre tags et entre les tags et les utilisateurs fournissent une nouvelle source de réseaux d'affiliation, ce qui nous permet, nous allons le voir dans ce chapitre, d'affiner le processus d'étiquetage des communautés.

Nous proposons de fusionner ces trois approches (RAK, étiquetage par tag et structuration de la folksonomie) afin d'effectuer des détections de communauté qui tirent profit non seulement de la structure de liens du réseau social, mais aussi de la sémantique émergente des folksonomies.

Nous introduisons d'abord SemTagP, un algorithme qui transforme la méthode de propagation des étiquettes aléatoires RAK en une propagation sémantique afin de détecter les communautés et de les étiqueter de manière significative.

Ensuite, nous présentons comment nous avons implémenté cet algorithme avec des frameworks du web sémantique afin de tirer profit des primitives ontologiques utilisées pour typer les graphes RDF.

Enfin, nous présentons le résultat que nous avons obtenu avec un réseau social réel construit à partir de thèses de doctorat financées par l'ADEME, l'Agence de l'environnement et de la maîtrise de l'énergie.

7.1.2 SemTagP : Propagation d'étiquettes sémantiques

SemTagP est un algorithme conçu pour détecter et caractériser les communautés à partir du graphique typé dirigé formé par les descriptions RDF des réseaux (sociaux) et des folksonomies.

En utilisant les ontologies existantes pour représenter les réseaux sociaux en ligne [10], nous pouvons lier et typer les réseaux sociaux en ligne, associer leurs acteurs aux tags et relier sémantiquement les tags les uns aux autres.

SemTagP (Figure 86) est une extension de l'algorithme RAK qui transforme la propagation des étiquettes en une propagation sémantique des tags : au lieu d'assigner et de propager des étiquettes aléatoires, nous assignons aux acteurs les tags qu'ils utilisent et nous les propageons en utilisant des relations de généralisation entre les tags (par exemple `skos:narrower` / `skos:broader`) pour fusionner des communautés spécialisées et généraliser leurs étiquettes à des hyperonymes communs (exemple avec la figure : on fusionne « rugby » et « hockey » en « sport », « salt », « pepper » et « mustard » en condiment)

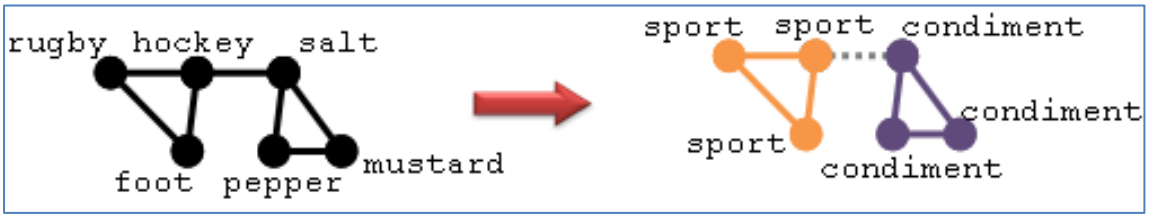


Figure 86. Propagation sémantique des étiquettes.

Nous utilisons le calcul de la modularité dirigée sur les graphes RDF [151] pour évaluer la qualité de la partition communautaire obtenue après chaque boucle de propagation.

Lorsqu'un réseau partitionné a une grande modularité, cela signifie qu'il y a plus de connexions entre les nœuds de chaque communauté qu'entre les nœuds de différentes communautés. Plus précisément, la modularité mesure la fraction des arrêtes au sein des graphes représentant les communautés du réseau moins la valeur attendue de la même quantité dans un réseau avec la même partition communautaire mais avec des connexions aléatoires entre les nœuds [152] (la randomisation des connexions préserve le degré des nœuds).

La modularité d'un réseau dirigé est définie dans [151] comme suit :

Définition 1, modularité dirigée : Soit m le nombre d'arcs du réseau, A_{ij} le nombre d'arcs entre i et j, c_i la communauté de i, $\delta(c_i, c_j) = 1$ si $c_i = c_j$, 0 sinon $d^{in}_{<i>}$, et $d^{out}_{<i>}$ les degrés entrants et sortants du sommet i, la modularité dirigée est :

$$Q = \frac{1}{m} \sum_{i,j \in V} [A_{ij} - \frac{d^{out}_{<i>} d^{in}_{<j>}}{m}] \delta(c_i, c_j)$$

SemTagP propage itérativement les tags dans le réseau afin d'obtenir un nouveau partitionnement : les nœuds qui partagent le même tag forment une communauté.

Lors d'une boucle de propagation, chaque acteur choisit le tag le plus utilisé parmi ses voisins, pour un tag t nous comptons 1 occurrence pour chaque voisin utilisant t et 1 occurrence pour chaque voisin utilisant une tag avec une relation `skos:narrower` de t (un tag plus spécialisé). Nous itérons jusqu'à ce que la modularité cesse d'augmenter. L'avant-dernier réseau partitionné est la sortie de l'algorithme.

Dans nos résultats précédents sur l'analyse sémantique des réseaux sociaux (chapitre 5, voir aussi [150]), nous avons souligné l'importance de considérer la diversité et la sémantique des liens entre acteurs. Propager des tags à travers différents types de relations, notamment dans différents sous-réseaux, pourrait produire différentes partitions communautaires. Par conséquent, SemTagP est paramétré par le type de la relation existant entre les tags. Nous formalisons SemTagP comme suit :

Algorithme `SemTagP(RDFGraph network, Type relation)`

```
1. DO
2.    old_network = network
3.    //propagate tags (i.e. compute new partitions)
4.    FOR user in network.users
5.       user.tag=mostUsedNeighborTag(user, relation)
6.     END
7. WHILE mod(network) > mod(oldNetwork)
8. RETURN old_network
```

Algorithme `mostUsedNeighborTags(User user, Type relation)`

```
1. resultTag = null; max = 0
2. tagTable = new hashTable()
3. FOR agent in user.neigbors[relation]
4.   IF tagTable.exists(agent.tag)
5.     tagTable[agent.tag] ++
6.   ELSE
7.     tagTable[agent.tag] = 1
8.   IF(max < tagTable[agent.tag]){
9.     resultTag = agent.tag;
10.     max = tagTable[agent.tag]
11.   FOR broadTag in agent.tag.broaders
12.     IF tagTable.exists(broadTag)
13.       tagTable[broadTag] ++
14.     ELSE
15.       tagTable[broadTag] = 1
16.     IF max < tagTable[broadTag]
17.       resultTag = broadTag;
18.       max = tagTable[broadTag]
19.   END
20. END
21. RETURN resultTag
```

Dans notre première expérimentation, nous avons constaté que certains tags avec de nombreuses relations `skos:narrower` ont absorbé trop de tags pendant la phase de propagation, comme le tag « environnement », qui est omniprésent dans le corpus de l'agence ADEME. De tels tags regroupaient les acteurs dans de très grandes communautés (presque tous les utilisateurs avaient utilisé « environnement »).

Par conséquent, nous avons ajouté une option pour affiner manuellement les résultats : après la première boucle de propagation, nous présentons la partition et l'étiquetage de la communauté actuelle à un utilisateur qui peut rejeter l'utilisation de tags typés `skos:narrower` produisant des communautés trop grandes.

Ensuite, nous redémarrons l'algorithme et répétons ce processus jusqu'à ce qu'aucune autre relation ne soit rejetée, avant de compléter l'algorithme décrit précédemment.

Par exemple, lors du partitionnement d'un réseau social avec des tags liées à des sujets tournant autour du web, l'utilisateur peut rejeter certaines relations `skos:narrower` telles que :

web `skos:narrower` *semantic web*

…s'il souhaite révéler la communauté du « web sémantique », et ne pas la fusionner avrc la communauté « web ».

Nous avons formalisé ici notre algorithme. Nous allons maintenant voir comment nous avons implémenté cet algorithme avec le moteur de graphes sémantiques KGRAM [153] qui supporte le langage de requête SPARQL 1.1.

Nous déléguons tout le traitement sémantique à ce moteur, en tirant parti au maximum des requêtes SPARQL pour exploiter les relations sémantiques entre les tags[98].

7.1.3 Mise en œuvre de SemTagP avec des frameworks du Web sémantique

7.1.3.1 RDF pour représenter les graphes sociaux

Nous avons montré dans [154] et [106] que RDF est bien adapté à la représentation des graphes sociaux et que ce format facilite le partage et l'interopérabilité des données sociales entre applications. RDF tire parti du typage des nœuds et des arrêtes avec les primitives des ontologies que nous avons créées spécialement pour représenter des réseaux sociaux ainsi que des ontologies de domaines spécifiques si nécessaire. Avec ce typage riche, les moteurs sémantiques sont capables d'effectuer des inférences de type à partir de schémas de données pour enrichir automatiquement le graphe et vérifier sa cohérence.

Nous utilisons l'ontologie FOAF pour décrire les personnes, leurs relations et leurs activités. Les propriétés définies dans l'ontologie RELATIONSHIP nous permettent de typer plus précisément les relations entre les personnes (par exemple, la relation `rel:livesWith` se spécialise dans la relation `foaf:knows`). Nous utilisons les primitives de l'ontologie SIOC pour modéliser les comptes utilisateurs en ligne, le contenu qu'ils publient et les interactions médiatisées correspondantes. Enfin, SCOT nous permet de décrire les activités de tagging social tandis que SKOS offre un moyen d'organiser les tags avec des propriétés sémantiques légères comme `skos:narrower`, `skos:closeMatch`, `skos:related`).

Nous inférons la sémantique entre les tags, en utilisant le cycle de vie complet proposé dans le chapitre 6, afin de révéler trois principaux types de relations entre les balises : `skos:related` (tags simplement *reliés*), `skos:closeMatch` (un tag est une *variante orthograp*hique d'un autre) et `skos:narrower` (un tag est *plus spécialisé* qu'un autre). Nous n'utilisons pas `skos:broader` dans l'algorithme SemTagP (son inverse nous suffit).

La Figure 87, présente un exemple jouet d'un graphe social RDF.

[98] Remarquez que l'implémentation du mécanisme de *pattern matching* SPARQL de KGRAM est basé sur l'homomorphisme de graphe qui est un problème NP complet. Cependant, de nombreuses optimisations nous permettent de réduire significativement le temps de calcul de l'interrogation du graphe RDF.

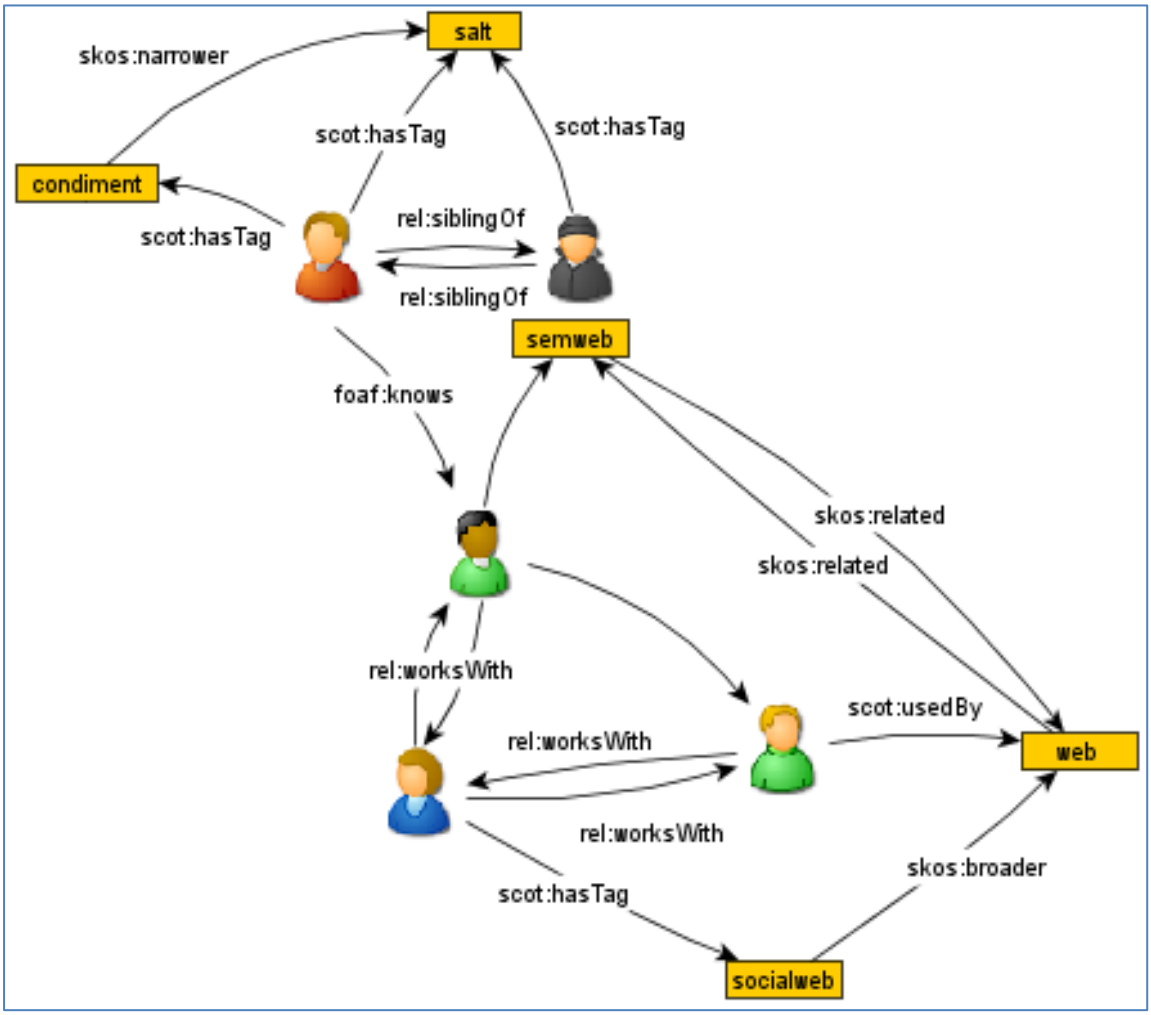


Figure 87. Exemple de graphe social RDF

Nous décrivons dans la section suivante la façon dont nous utilisons la folksonomie structurée résultante pour propager les tags, en tirant profit des graphes typés RDF et des requêtes SPARQL pour faciliter l'implémentation des différentes étapes requises par l'algorithme.

7.1.3.2 SPARQL pour propager les tags !

L'étape de propagation consiste à assigner de façon itérative à chaque acteur le tag le plus fréquent parmi les acteurs auxquels il est lié. Afin de considérer les relations de généralisation entre les tags, nous renforçons le score d'un tag avec le score de ses tags plus spécialisés (reliés par la relation `skos:narrower`). Par exemple, nous exploitons l'énoncé sémantique :

energy `skos:narrower` *renewable energy*

…en comptant une occurrence supplémentaire de « energy » pour chaque occurrence du tag « renewable energy ».

Nous commençons chaque boucle par une requête qui extrait pour chaque acteur les tags de ses voisins (pour une relation paramétrée donnée), leurs tags plus larges, et nous classons les résultats par acteurs et tags :

```
1. select ?user ?tag ?y where {
2.  ?user param[rel] ?neighbor
3.  {{?neighbour scot:hasTag ?tag }
4.    UNION
5.    {?neighbour scot:hasTag ?tag2
6.     ?tag skos:narrower ?tag2
7.     filter(exists{?x scot:hasTag ?tag})}}
8. } order by ?user  ?tag
```

Différentes parties de la fonction `mostUsedNeighboursTags()` décrite ci-dessus sont encodées dans cette requête :

- La ligne 3 code la sélection d'un tag des voisins d'un utilisateur.
- Les lignes 5 à 7 codent la sélection d'un tag plus général que le tag du voisin d'un utilisateur.
- La ligne 8 ordonne les tag résultats par utilisateur pour faciliter le post-traitement.

Après l'achèvement de cette requête, nous effectuons un post-traitement sur le résultat et remplaçons le tag de chaque acteur par le tag le mieux classé parmi ses voisins.

Afin de gérer le rejet d'une généralisation entre deux tags, nous ajoutons une clause de filtrage dans le deuxième bloc de la clause UNION (lignes 5 à 7) pour exclure l'utilisation d'un tag plus général spécifiée,

Par exemple :

```
filter(?tag != <http://ademe.fr/energie>)
```

Notez que la relation analysée est paramétrée et peut être remplacée par n'importe quel type de relation définie dans le graphe RDF (par exemple `sioc:follows`, `rel:worksWith`, `foaf:member`).

7.1.3.3 Modularité d'un graphe RDF

Les triplets d'une description RDF forment un graphe étiqueté dirigé qui peut être vu comme les arcs étiquetés d'un graphe Entité-Relation [153], défini comme suit :

Définition d'un ERGraph : Un ERGraph relatif à un ensemble d'étiquettes L est un 4-tuple $G=(E_G, R_G, n_G, , l_G)$ où :

- E_G et R_G sont deux ensembles finis disjoints, respectivement, de nœuds et de relations.
- n_G: $R_G \rightarrow E_G$ * associe à chaque relation $r \in R_G$ un couple d'entités $e_i,e_j \in E_G$ appelé *les arguments de la relation*. Si $n_G(r)=(e_1,e_2)$ on note $n_G^i(r)=e_i$ le $i^{ème}$ argument de r.
- l_G: $E_G \cup R_G \rightarrow L$ est une fonction d'étiquetage des entités et des relations.

Ainsi, nous définissons la modularité d'un graphe Entité-Relation comme suit :

Définition 2, modularité d'un ERGraph : la modularité d'un graphe Entité-Relation $G = (E_G, R_G, n_G, l_G)$ relative à un ensemble de labels L, pour un label donné de relation $p \in L$, est :

$$Q(G,p) = \frac{1}{\left|R_G^p\right|} \sum_{i,j \in E_G} [A_{ij}^p - \frac{d_{<p,i>}^{out}(G) d_{<p,j>}^{in}(G)}{\left|R_G^p\right|}] \delta(c_i, c_j)$$

Où :

- $R_G^p = \{r \in R_G ; l_G(r) = p\}$
- $A_{i,j}^p = 1$ if $\exists r \in R_G^p ; n_G^1(r) = i$ et $n_G^2(r) = j$, 0 sinon.
- $d_{<p,i>}^{in}(G)$ et $d_{<p,i>}^{out}(G)$ sont respectivement le nombre de relations $r^{in}, r^{out} \in R_G^p ; n_G^2(r^{in}) = i$ et $n_G^1(r^{out}) = i$ les degrés entrants et sortants de i pour la relation labélisée par p.

Nous implémentons cette définition de la modularité en interrogeant le graphe RDF à l'aide de requêtes SPARQL qui calculent différentes parties de cette formule.

Dans [106], nous avons défini des requêtes pour récupérer différentes métriques de réseau qui nous permettent de calculer R_G^p, $d_{<p,i>}^{in}(G)$ et $d_{<p,i>}^{out}(G)$.

Tout d'abord nous calculons R_G^p avec une requête qui récupère simplement le nombre de paires de ressources RDF qui sont liées par la propriété p. Ensuite nous récupérons les degrés d'entrée et de sortie de toutes les ressources RDF liées par une propriété p, avec deux requêtes qui calculent $d_{<p,i>}^{in}(G)$ et $d_{<p,i>}^{out}(G)$ pour chaque valeur possible de i. Enfin, nous calculons la formule en itérant sur les résultats des deux requêtes ci-dessous.

La requête suivante récupère toutes les paires de ressources connectées appartenant à la même communauté pour la propriété donnée en paramètre :

```
1. select ?user1 ?user2 ?tag where{
2.    ?user1 param[property] ?user2
3.    ?user1 scot:hasTag ?tag
4.    ?user2 scot:hasTag ?tag
5.  } group by ?user1 ?user2 ?tag
```

La requête suivante récupère toutes les paires de ressources déconnectées appartenant à la même communauté pour la propriété donnée en paramètre :

```
1. select ?user1 ?user2 ?tag where{
2.    ?user1 scot:hasTag ?tag
3.    ?user2 scot:hasTag ?tag
4.    filter(?user1 != ?user2)
5.    filter(not exists{?user1    param[property]
   ?user2})
6.  } group by ?user1 ?user2 ?tag
```

Nous effectuons ensuite un post-traitement sur les résultats des requêtes ci-dessus pour calculer la modularité de la partition communautaire correspondante.

7.1.4 Expérimentations et résultats

Afin de valider les bénéfices de notre approche, nous avons appliqué notre algorithme sur un ensemble de données des thèses de doctorat financées par l'ADEME : le dataset *thèsenet*. Les thèses de doctorat ont été classées à l'aide de tags et impliquent plusieurs acteurs qui forment un réseau social composé d'employés de l'ADEME et de chercheurs universitaires qui collaborent aux thèses financées. Les agents académiques sont les étudiants au doctorat, les directeurs de thèse et les laboratoires et instituts auxquels ils appartiennent. Du côté de l'ADEME, chaque thèse est suivie par un ingénieur et rattachée à une organisation interne appelée "secteur". Des tags libres sont utilisées pour étiqueter les thèses, à des fins de classification. A partir de cet ensemble de données, nous avons extrait un graphe RDF (qui comprend à la fois la folksonomie et une description du réseau social), puis nous avons appliqué notre algorithme afin de comprendre la structure de la communauté et les activités des différents acteurs, étiquetés avec les balises qui ont été utilisées.

7.1.4.1 Dataset

L'ensemble de données de l'ADEME que nous avons analysé a été fourni sous forme de base de données relationnelle et nous avons utilisé la méthode présentée dans [154] pour construire les descriptions RDF correspondantes. La Figure 88 montre un schéma des concepts que nous avons utilisés pour représenter le réseau doctoral de l'ADEME avec les ontologies décrites dans [10] et une ontologie du domaine de l'ADEME que nous avons conçue pour cette analyse. Les personnes (ingénieurs, étudiants et superviseurs) sont déclarées comme instances de `foaf:Person` et le laboratoire et les « secteurs » comme instances de `foaf:Organization`. L'appartenance d'une personne à une organisation est décrite avec la propriété `foaf:memberOf`. Un étudiant est relié à son superviseur par la propriété `rel:mentorOf` et à sa thèse par la propriété `dc:creator`. Nous avons créé la propriété `ademe:follows`, pour lier un ingénieur de l'ADEME à une thèse de doctorat qu'il suit. Enfin, nous avons généré une URI pour

chaque tag utilisée pour décrire une thèse de doctorat et nous avons utilisé la propriété `scot:hasTag` pour lier une thèse à ses tags.

La Figure 88 décrit la manière dont nous avons enrichi les descriptions RDF des thèses de doctorat de l'ADEME afin de révéler et de structurer le réseau social correspondant. Nous avons mis en relation deux personnes travaillant sur le même doctorat avec la propriété `rel:worksWith`. Nous avons spécifiquement défini la propriété `ademe:collaboratesWith` pour lier deux agents (`foaf:Person` ou `foaf:Organization`) impliqués dans la même thèse. Deux ingénieurs du même « secteur » sont liés par une propriété `rel:colleagueOf`. Nous avons structuré ces liens sociaux en déclarant la propriété `rel:worksWith` comme sous-propriété d'`ademe:collaboreWith`. Enfin, nous avons attaché les tags d'une thèse à tous les acteurs impliqués avec la propriété `scot:hasTag`, produisant une folksonomie avec des agents associant les tags à la thèse. Nous enrichissons également sémantiquement cette folksonomie avec les relations `skos:narrower`, `skos:closeMatch`, `skos:related` calculées selon les méthodes du chapitre 6.

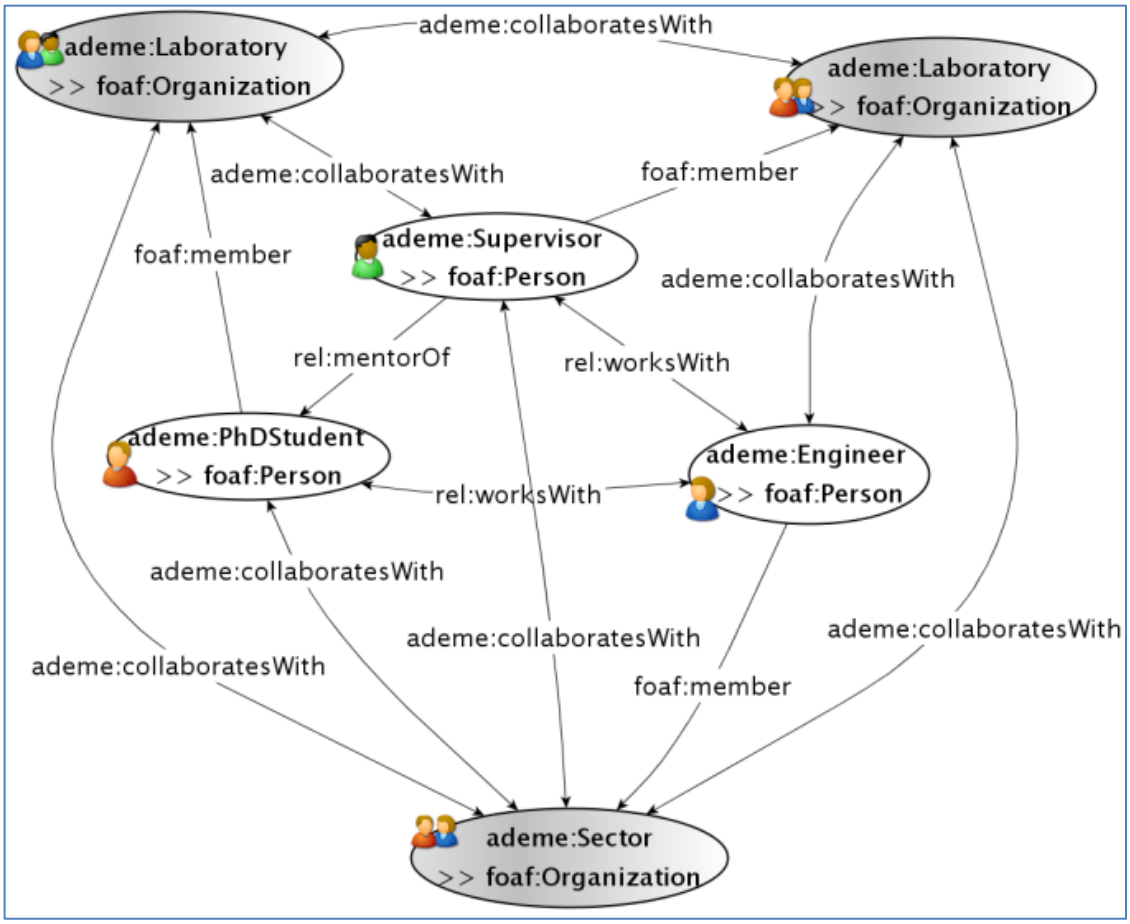


Figure 88. Réseau de financement des thèses de l'ADEME.

7.1.4.2 Expérimentation

Nous avons concentré notre expérience sur le sous-réseau de relations entre les directeurs de thèse et les ingénieurs de l'ADEME, qui sont les acteurs les plus actifs de ce réseau. En utilisant la méthode d'analyse sémantique des réseaux sociaux que nous avons

détaillée dans le chapitre 5, nous avons mesuré les caractéristiques de cet ensemble de données :

- 1 853 agents avec 1 597 superviseurs académiques et 256 ingénieurs de l'ADEME.
- 13 982 relations avec 10 246 `rel:worksWith` des relations entre les ingénieurs de l'ADEME et les superviseurs académiques, et 3 736 relations `rel:colleagueOf` entre ingénieurs de l'ADEME.
- 6 583 tags, avec 3 570 `skos:narrower` entre 2 785 tags (formant un arbre de de profondeur 3).

Ce réseau est un graphe connecté qui a un diamètre de 8, une faible densité (0.004) et un faible coefficient de regroupement (0.031). Ce réseau est fortement centralisé autour des 256 ingénieurs qui partagent un total de 8859 relations alors que les 1 597 acteurs académiques ont un total de seulement 5 123 relations. En effet, les ingénieurs suivent plusieurs thèses de doctorat simultanément et ont des collègues au sein de l'ADEME tandis que les acteurs académiques les plus actifs supervisent un maximum de 14 doctorants.

Afin d'évaluer les avantages de l'introduction de la sémantique dans la propagation des étiquettes, nous avons comparé la communauté que nous avons détectée avec 4 algorithmes différents sur cet ensemble de données (les algorithmes 2, 3, 4 sont des variantes que nous avons développées à des fins de comparaison) :

1. **RAK** : propagation aléatoire de tags.
2. **TagP (Tag Propagation)** : propagation sans exploiter les relations sémantiques entre les tags.
3. **SemTagP** sans intervention manuelle.
4. **Controlled SemTagP**, qui introduit un contrôle manuel pour éviter l'utilisation de certaines relations entre les tags. Nous utilisons la notation *SemTagP(tag1, tag2, ...)* pour spécifier les tags qui sont reliés par `skos:narrower` et qui doivent être ignorés ; par exemple, *SemTagP(env, energ, model)* excluera les relations `skos:narrower` avec les tags *environnement*, *énergétique* et *modélisation*.

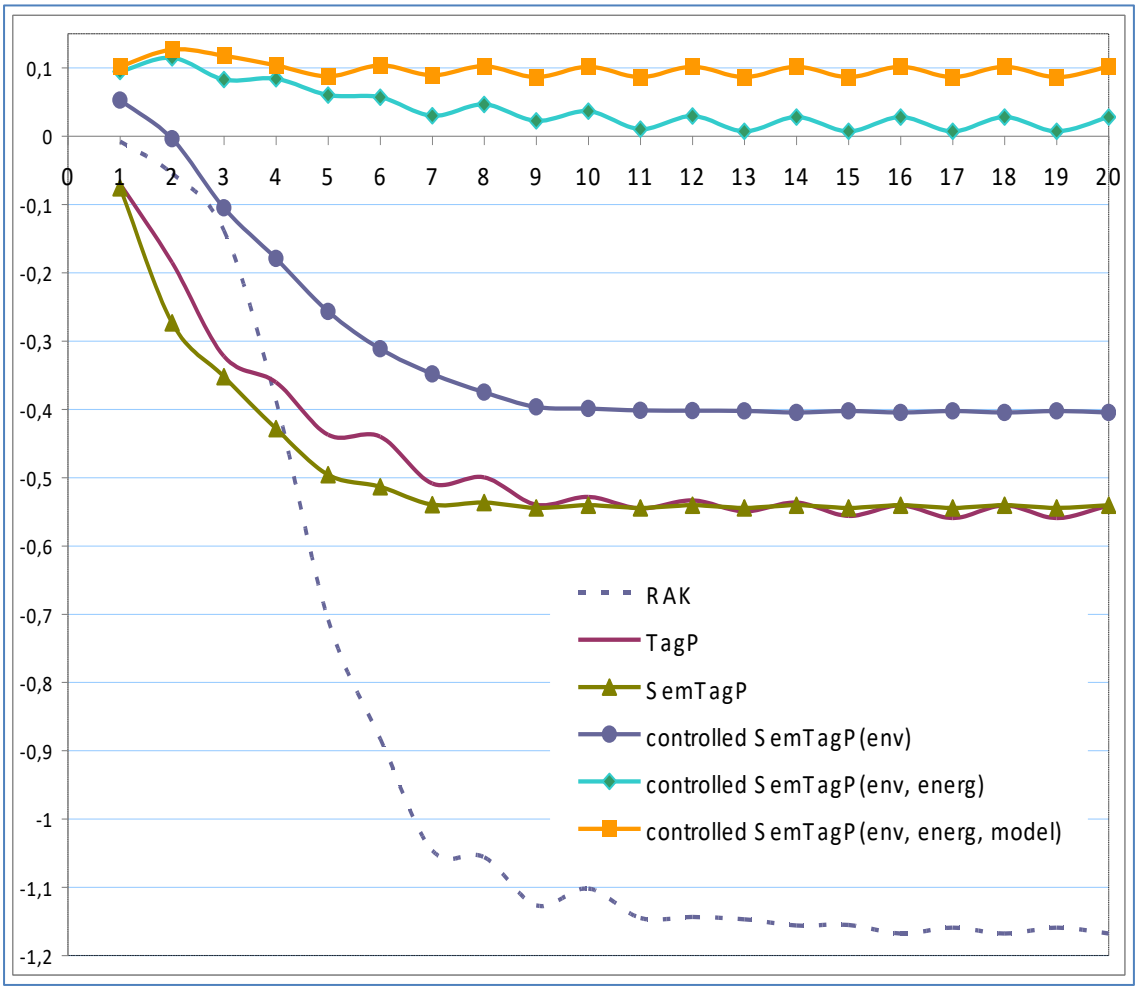


Figure 89. Modularité (axe Y) de la partition communautaire obtenue, après chaque boucle de propagation (axe X), avec les algorithmes RAK, TagP, SemTagP, SemTagP et 3 SemTagP contrôlés.

Nous avons analysé les évolutions (à chaque étape de la propagation des tags) de la modularité de la partition communautaire retournée par les 4 algorithmes et nous avons comparé ces évolutions afin d'observer la valeur ajoutée de l'exploitation des relations sémantiques.

La Figure 89 présente les courbes d'évolution de la modularité de la partition communautaire obtenue après chaque boucle de propagation. Nous observons que SemTagP(env, energ, modèle) offre une partition communautaire, dont la modularité surpasse le résultat de RAK, TagP et SemTagP. L'algorithme RAK offre la plus faible qualité de partition de la communauté sur cet ensemble de données qui est hautement centralisé avec une faible densité de liens.

En d'autres termes, les liens sociaux de ces ensembles de données ne sont pas suffisants pour révéler la structure communautaire de ce réseau social en utilisant la propagation aléatoire RAK.

TagP et SemTagP quant à eux produisent des partitions communautaires avec une modularité bien meilleure que RAK, cependant, en considérant la sémantique entre les

tags avec SemTagP, nous avons toujours une valeur de modularité proche de la modularité obtenue avec TagP (c'est-à-dire *sans exploiter les relations sémantiques* !).

Ceci est dû à une balise très large : « environnement », qui a de nombreuses relations `skos:narrower` et qui regroupe la plupart des acteurs dans une même communauté.

Avec SemTagP(env), nous excluons l'exploitation de relations `skos:narrower` avec le tag « environnement », ce qui améliore considérablement la valeur de modularité, mais on obtient en revanche beaucoup d'acteurs regroupés dans une communauté étiquetée avec « energetique ».

Enfin, nous obtenons une meilleure modularité, 0.12, avec SemTagP(env, energ, model) qui exclut l'utilisation des relations `skos:narrower` avec des balises comme: *environnement*, *énergétique* et *modélisation*, trop courantes dans l'ensemble des tags utilisés.

Nous observons 4 modèles différents de propagation des tags dans le réseau ADEME qui mettent en évidence l'exploitation à la fois de la structure du graphe social et de la sémantique émergente de la folksonomie.

D'un côté, la propagation des tags aide à diviser le réseau en groupes d'acteurs étroitement liés, et de l'autre côté, l'utilisation de relations sémantiques entre les tags aide à préserver l'identité des petites communautés, destinées à disparaître pendant la propagation, en les rassemblant en communautés plus larges mais sémantiquement liées (voir Figure 90) :

- La plupart des tags utilisées par les utilisateurs dispersés dans le réseau social disparaissent lors de la première itération, même s'ils sont utilisés par un grand nombre d'utilisateurs, et au final n'étiquettent pas une communauté dans la partition finale.
- Certains tags utilisés par un groupe d'utilisateurs bien connectés sont renforcés par la propagation et finiront par servir à l'étiquetage d'une communauté dans la partition résultante.
- Certains tags utilisés par un groupe d'utilisateurs bien connectés sont absorbés par des tags plus généraux qui finiront par étiqueter leur communauté dans la partition de la communauté résultante.
- Certains tags sont renforcés par l'exploitation des relations sémantiques qui permettent à l'algorithme de connecter des tags sémantiquement liés et de rassembler des acteurs travaillant sur des sujets similaires mais utilisant des balises plus spécialisées représentant des sous-thèmes différents.

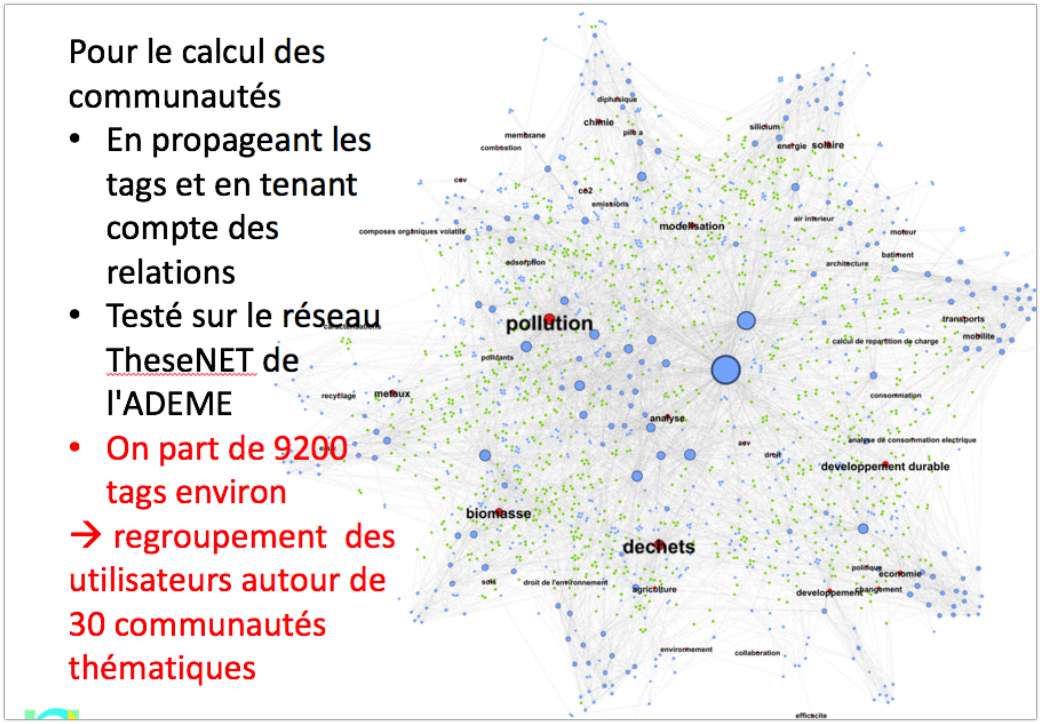


Figure 90. Les algorithmes présentés ont été capables de regrouper 9200 tags en 30 communautés thématiques.

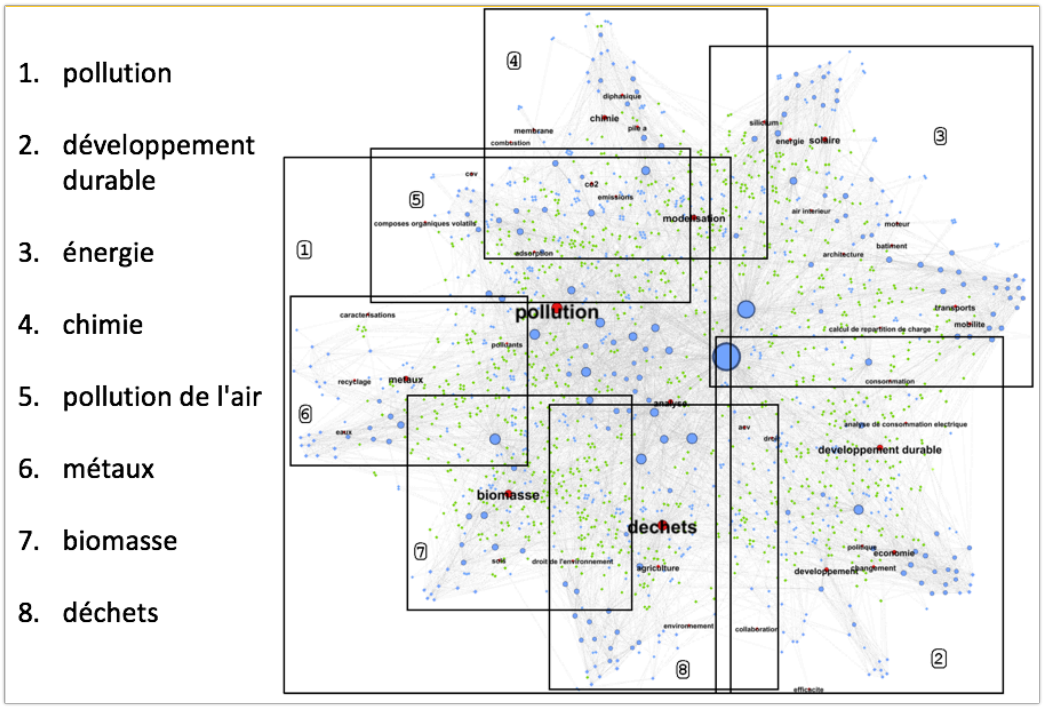


Figure 91. Réseau social des thèses de l'ADEME avec des tags labellisant les communautés obtenues avec SemTagP(env, energ, modèle). Les zones encadrées contiennent des communautés étiquetées.

La table 4 compare la taille des communautés étiquetées avec 7 tags, initialement utilisés par un nombre similaire d'utilisateurs (entre 48 et 54), avec TagP et SemTagP(env, energ, modèle). Nous observons les 4 modèles de propagation différents décrits ci-dessus :

- Les tags « évaluation », « photovoltaïque » et « innovation » ont disparu dans les deux cas parce que ces tags ainsi que les tags qui leur sont reliés par `skos:narrower` ont été utilisés par des utilisateurs dispersés dans les réseaux.
- Les tags « adsorption » et « recyclage » n'ont respectivement que 1 et 2 `skos:narrower` tags reliés (et ce sont des tags utilisés par moins de 5 acteurs). Ces tags n'ont pas été absorbés pendant la phase de propagation, ni avec TagP, ni avec SemTagP(env, energe, modèle).
- Le tag « transport » a disparu avec les deux types de propagations mais a été généralisé par SemTagP(env, energ, modèle) à une *variante orthographique*, considérée comme une balise plus large : « transports » (au pluriel), qui a 38 `skos:narrower` tags reliés.
- Le tag « métaux » qui a presque disparu avec TagP est au contraire renforcé avec SemTagP par ses relations sémantiques. En particulier, cette balise a une relation `skos:narrower` avec la balise « metaux lourds » qui est utilisée par 75 acteurs dans la folksonomie initiale.

Table 4. Comparaison de la taille des communautés étiquetées avec 7 tags (utilisés par un nombre similaire d'acteurs dans la folksonomie) avec TagP and SemTagP (env, energ, model)

Tag	**Folksonomie initiale**	**TagP**	**SemTagP(env, energ, model)**
adsorption	54	58	15 1 relation non pertinente `skos:narrower` avec *absorption spectroscopy*.
évaluation	54	4	0 no `skos:narrower`
Transport	51	1	0 28 tags `skos:narrower` et *transports* `skos:narrower` *transport*
Métaux	51	2	87 14 `skos:narrower`
Photovoltaïque	49	5	0 2 `skos:narrower`
Innovation	48	0	0 6 `skos:narrower`
Recyclage	48	8	9 2 `skos:narrower`

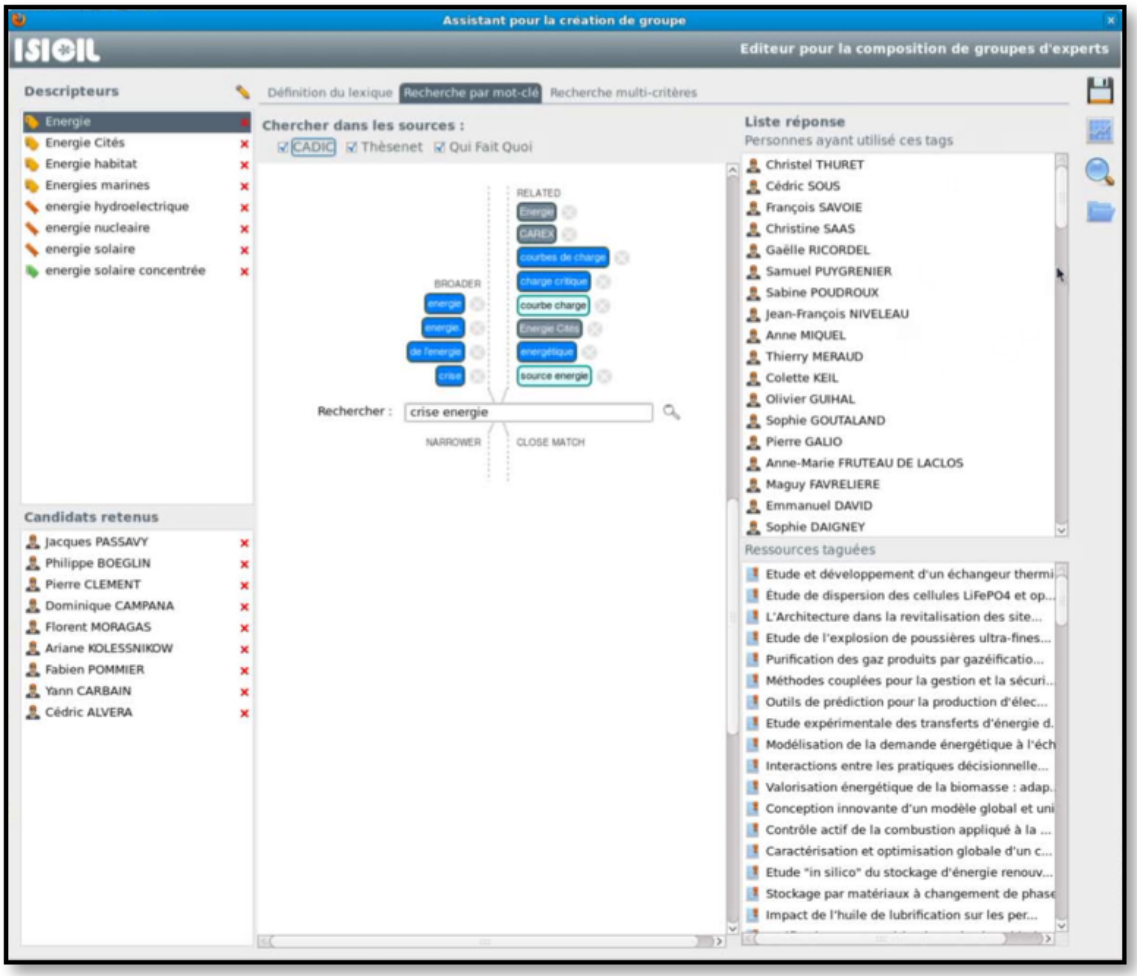


Figure 92. L'extension ISICIL pour Firefox permet d'explorer le réseau social et de manipuler les tags. Au centre, l'interface graphique de tagging social qui permet d'effectuer des recherches en sélectionnant un ensemble de tags. L'interface propose des tags reliés, plus généraux, plus spécialisés ou des tags avec une variante orthographique. On peut sélectionner plusieurs de ces tags et le système affiche la liste des personnes et des documents qui en résulte.

La Figure 91 présente une visualisation du réseau social de l'ADEME avec les tags des communautés produites par SemTagP(env, energ, energ, modèle). Nous avons utilisé un outil de visualisation graphique, GEPHI.

La taille des nœuds est proportionnelle à leur degré, et la taille rectangulaire des tags est proportionnelle à la taille des communautés étiquetées. Des groupes d'acteurs étroitement liés sont rassemblés autour de quelques tags, qui mettent en évidence l'efficacité de l'algorithme de partitionnement du réseau.

De plus, les communautés qui sont étiquetées avec des tags représentant des sujets connexes sont proches dans la visualisation, ce qui nous permet de construire des zones thématiques du réseau.

Dans la Figure 91, les communautés affichées dans les zones encadrées sont respectivement étiquetées avec : « pollution » (1), « développement durable » (2),

« énergie » (3), « chimie » (4), « pollution de l'air » (5), « métaux » (6), « biomasse » (7), « déchets » (8).

Par exemple, la zone 3 contient des tags liés à la production et à la consommation d'énergie avec les balises « énergie », « silicium », « solaire », « moteur », « bâtiment » et « transports ».

Les Figure 92 montre l'extension Firefox qui permet de faire des recherches par tags ou par personnes. En cliquant sur une personne on peut voir les tags qu'elle a utilisées et les ressources qu'elle a tagguées. En cliquant sur un tag on peut voir l'ensemble des tags ayant une relation sémantique, les personnes qui ont utilisé ce tag ou un tag relié, les ressources tagguées etc. On peut même demander (Figure 93) des visualisations plus avancées comme le graphe des relations personne-tag-personne avec les tags ayant une relation avec le tag recherché. La figure 66 montre le graphe pour les personnes « autours » du tag recherché dans la Figure 92 (« crise énergie »). On voit distinctement les personnes ayant utilisé le même tag, etc. On peut obtenir des visualisations encore plus avancées en utilisant les outils de reporting du wiki du réseau ISICIL, qui utilise le même réseau social et la même folksonomie (voir section 4.1.4).

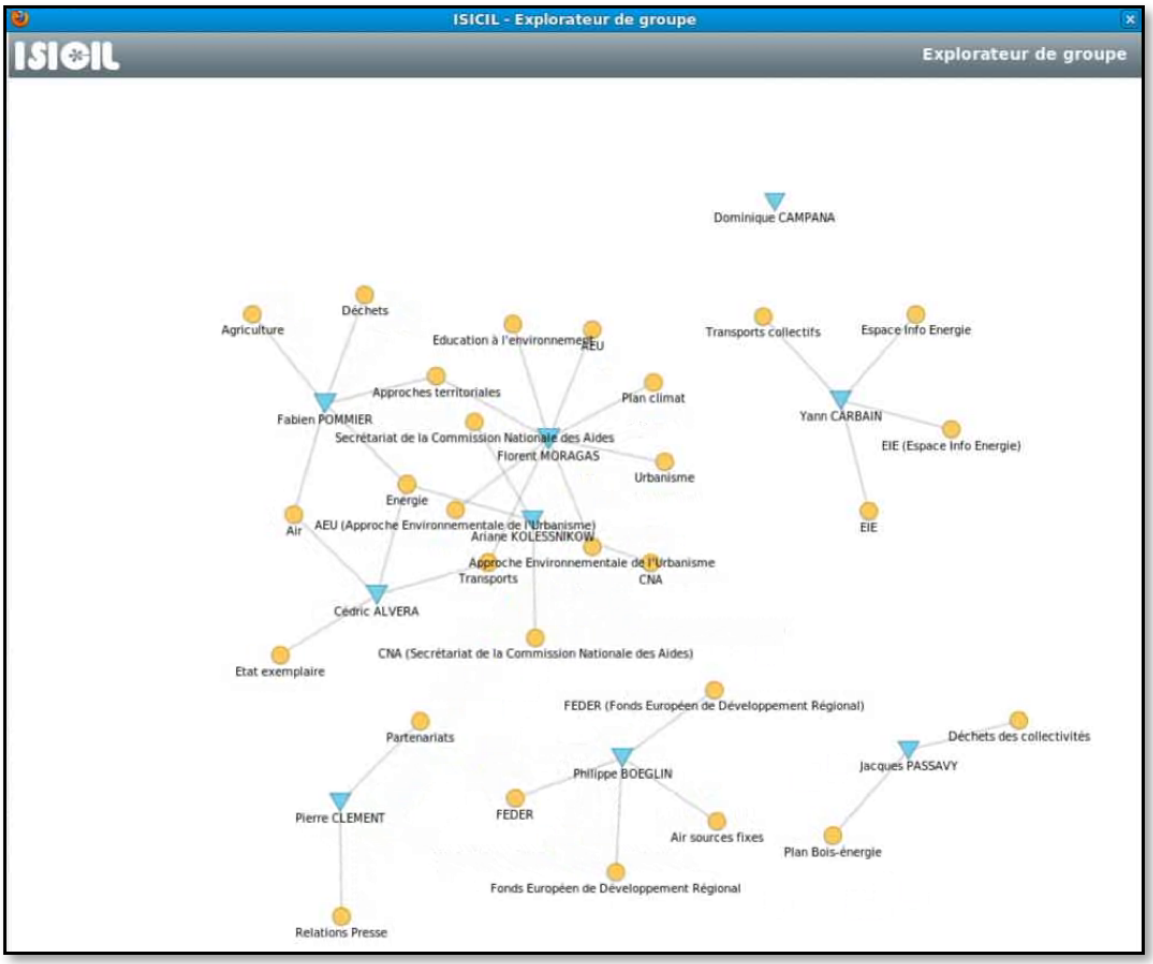


Figure 93. Relations *personne-tag-personne* de l'ensemble des tags sélectionnés dans la figure précédente. Par ex à gauche, Fabien Pommier et Cédric Alvera ont tous les deux utilisé les tags « Air » et « Energie »

7.1.5 Améliorations possibles

En termes de perspective et d'améliorations, nous pourrions aller plus loin dans l'exploitation des liens sémantiques entre les balises. (1) Dans [26] la folksonomie de l'ADEME est également enrichie avec les relations entre les tags `skos:related` et `skos:closeMatch`, dont l'exploitation devrait être étudiée. Par exemple, le triplet (*photovoltaïque* `skos:related` *énergie renouvelable*), pourrait être exploité pour compter une occurrence supplémentaire du tag « énergie renouvelable » pour chaque occurrence du tag « photovoltaïque ». (2) Nous pouvons exploiter d'autres relations sémantiques entre les tags et utiliser les implications OWL telles que les propriétés transitives. Par exemple, SKOS a des propriétés comme `skos:transitiveNarrower` (remarquez que cette fermeture transitive est indirectement réalisée par la propagation itérative de SemTagP) ; cela pourrait donner un meilleur regroupement des tags mais pourrait aussi produire des généralisations trop larges. Des énoncés sémantiques comme *énergie* `skos:transitiveNarrower` *énergie renouvelable* et *énergie renouvelable* `skos:transitiveNarrower` *photovoltaïque* pourrait être exploités pour compter une occurrence du tag « énergie » pour chaque occurrence du tag « photovoltaïque ». (3) Les primitives ontologiques

utilisées pour typer les liens entre les acteurs peuvent décrire différentes intensités de relations. Par conséquent, lorsque nous choisissons de propager des tags à travers différentes propriétés, nous pourrions donner plus de poids aux tags propagées à travers des propriétés données. Par exemple, dans un environnement de travail, les tags utilisées par des acteurs reliés par `rel:worksWith` pourraient être pondérées deux fois plus que les tags utilisées par des acteurs reliés par `rel:colleagueOf`. (4) L'algorithme peut rarement générer des communautés déconnectées mais étiquetées toutefois avec le même tag. Cela pourrait être un moyen de détecter les *trous structurels* [155]. (5) Enfin, l'algorithme actuel ne propage qu'un seul tag par acteur, une extension intéressante serait de permettre la propagation de plusieurs tags, ce qui permettrait également de détecter les communautés qui se chevauchent.

7.2 Conclusion

En s'appuyant les résultats du chapitre 6, nous avons proposé un algorithme de détection de communautés qui permet également d'étiqueter leurs activités de manière significative. Cet algorithme s'appuie sur la sémantique des descriptions des réseaux sociaux en RDF (relations entre acteurs, ressources, tags, mais aussi entre tags – la folksonomie est structurée).

À notre connaissance, il s'agit de la première détection communautaire qui détecte des communautés et contrôle à la fois leur étiquetage. Basé sur une propagation sémantique des tags, SemTagP transforme les grandes folksonomies en un sous-ensemble de tags significatifs identifiant et caractérisant les communautés. L'introduction de la sémantique dans l'algorithme de propagation de labels RAK a permis de prendre en compte non seulement la structure de liens des graphes sociaux mais aussi la sémantique des tags utilisés par ses acteurs. Le mécanisme de propagation des tags a été conçu pour exploiter la structure des liens typés des réseaux sociaux et pour regrouper les tags. L'utilisations de tags structurés, au lieu d'étiquettes aléatoires dans RAK, améliore la propagation et la fusion des tags lors de la propagation, tout en gardant un certain contrôle (on peut tenir compte de certaines relations pour empêcher la fusion de tags).

Nous avons testé et évalué ces deux principales contributions que nous avons présentées dans cet article en utilisant le réseau social construit à partir des thèses de doctorat financées par l'ADEME, l'Agence de l'environnement et de la maîtrise de l'énergie. Nous avons montré comment cette approche nous permet de détecter et d'étiqueter les communautés d'intérêt et de contrôler la précision de l'abstraction des étiquettes et la granularité des communautés. Cela nous a permis de détecter et de caractériser la distribution des agents et leurs activités à l'ADEME. Nous avons comparé la qualité de la partition obtenue avec 4 types de propagations différentes : RAK, TagP, SemTagP et SemTagP contrôlé. L'algorithme SemTagP contrôlé a surpassé les résultats des 3 autres algorithmes, soulignant que l'introduction à la fois des tags et de la sémantique entre les tags offre une amélioration significative par rapport à l'algorithme RAK.

De nombreuses tâches relatives aux systèmes d'information peuvent bénéficier de cette identification des communautés d'intérêts, allant par exemple de la gestion des ressources humaines aux notifications et à l'acheminement des demandes.

La plateforme finale a été testée avec succès à l'Ademe[99] afin d'améliorer la navigation et la recherche dans les ressources web internes et externes. Les actions de reporting effectuées par les veilleurs ont également été facilitées par les outils développés dans le cadre du projet ISICIL et intégrés (le wiki sémantique SweetDeki, présenté dans la section 4.1 de ce manuscrit utilise le système de social tagging présenté ici, par exemple, et sert à générer des rapports à partir de recherches effectuées par les veilleurs).

A noter que le travail présenté ici a été poursuivi par la thèse de Zide Meng « *Temporal and semantic analysis of richly typed social networks from user-generated-content sites on the web* », dans le cadre du projet ANR Oktopus. Zide a notamment amélioré les algorithmes présentés ici pour intégrer la détection de communautés recouvrantes [206, 207].

[99] http://isicil.inria.fr/v2/res/docs/livrables/LivrableTestsUsageISICILAdeme.pdf

8. Accès aux connaissances sur le Web et robots de télé-présence, 2013-2016

Contexte

En 2014 je suis responsable scientifique du projet AZKAR (36 mois) financé par la BPI, qui a pour partenaire le laboratoire I3S (WIMMICS), la société Robosoft (fabricant de robots), la société Another World (spin off d'Orange, spécialisée dans la visite à distance de biens culturels), le musée de la grande guerre de Meaux, et l'association Approche qui s'occupe d'enfants handicapés.

Le projet propose plusieurs scénarios mettant en œuvre un robot mobile piloté à distance, notamment pour la visite de musées pour des personnes ne pouvant se déplacer (écoles, maisons de retraites, prisons, personnes handicapées ou trop éloignées). Dans notre proposition nous décidions de placer les technologies Web au cœur du projet, en particulier le Web Sémantique (pour la modélisation des expositions, scènes, objets, parcours, profils de visiteurs et documents reliés) et le standard émergeant WebRTC qui permet les échanges temps réel audio, vidéo et data en pair à pair. C'est cette technologie qui sera utilisée pour le contrôle à distance du robot.

De nombreuses vidéos sont disponibles sur le site web du projet : http://www.azkar.fr/. Ce projet a obtenu plusieurs récompenses en 2016 (voir section suivante).

Figure 94. Plusieurs prototypes du robot utilisé lors des expérimentations. A gauche dans le musée de la grande guerre de Meaux, à droite à la cité des Sciences.

Contributions

Ce projet a consisté non seulement à montrer qu'il était possible de contrôler un robot mobile à distance en se basant massivement sur les technologies standards du W3C et en

utilisant un simple navigateur Web, mais aussi qu'il était possible de le rendre intelligent à la fois dans ses déplacements (à l'aide de capteurs et d'algorithmes de planification et d'évitement d'obstacles) mais aussi en lui donnant une connaissance du contexte dans lequel il évolue.

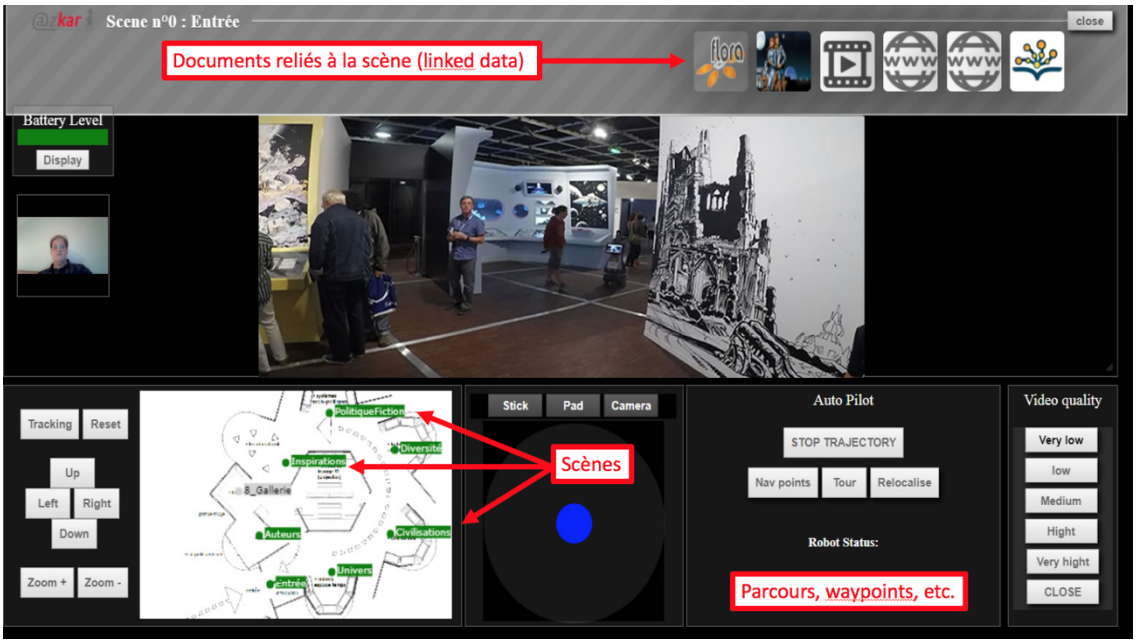


Figure 95. Interface Web proposée aux personnes qui effectuent la visite à distance. On voit ce que voit le robot, la carte des lieux, etc. En haut, dans le bandeau les documents "reliés" issus du Web de données.

Le cas d'utilisation principal sur lequel nous avons travaillé est la visite de musées. La partie « contrôle à distance en pair à pair » (Figure 95) a été réalisée à l'aide du standard WebRTC du W3C (Web Real Time Communication, dont les premières implémentations étaient disponibles au début du projet), et la partie "intelligence du robot" a fait appel aux technologies du Web Sémantique pour la modélisation du contexte dans lequel évolue le robot (exposition, scènes dans l'exposition, objets dans la scène, parcours de visite, profil des utilisateurs, etc. voir Figure 96).

Avec mon équipe, nous avons modélisé, conçu et développé la quasi intégralité de ces couches logicielles (contrôle et outils de modélisation de visites). En particulier, nous avons formalisé les modèles ontologiques utilisés, conçu des applications Web à destination du personnel des musées pour faciliter la modélisation des parcours, des scènes, des objets, et la recherche de documents reliés dans le Web de données. Nous avons aussi créé toute la couche de contrôle à distance, en particulier les applications web à destination des visiteurs, qui peuvent piloter le robot, contrôler les nombreuses caméras dont il est équipé, et qui peuvent également profiter d'une visite augmentée, le robot présentant des documents reliés à ce que le robot voit, identifiés à partir de requêtes SPARQL dynamiques dans le Web de données.

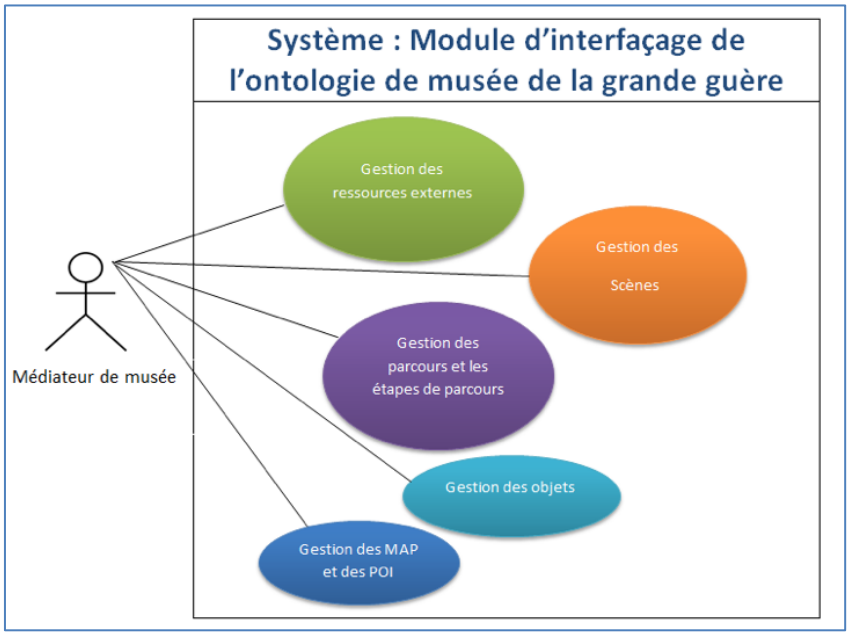


Figure 96. Des ontologies sont utilisées pour modéliser les scènes dans le musée, les objets, parcours et étapes de la visite, les points d'intérêts et les médias reliés, issus du web de données.

Ce projet a donné peu de publications scientifiques, s'agissant principalement de recherche appliquée, mais a obtenu plusieurs récompenses : le « best demo award » à la conférence ISWC 2016 et un prix délivré par l'Université Côte d'Azur de 15.000 euros en 2016.

Il a été présenté à SemWebPro 2017, et a eu de nombreuses retombées dans la presse grand public. Ce projet de recherche appliquée a été un excellent vecteur de vulgarisation scientifique et a atteint un stade préindustriel, qui fait qu'il va certainement être poursuivi. Il a fait l'objet de plusieurs publications dans des médias nationaux et internationaux (New York Times, Télé Japonaise NHK, Dauphiné Libéré, Nice Matin etc. Voir le site Web du projet http://www.azkar.fr).

Il a été utilisé pour l'inauguration de l'exposition Valerian à la Cité des Sciences et de l'industrie en Septembre 2017, à cette occasion le robot a été piloté depuis Israël, le Canada et la Corée du Sud pendant plusieurs heures.

Collaboration, encadrement

J'ai été le responsable scientifique de ce projet pour l'équipe Wimmics, mais aussi au niveau national en coordonnant le travail des différents partenaires. J'ai supervisé le travail d'un ingénieur pendant trois ans et de plusieurs stagiaires, et coordonné le travail des chercheurs de l'équipe impliqués. Il est à noter que nous avons réalisé 90% de la conception et de l'implémentation de la couche logicielle permettant le contrôle à distance, la modélisation d'expositions, parcours de visite, et la visite temps réel augmentée par des documents issus du web de données et venant enrichir l'expérience.

Publications et présentations principales ayant résulté de ces travaux :

Michel Buffa, *"*Visite de musées à distance par un robot mobile intelligent », session « session Mobilité, Véhicule Autonome et Connecté et Smart Cities*"*, conférence RoboPACA 2018. Sophia-Antipolis, France. (2018).

Michel Buffa, Catherine Faron Zucker, Thierry Bergeron, Hatim Aouzal. *"Semantic Web Technologies for improving remote visits of museums, using a mobile robot"*. Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC2016), Oct 2016, Kobé, Japan. CEUR, 2016, 〈hal-01400924〉 **Best demo award ISWC 2016, ranked A.**

Michel Buffa, *« Visites de musées à distance par un robot autonome intelligent »*, SemWeb Pro Conference, Novembre 2017, Paris (http://semweb.pro/semwebpro-2017.html)

Michel Buffa, "Le projet AZKAR : navigation d'un robot mobile temps réel à travers le Web en P2P avec l'API WebRTC", Conférence Blend WebMix 2016, 2 et 3 Novembre, Lyon, France

Michel Buffa, *"Les défis technologiques à relever au niveau de l'IA et de la robotique.",* Colloque sur l'ingénierie de la connaissance et l'intelligence artificielle, Campus Sophia Tech, 10 Mars 2017, Sophia Antipolis.

Résumé de cette section

Dans cette section, nous présentons un travail entamé en 2015 et terminé fin 2017 dans le cadre du projet AZKAR[100], financé par la Banque Publique d'Investissement (BPI), qui se concentre sur le contrôle à distance d'un robot mobile utilisant les technologies du Web sémantique pour la modélisation du contexte dans lequel évolue le robot, et Web WebRTC pour la communication en temps réel. Un des cas d'utilisation abordés dans ce projet est l'exploration télé-robotique des musées pour les écoles primaires et secondaires. Nous avons développé intégralement la couche de contrôle à distance, les modèles ontologiques et les outils pour décrire la connaissance du robot (contexte, parcours, utilisateurs, etc.)

De très nombreuses expérimentations ont été menées avec le musée de la grande guerre de Meaux, avec la Cité des Sciences à Paris, avec un robot piloté parfois pendant plusieurs heures, et sur de très longues distances (Japon-France, Canada-France, Israël-France etc.)

La question de recherche typique que nous abordons est : *« Comment pouvons-nous aider une enseignante à planifier une visite guidée d'un musée pour sa classe et comment pouvons-nous l'assister pendant la visite* » ? Notre contribution réside dans l'utilisation conjointe (1) d'un robot mobile avec des caméras et des capteurs, (2) du standard du W3C émergeant WebRTC pour la communication en temps réel, et (3) d'un ensemble de données RDF et des données liées pour représenter les données muséales et les ressources associées.

Une autre problématique concerne les outils mis à dispositions des musées pour « préparer une visite », notamment pour générer les triplets RDF nécessaires aux différentes descriptions formelles utilisées par les logiciels de navigation et de séquençage des visites (une visite contient des étapes, des points d'intérêts, propose de l'affichage contextuel de documents multimédias, de l'interactivité dans le contrôle du robot ou des caméras embarquées etc.)

[100] http://www.azkar.fr/ : site web du projet.

Dans l'expérience menée avec le Musée de la Grande Guerre en 2016, par exemple, un humain contrôle à distance, depuis son navigateur web, le robot mobile situé dans le musée, et joue le rôle de guide pour les enfants à l'école, en utilisant des outils de haut niveau pour l'aider dans cette tâche. Il dispose d'outils simples pour déplacer le robot, sélectionner les emplacements et les orientations du robot devant certaines scènes, pour visualiser des ressources multimédia en lien avec la scène observée (ex : le robot est devant une scène de tranchée, on proposera une description issue de la base de connaissance préalablement construite, mais aussi des vidéos issues du Web de données), etc. Ces outils s'appuient sur l'exploitation des descriptions sémantiques des scènes, des objets dans les scènes, des lieux / orientations possibles pour observer une scène, et plus généralement des contraintes de topologie (distances, temps pour aller d'un endroit à un autre, champ de vision des caméras embarquées, etc.).

Nous avons conçu l'ontologie AMO (**A**zkar **M**useum **O**ntology, à ne pas confondre avec l'Access Management Ontology présentée dans la section 4.3 de ce manuscrit !) et créé un ensemble de données RDF pour chaque exposition à visiter. Lors de la planification de la visite, cet ensemble de données est interrogé en combinaison avec le Web of Data pour récupérer les ressources multimédia pertinentes et les proposer aux visiteurs. Lors de la visite, des requêtes SPARQL sont déclenchées dans certaines situations (géolocalisation, temps écoulé) pour prendre ou suggérer des décisions (afficher telle vidéo, aller à l'emplacement suivant).

Dans ce chapitre du manuscrit nous détaillerons les divers aspects de ce projet, en partant d'un état de l'art sur les projets de recherche proposant de la visite à distance de musées, puis nous parlerons de la mise en œuvre de la couche de communication WebRTC, qui a demandé de nombreux benchmarkings pour trouver une solution technique efficace (couverture réseau, choix des types de caméras Pan/Tilt/Zoom, spécification de la couche de commande par échange de données pair à pair), et nous poursuivrons par l'aspect "modélisation des connaissances" et implémentation des outils finaux utilisés par les personnels des musées. Il s'agit d'un projet de recherche appliqué et la partie "ingénierie" mise en œuvre a été très importante.

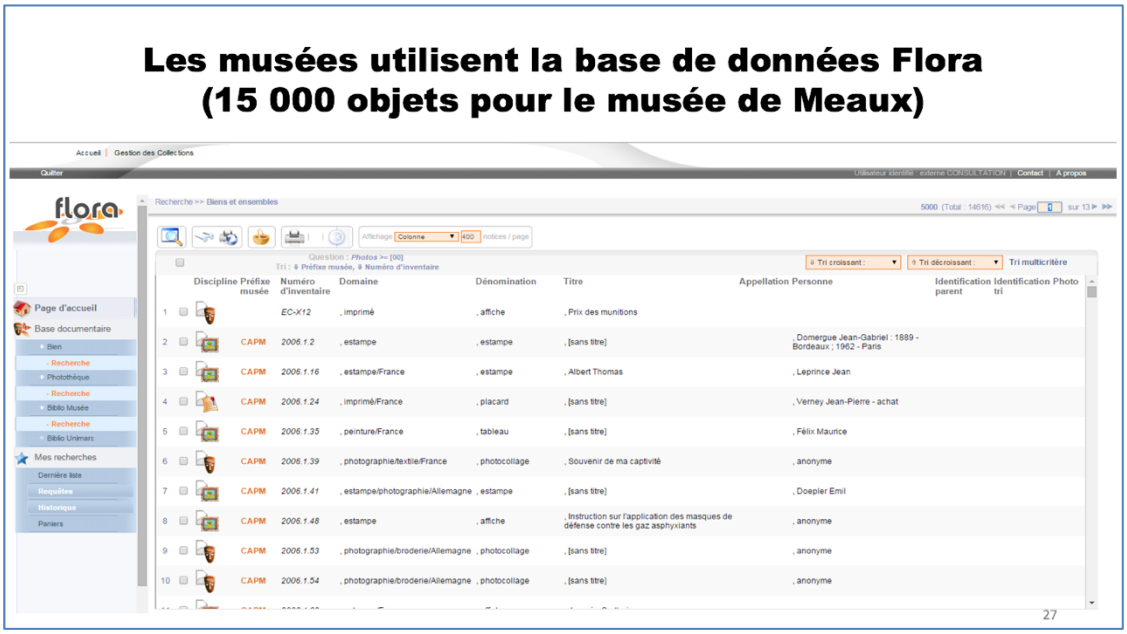


Figure 97. Les données proviennent en partie de la base de données FLORA

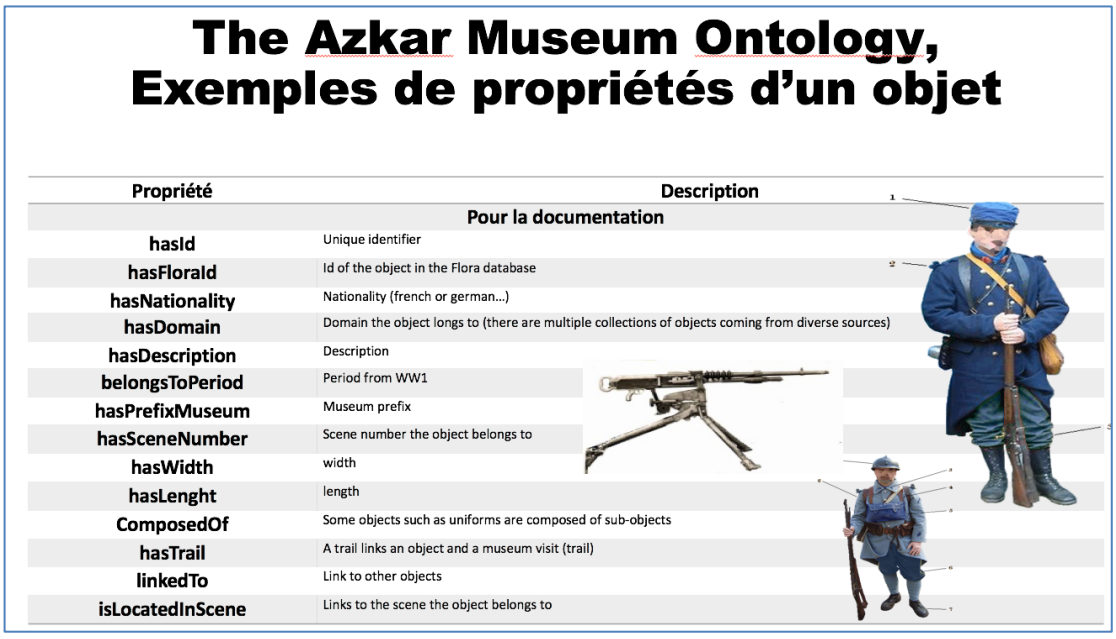


Figure 98. Exemple de propriétés d'un objet de la première guerre mondiale, modélisé avec l'ontologie AMO

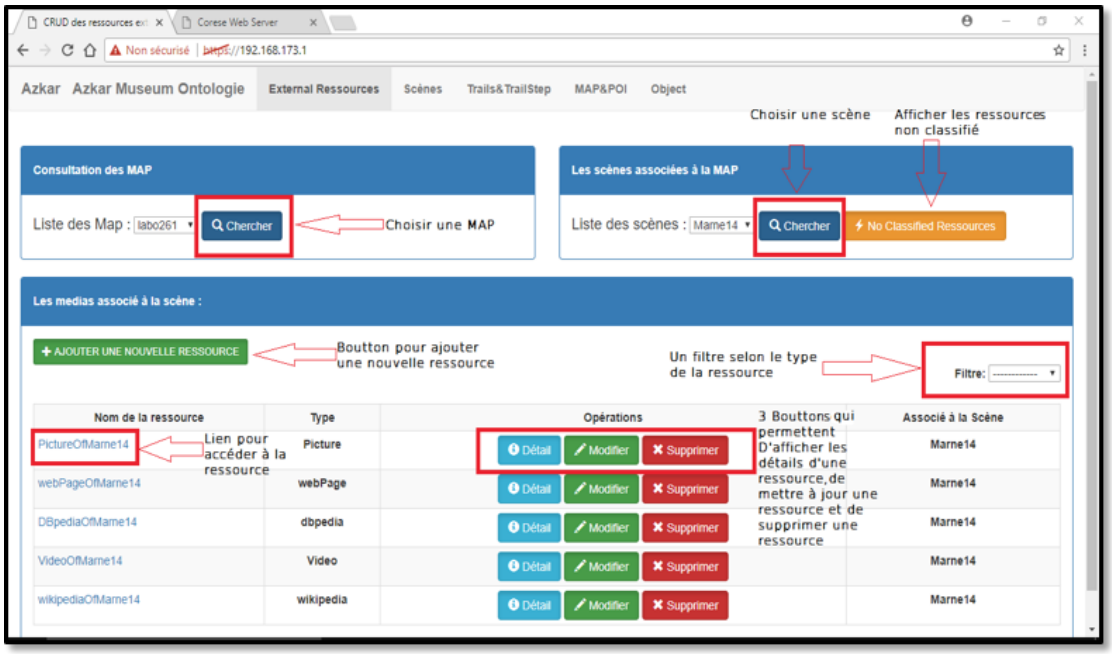


Figure 99. Exemple d'interface Web pour la saisie des médias reliés à une scène

Nous présenterons notamment l'ontologie AMO au cœur de la partie modélisation des connaissances et comment nous avons aligné des données issues de la base de données FLORA utilisée par de nombreux musées (Figure 97 et Figure 98). Nous détaillerons l'implémentation de l'architecture logicielle mettant en œuvre un moteur sémantique CORESE déporté dans le cloud et connecté de manière synchrone avec le robot via des WebSockets. Enfin, nous montrerons les outils end user développés pour faciliter l'entrée de la description sémantique d'une exposition, d'un parcours, etc. destinés aux personnels des musées (ex Figure 99).

8.1 Introduction

Le travail présenté ici a commencé en 2015 dans le cadre du projet AZKAR, financé par la Banque Publique d'Investissement française. Il se concentre sur le contrôle à distance d'un robot mobile utilisant les technologies du Web Sémantique pour la modélisation du contexte et WebRTC pour la communication en temps réel. L'un des cas d'utilisation abordés dans ce projet est l'exploration télé-robotique des musées pour les écoles primaires et secondaires et nous rendons compte ici de nos travaux permettant une visite à distance d'expositions ayant eu lieu au Musée français de la Grande Guerre situé dans la ville de Meaux et à la Cité des Sciences et de l'Industrie au cours des années 2016 et 2017 (Figure 100).

La question de recherche que nous nous sommes posée est la suivante : « *comment pouvons-nous aider une enseignante à planifier une visite guidée d'un musée pour sa classe et comment pouvons-nous l'aider pendant la visite elle-même ?* »

Notre contribution réside dans l'utilisation conjointe (1) d'un robot mobile équipé de caméras et de capteurs, (2) d'un navigateur Web pour le contrôle à distance via le standard émergent WebRTC pour la communication audio/vidéo/data et (3) d'un ensemble de

données RDF locales ou issues du Web de données pour représenter les données du musée et les ressources connexes.

Figure 100. Le dernier prototype du robot mobile utilisé lors du projet AZKAR. Ici pendant l'inauguration de l'exposition Valérian à la Cité des Sciences et de l'Industrie.

Dans l'expérience menée avec le Musée de la Grande Guerre en 2016, un humain commande à distance le robot mobile dans le musée, et joue le rôle de guide pour les enfants à l'école, en utilisant des outils de haut niveau pour l'aider dans cette tâche, pour concevoir la visite, pour sélectionner les emplacements et orientations du robot devant certaines scènes, pour proposer des ressources multimédia liées, etc.

Ces outils reposent sur l'exploitation des descriptions sémantiques des scènes, des objets dans les scènes, des lieux/orientations possibles pour l'observation d'une scène, et plus généralement, des contraintes topologiques (distances, temps de déplacement d'un lieu à un autre, temps de prise de vue à bord, etc.).

Nous avons conçu l'Azkar Museum Ontology (AMO) et créé un jeu de données RDF correspondant à deux des scènes de musée de la Grand Guerre (une tranchée et une exposition de soldats en uniformes, Français et Allemands, lors de l'année 1914, voir Figure 101).

Figure 101. Scène de référence « Marne 14 », soldats français et Allemands équipés de leurs uniformes.

Figure 102. Seconde scène de référence : tranchées françaises et allemandes.

Lors de la planification de la visite, cet ensemble de données est interrogé en combinaison avec le Web de données pour récupérer les ressources multimédia pertinentes à proposer aux visiteurs, et lors de la visite, des requêtes sont déclenchées dans certaines situations (géolocalisation, temps écoulé) afin de prendre ou de proposer des décisions pour afficher une vidéo, une explication issue de DBPedia, pour aller au lieu suivant).

La section suivante résume les travaux connexes au projet AZKAR. La section 8.3 présente l'architecture matérielle et logicielle mise en œuvre. La section 8.4 présente l'ontologie de l'AMO et l'ensemble de données que nous avons construit pour le Musée de la Grande Guerre. La section 8.5 montre des exemples d'applications Web conçues pour faciliter la création des datasets et des résultats d'expérimentations à la Cité des Sciences et de l'Industrie lors de l'événement de fin de projet, la section 8.6 conclut.

8.2 Travaux connexes

Le nombre de robots déployés dans les musées et les expositions n'a cessé d'augmenter (voir [209] pour une synthèse), la plupart d'entre eux agissant comme de simples systèmes de vidéoconférence mobile, comme dans les exemples cités dans l'offre Robot Tours[101] du National Australia Museum. Cependant, dans ce domaine précis, les travaux de recherche impliquant des robots mobiles ne reposent généralement pas sur des descriptions sémantiques des scènes, et se concentrent davantage sur des contraintes de bas niveau reliés à la nature des capteurs, à la latence ou à la sécurité (voir par exemple [210] ou [211]).

En ce qui concerne l'utilisation de modèles de connaissance, plusieurs travaux ont été menés dans le but de développer des systèmes de soutien à la visite des musées et à l'accès au patrimoine culturel, la plupart d'entre eux impliquant des dispositifs mobiles (téléphones, tablettes) adaptables au profil de l'utilisateur et sensibles à son contexte, pour améliorer son expérience et aider à construire sa visite du musée en fonction de ses préférences et contraintes.

Plusieurs types de systèmes de recommandation de ce type ont été développés par des chercheurs, par exemple, avec des recommandations basées sur le contenu dans le cadre du projet CHIP [212], ou basées sur le filtrage collaboratif comme par exemple dans les travaux de [213] et [214].

Le projet AZKAR est à l'intersection des deux mondes : il utilise un robot mobile et un jeu de données RDF pour décrire des profils de visiteurs de haut niveau (écoles primaires et secondaires) ainsi que le contenu historique des scènes muséales que le robot va explorer. Les articles cités ci-dessus, ainsi que le projet Hippie [215] ont été une bonne source d'inspiration pour le contrôle à distance que nous avons mis en place et pour la conception de haut niveau du vocabulaire AMO. Comme Hippie, nous sommes partis d'une grande base de donnés (FLORA) qui décrit l'ensemble détaillé de la collection du musée (plusieurs centaines de milliers de pièces), avec de nombreux détails et objets qui ne sont pas visibles pendant une visite –seule une infime partie est exposée- et pour arriver à une description résumée et synthétique, adaptée au scénario envisagé, des scènes du musée pertinentes pour la visite.

[101] http://www.nma.gov.au/engage-learn/robot-tours

8.3 Architecture logicielle et matérielle de la solution AZKAR

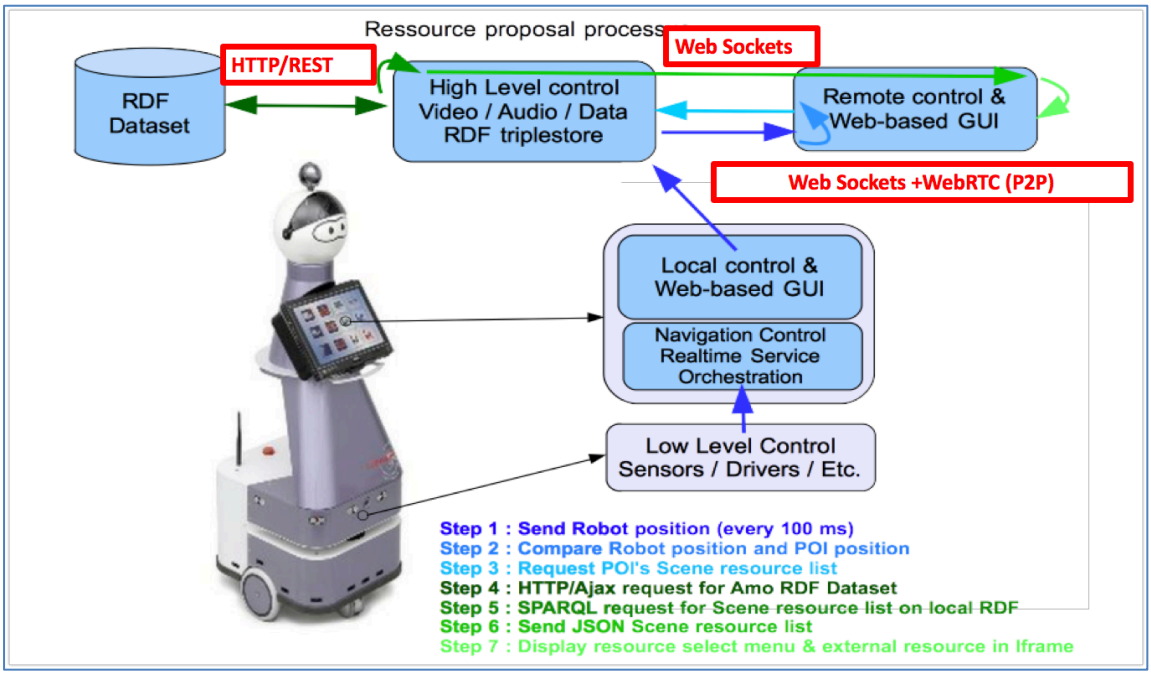


Figure 103. Architecture logicielle et matérielle de la solution AZKAR.

Le robot est équipé de plusieurs ordinateurs embarqués, et de plusieurs types de capteurs (laser range finder pour la cartographie et la navigation, il est capable d'aller d'un point de la carte à un autre en évitant les obstacles éventuels, capteur à ultrason pour les arrêts d'urgence, plusieurs caméras orientables et zoomables, caméra spéciale pour le débloquer, orientée vers le sol, etc.). Il communique avec l'extérieur via WebRTC pour la partie vidéo/audio/data et via des WebSockets pour communiquer avec un triple store RDF déporté de type CORESE (Voir Figure 103).

Lorsque le robot se déplace, des requêtes SPARQL sont envoyées pour décider des actions à prendre en fonction de sa position (s'arrêter, orienter la caméra vers un point d'une scène, envoyer une requête sur le Web de données pour afficher un document DBPedia relié, une vidéo YouTube, un document de la base RDF AZKAR, etc.

8.4 Le Vocabulaire de l'ontologie AMO et le jeu de données RDF décrivant des scènes du Musée de la Grande Guerre

Le vocabulaire de l'ontologie AMO développée à l'époque de cette expérience est disponible en ligne[102] ; il comprend 11 classes principales et 39 propriétés. Ses principales classes représentent des objets de musée, des scènes, des points d'intérêt, des cartes, des chemins de visite (parcours composés de points où le robot s'arrête, et de points d'observations). Ses principales propriétés permettent de relier des objets, des points d'intérêt et des médias externes à des scènes, des scènes à des parcours, des tracés à des cartes et de décrire ces instances.

[102] http://mainline.i3s.unice.fr/azkar/ontology/

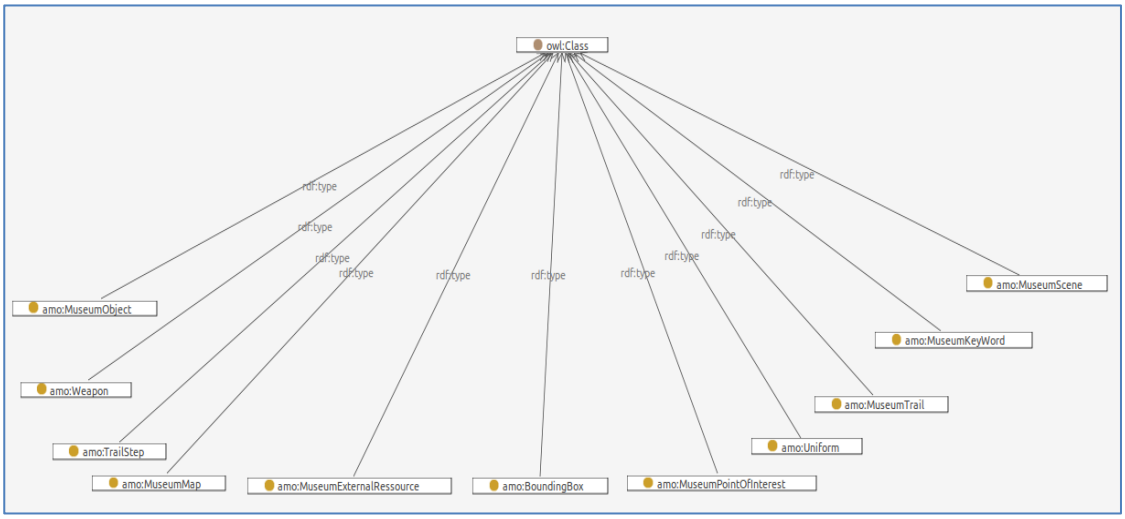


Figure 104. Graphe des classes de l'ontologie AMO

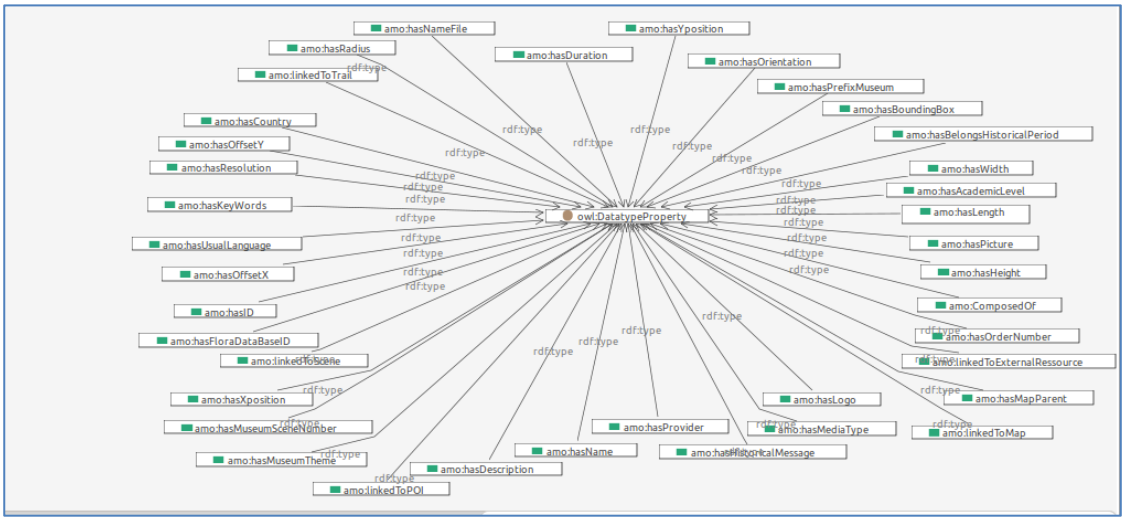


Figure 105. Graphe des propriétés de l'ontologie AMO.

Sur la base de l'ontologie AMO, nous avons créé un jeu de données RDF à partir de la base de données relationnelle FLORA utilisée par de nombreux musées français, qui contient des descriptions très détaillées de chacun des objets de la collection du musée. Le jeu de données RDF développé pour cette expérience comprenait 421 instances et 2401 triplets décrivant deux scènes (un ensemble de soldats français et allemands entièrement équipés, appelés "Marne 14", et deux tranchées).

La base de données SPARQL est constituée de 32 requêtes, dont les principales effectuent des tâches telles que « *donner la position x et y du robot et le champ de vision des caméras, donner la description de la scène actuelle ainsi que les ressources multimédia associées* », ou « *envoyer la description précise de tous les soldats de la scène Marne 14 avec les détails et les photos (urls) de leur équipement* ».

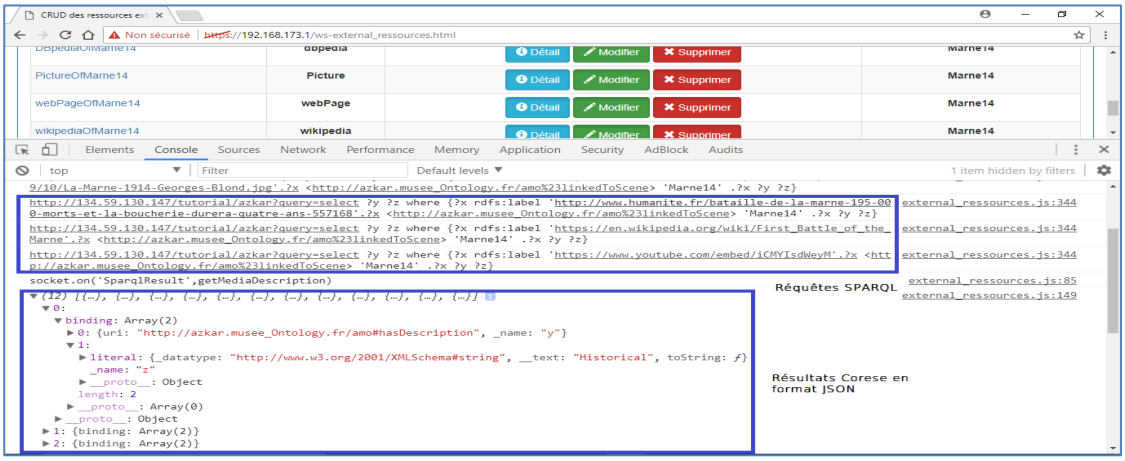


Figure 106. Extrait d'une des interfaces graphiques de création de datasets RDF. On voit les requêtes SPARQL utilisées pour connaître les documents reliés à une scène dans le debugger.

Dans notre expérimentation, nous avons télécommandé le robot à partir de nos bureaux, à 800 kilomètres du musée. Les commandes, les données des capteurs, les flux audio et vidéo sont échangés via WebRTC, avec une connexion pair à pair avec le robot. Lorsque ce dernier se trouve à proximité d'une scène donnée, différents points d'observation apparaissent sur la carte, ainsi que des ressources liées (description multimédia de la scène : ressources locales mais aussi des ressources provenant de sources de données externes telles que DBpedia.fr ou autres), comme le montre la Figure 107 et la Figure 108. Les requêtes SPARQL sont déclenchées en fonction de l'emplacement du robot, de l'heure, de la scène observée, des interactions du pilote et de la façon dont la visite a été conçue (profils des utilisateurs, scénario).

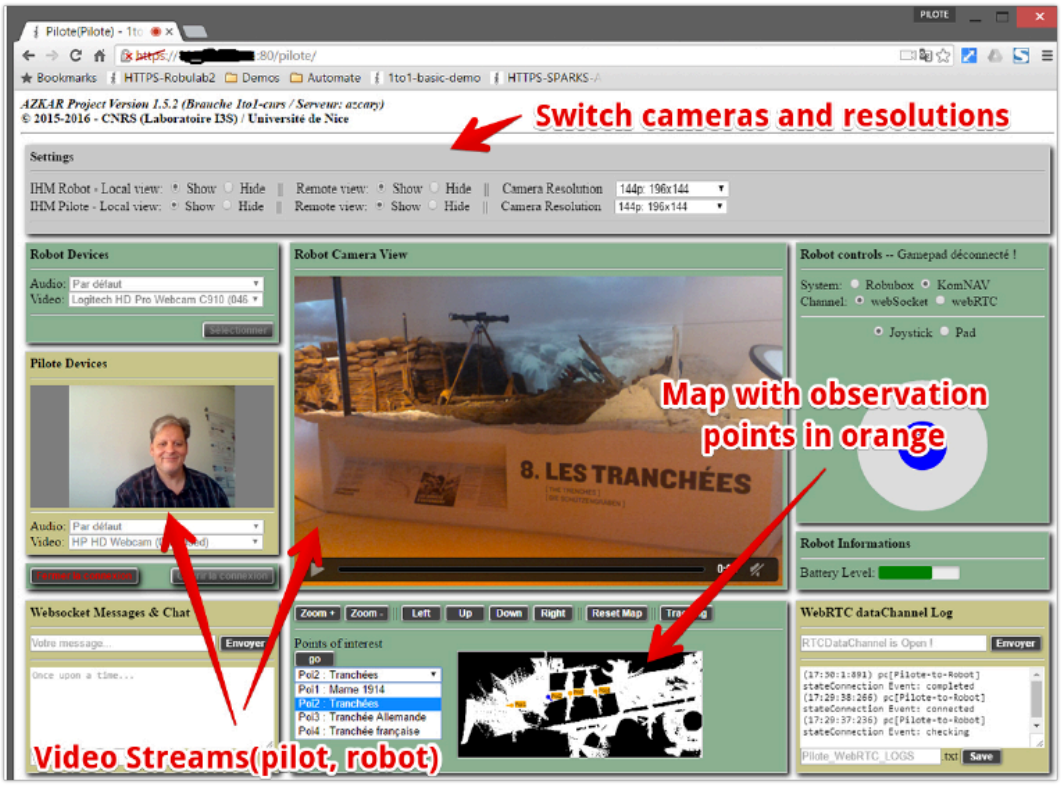


Figure 107. Interface graphique du pilote distant. On peut voir les points d'observation associés à la scène de la tranchée, en orange, en haut de la carte.

Figure 108. Cette deuxième vue commutable présente les ressources multimédias suggérées. En cliquant sur l'une d'elles, l'url de la ressource sera envoyée au client pour un rendu local (vidéo, image, html).

8.5 Expérimentations à la Cité des Sciences et de l'industrie

Cette expérimentation a eu lieu en Septembre 2017 au cours de l'exposition dédiée à la bande dessinée Valérian. Arrivée à la fin du projet, elle a été l'occasion pour nous de mettre en œuvre les derniers outils et la dernière version de l'ontologie AMO. En particulier des outils *end user* ergonomiques ont été développés afin de faciliter la création des datasets (définition des parcours, documents reliés, etc.), destinés à être utilisés par les équipes dirigés par les « commissaires d'expositions », c'est-à-dire les personnes responsables de la création d'expositions.

Ces outils ont été présentés lors de la conférence SemWebPro 2017[103]

8.5.1 Quelques exemples d'outils utilisés pour la création du Dataset

Construction des chemins de visite, définition des points d'intérêt et d'observation.

La première étape consiste à demander au robot de cartographier les lieux. Pour cela il suffit de le promener dans l'exposition à l'aide d'un Game pad sans fil de type console de jeux Xbox. On peut également, si la zone est fermée, le laisser explorer seul la zone (Figure 109).

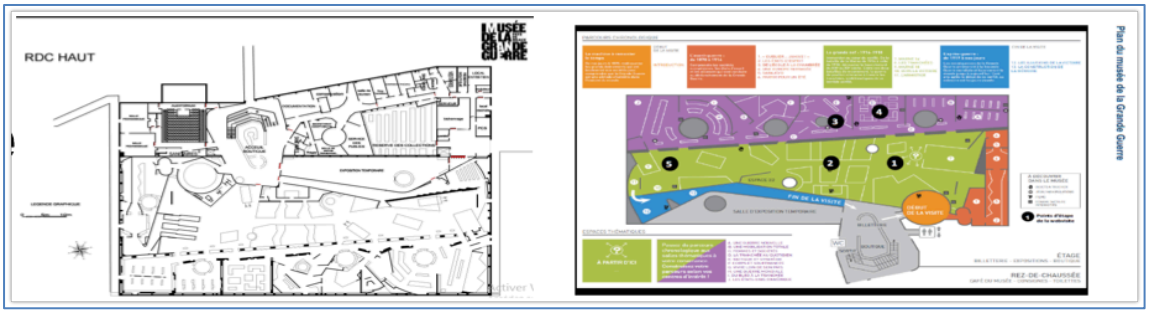


Figure 109. Carte du musée de la Grande Guerre. Préalablement à la visite le robot a cartographié les endroits dans lequel il allait évoluer avec un système DSLAM (construction dynamique de carte à partir de capteurs lasers).

La seconde étape consiste à piloter manuellement, toujours à la manette, le robot vers des points d'intérêt et à les définir en cliquant sur certains boutons de la manette. Le robot va alors générer des fichiers de description qui seront ensuite utilisés pour la définition de la visite (Figure 110).

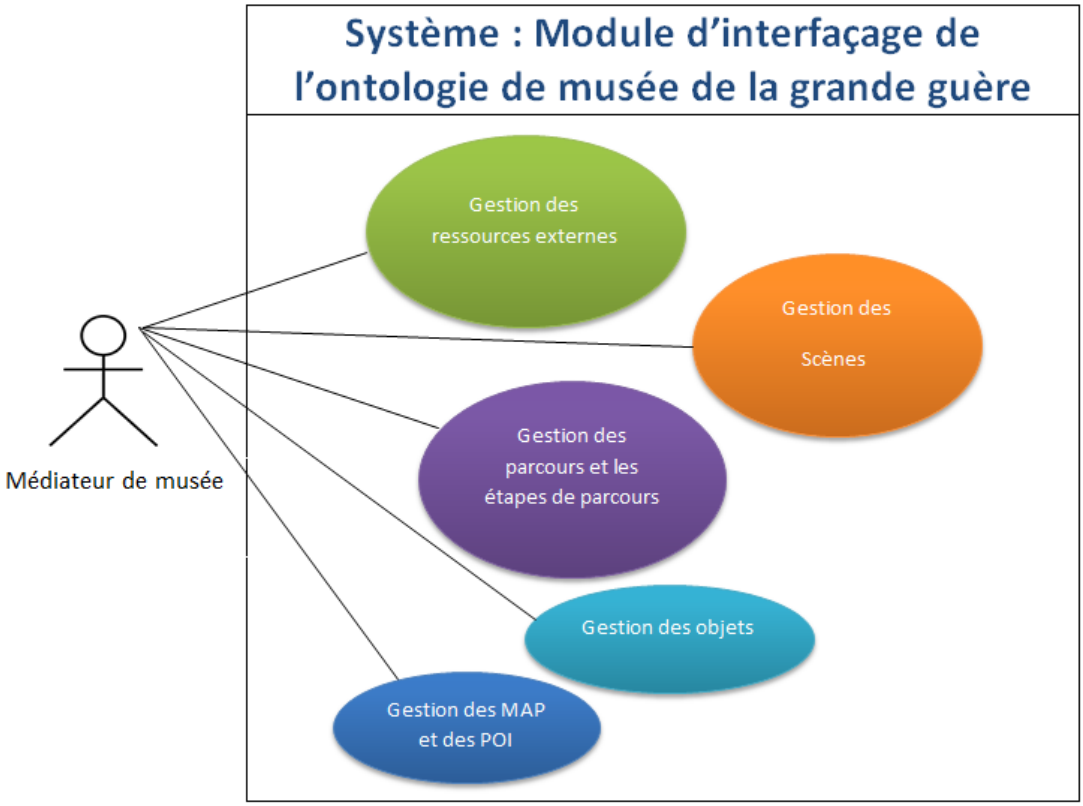
[103] Vidéo et présentation : http://www.semweb.pro/semwebpro-2017.html - programme

```
var Map1 = {"Name":"C:\\Maps\\labo261.png","Offset":{"X":-20.7969943946,"Y":-3.02713963867},
           "Width":1485,"Stride":1488,"Height":1187,"Resolution":0.019999999553};

var listPoisMap1 = [
  {"Id":0,"Name":"Start","Label":"Start","Pose":{"X":0,"Y":0,"T":0}},
  {"Id":1,"Name":"Pilier1","Label":"Pilier1","Pose":{"X":2.657378405937024,"Y":6.2462976540686714,"T":0}},
  {"Id":2,"Name":"Pilier2","Label":"Pilier2","Pose":{"X":2.4719096600822503,"Y":11.699078782199011,"T":0}},
  {"Id":3,"Name":"Est","Label":"Est","Pose":{"X":-3.3665679723240238,"Y":14.266605293166855,"T":-3.119}},
  {"Id":4,"Name":"Pilier3","Label":"Pilier3","Pose":{"X":-10.89460775413475,"Y":10.917516728817986,"T":0}},
  {"Id":5,"Name":"Pilier4","Label":"Pilier4","Pose":{"X":-10.124021004815544,"Y":5.938340810140021,"T":0}},
  {"Id":6,"Name":"Ouest","Label":"Ouest","Pose":{"X":-3.8404098830888453,"Y":3.2119936603855423,"T":-0.3341195681236033 8}}
];


var Map2 = {"Name":"C:\\Maps\\valerian3.png","Offset":{"X":-10.0026429165,"Y":-9.67300262729},
           "Width":1577,"Stride":1584,"Height":1577,"Resolution":0.019999999553};

var listPoisMap2 = [
  {"Id":0,"Name":"1_Start","Label":"Start","Pose":{"X":0,"Y":0,"T":0}},
  {"Id":1,"Name":"2_Entrée","Label":"Entrée","Pose":{"X":2.2366425481400523,"Y":3.8208672334191327,"T":0.415}},
  {"Id":2,"Name":"3_Univers","Label":"Univers","Pose":{"X":7.40450760117759,"Y":4.7484327557592021,"T":-1.52}},
  {"Id":3,"Name":"4_Civilisations","Label":"Civilisations","Pose":{"X":12.327421051402013,"Y":7.6396675126498863,"T":-0.37986
  {"Id":4,"Name":"5_Diversité","Label":"Diversité","Pose":{"X":12.7039813938932,"Y":14.028970283373626,"T":0.5191068077938954
  {"Id":5,"Name":"6_PolitiqueFiction","Label":"PolitiqueFiction","Pose":{"X":7.2040224556035417,"Y":16.827064742658582,"T":1.
  {"Id":6,"Name":"7_Inspirations","Label":"Inspirations","Pose":{"X":3.5437583861210982,"Y":13.040015625263182,"T":2.68127015
  {"Id":7,"Name":"8_Gallerie","Label":"Gallerie","Pose":{"X":-0.41224895371263182,"Y":10.876432910249717,"T":2.3574827652894
  {"Id":8,"Name":"9_Auteurs","Label":"Auteurs","Pose":{"X":1.3194304490554116,"Y":7.515091839944076,"T":-1.00853705665336}}
];
```


Figure 110. Exemple de fichier de description de points de passage et de points d'observation.

Création de visites

Ensuite, on peut passer à la création de visites à proprement parler, en utilisant des interfaces graphiques qui vont exploiter les différentes parties du vocabulaire AMO. L'ontologie contient en effet des modules permettant de décrire les points suivants :



Nous montrons ici des extraits des interfaces utilisateurs utilisées pour la création de visites dans l'exposition Valérian mais aussi au Musée de la Grande Guerre.

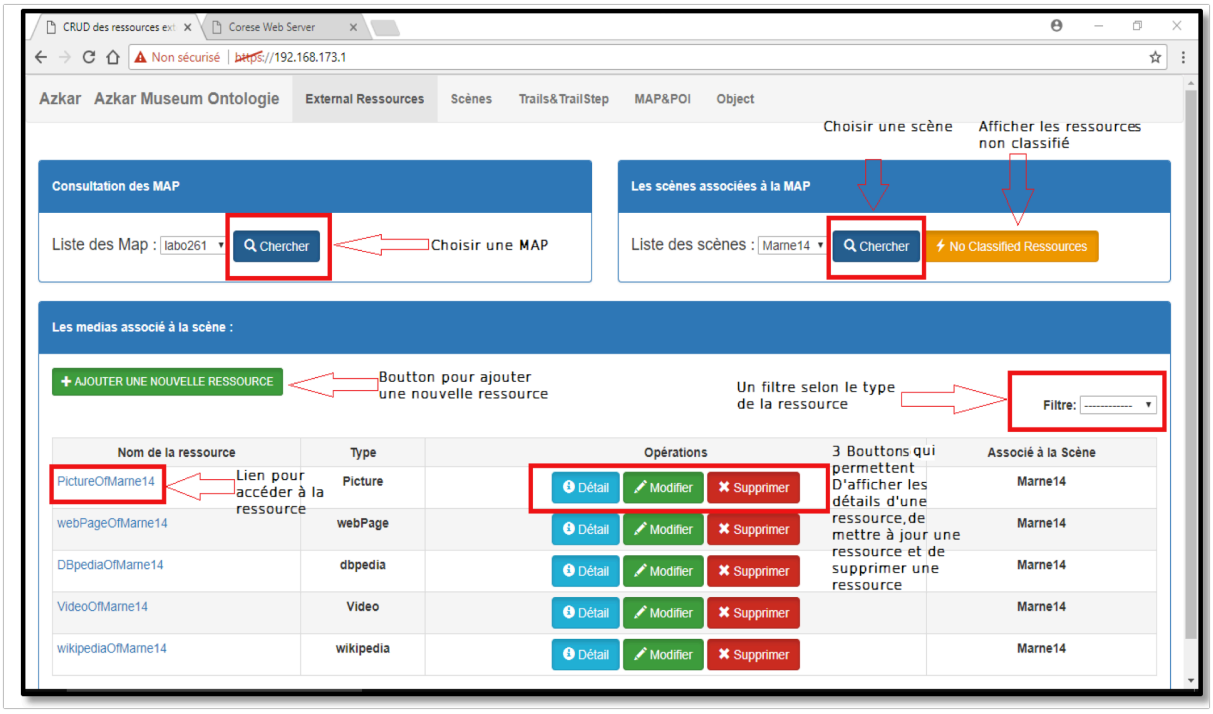


Figure 111. Outil de gestion des « cartes » et des « scènes ». Dans la zone du bas, ajout de documents liés.

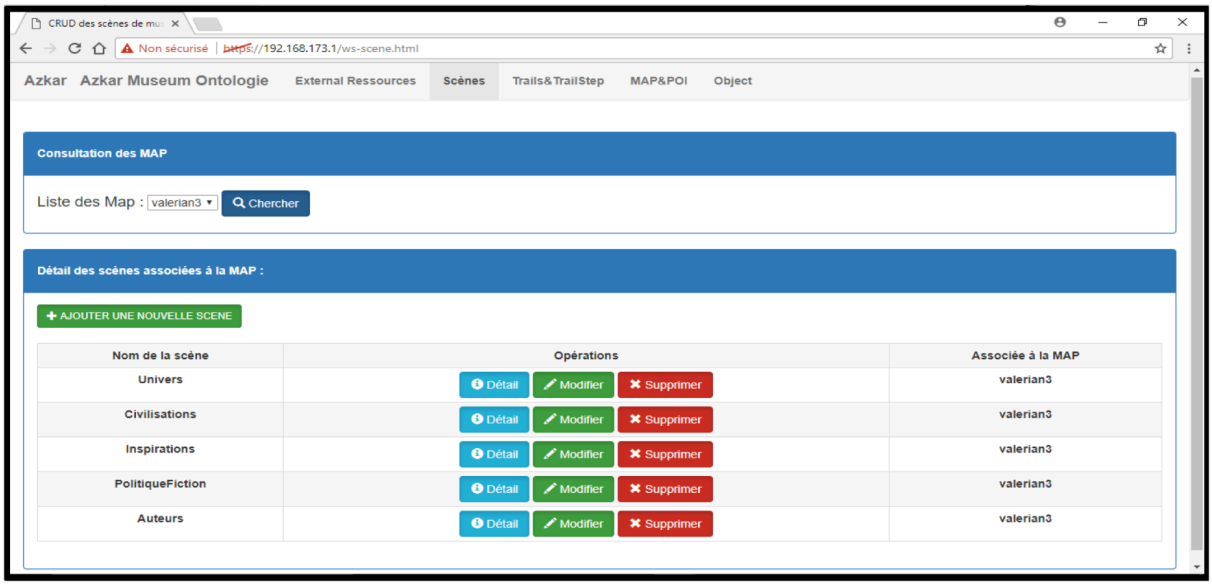


Figure 112. Gestion de scène, on peut ajouter/modifier/supprimer une scène dans une carte.

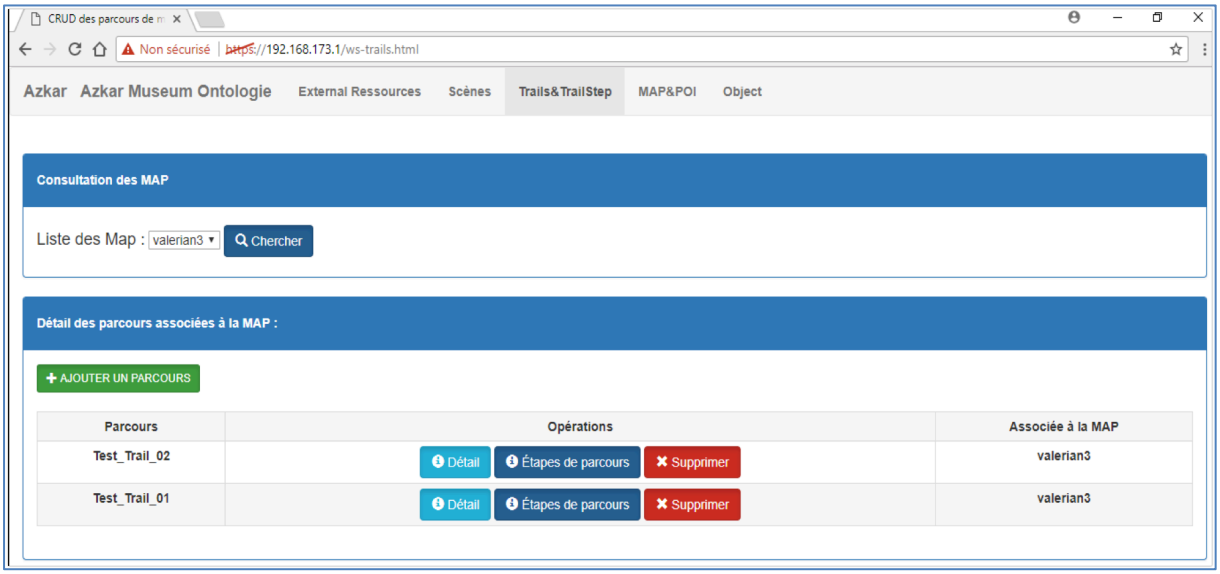


Figure 113. Définitions des points de passage.

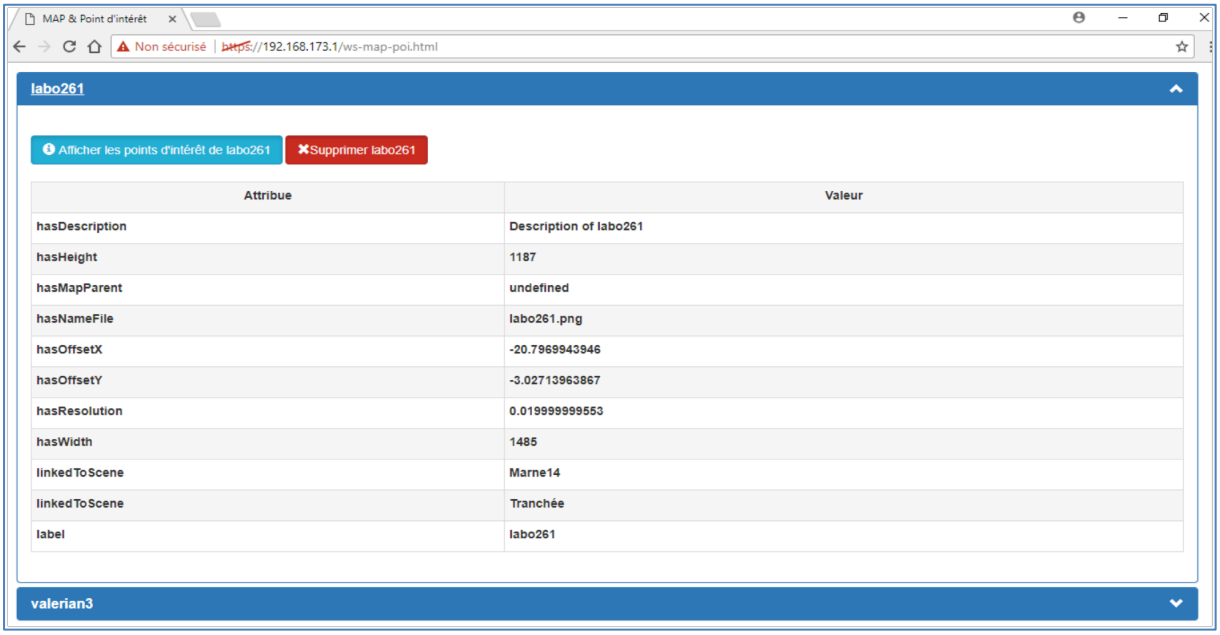


Figure 114. Détails d'un point de passage.

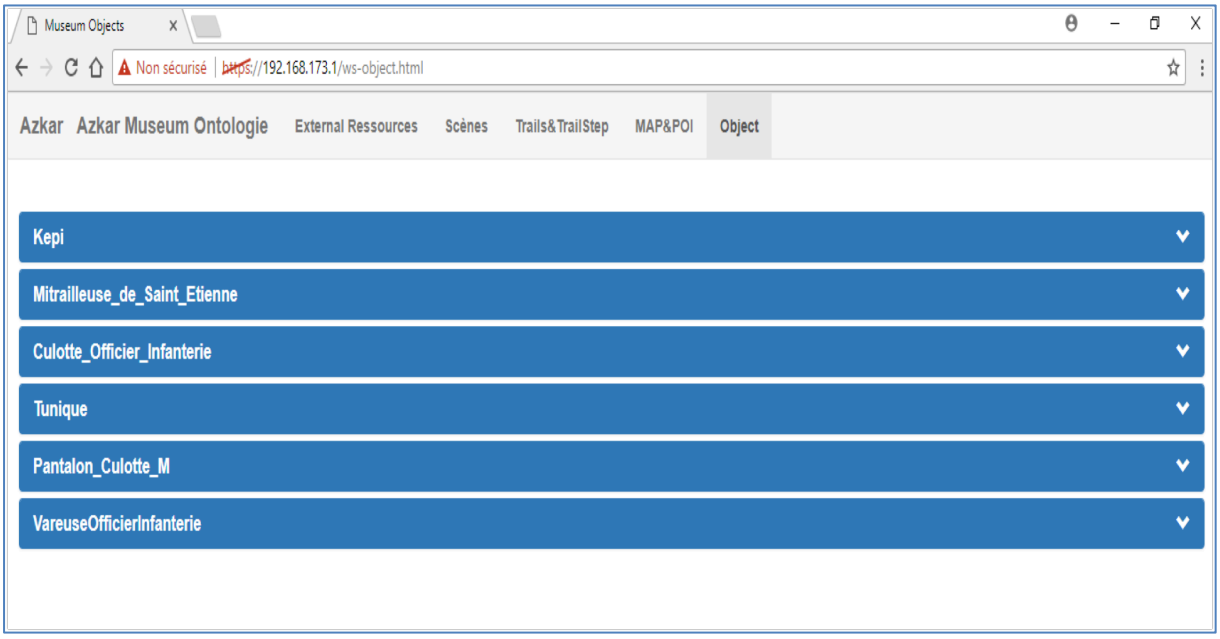


Figure 115. Gestion des objets d'une scène.

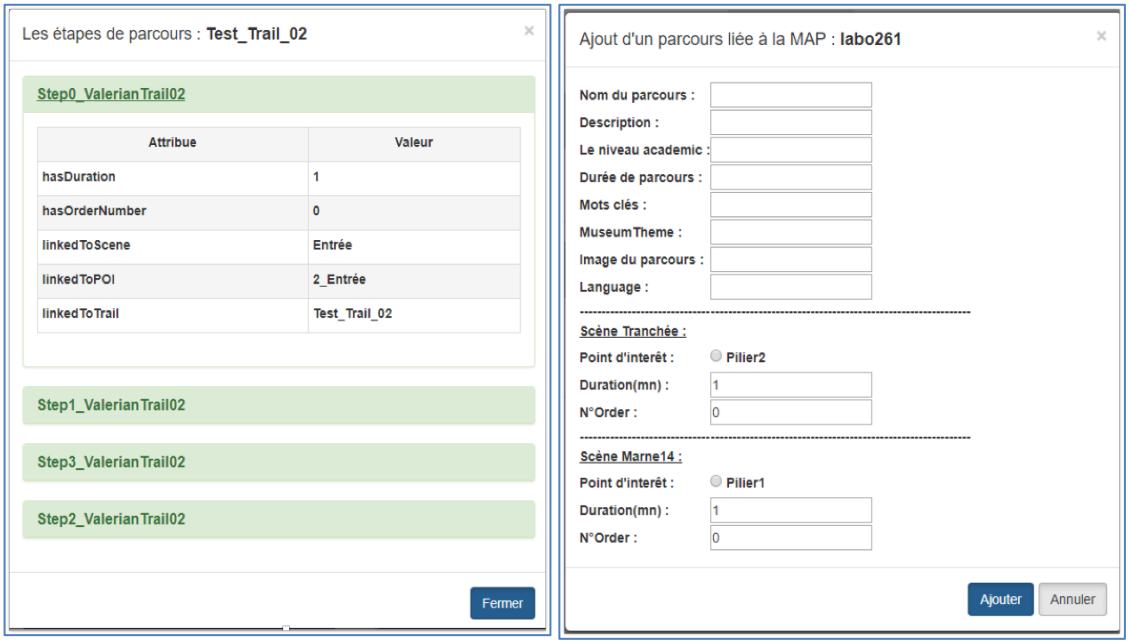


Figure 116. Définition de parcours de visite…

8.5.2 Extraits de l'expérience utilisateur lors d'une visite

Le lecteur pourra voir des vidéos qui ont été réalisée à la fin du projet[104].

[104] Vidéo de fin de projet : https://www.youtube.com/watch?v=Lqisd_qMYOI&t=3s

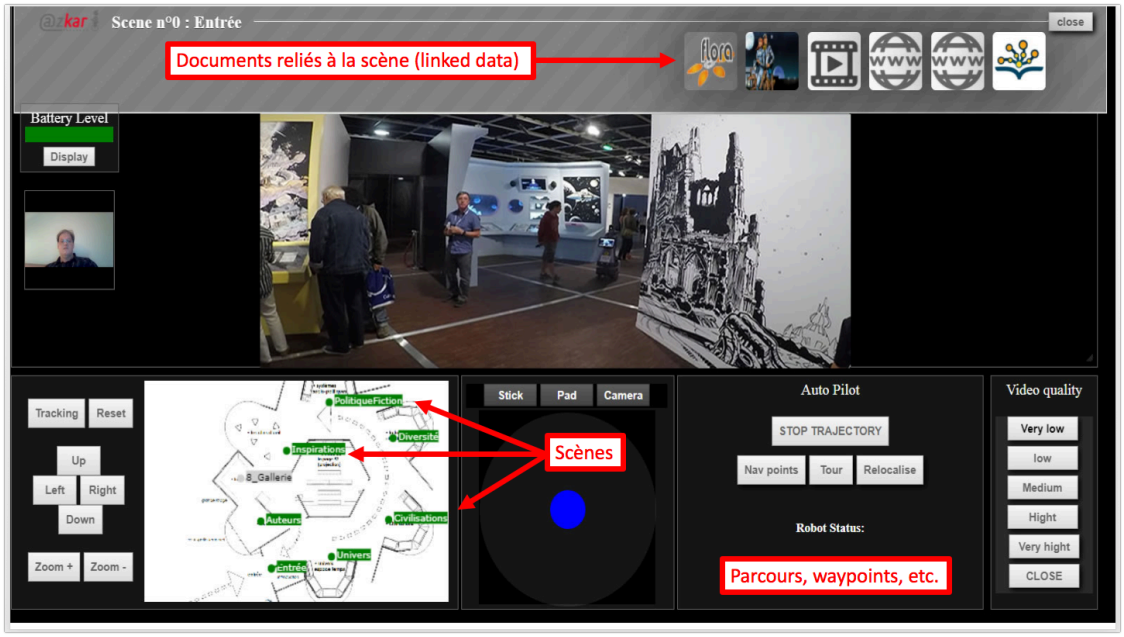


Figure 117. Interface de visite à distance. Ici dans lors de l'exposition Valérian en Septembre 2017. Le robot s'arrête à un point d'observation, on peut voir la carte des lieux, la vision de la caméra principale, et les documents reliés en haut à droite.

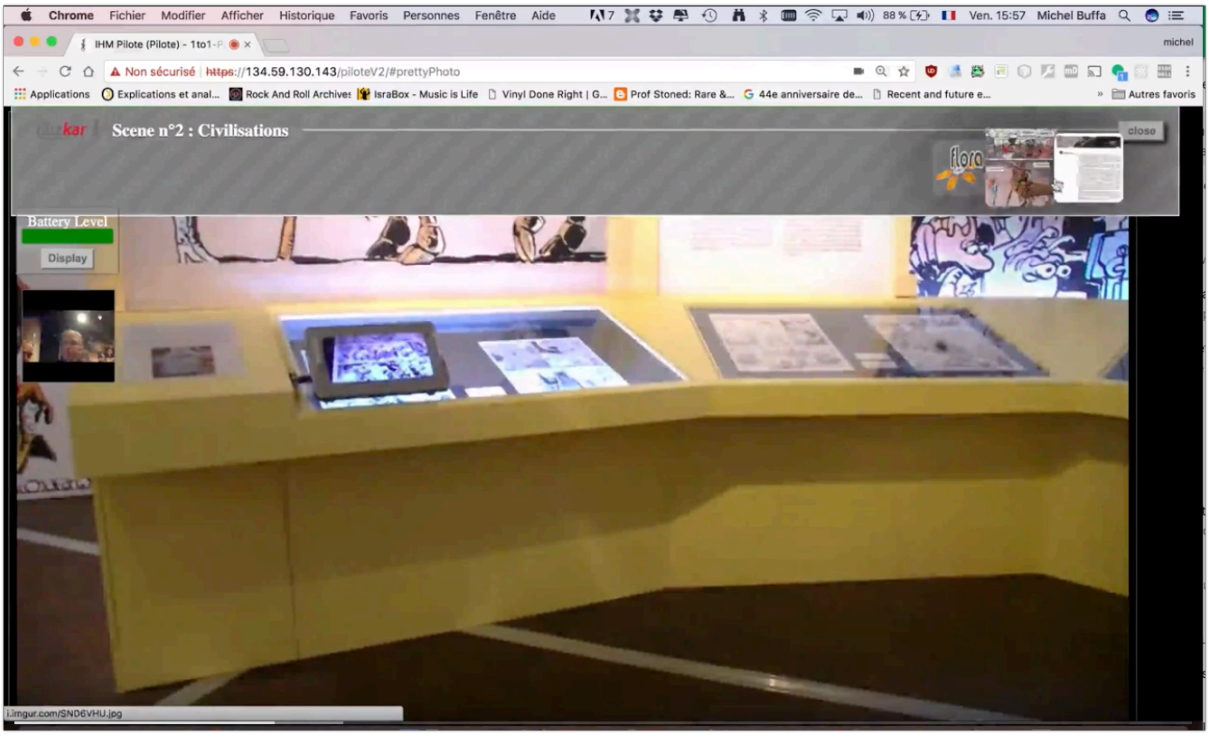


Figure 118. Le robot atteint une scène, ici consacrée aux « civilisations » extra–terrestres dans le monde de Valérian.

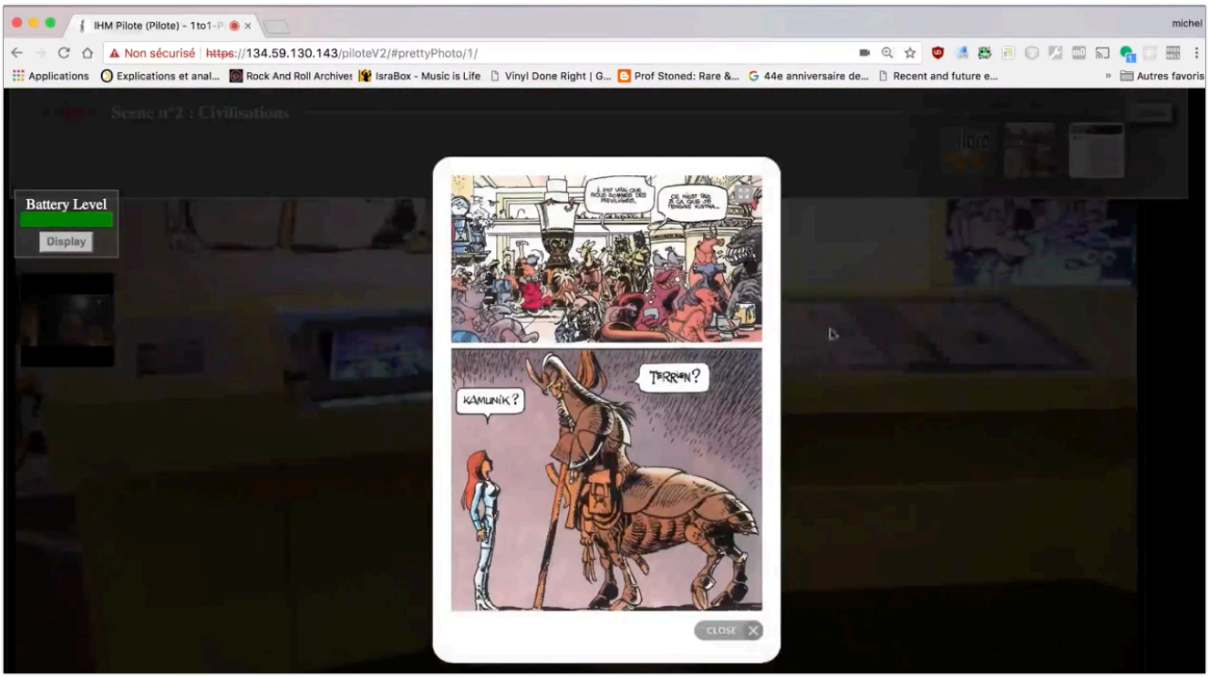


Figure 119. Un clic sur un des documents liés à la scène affiche une image HD de l'illustration située en face du robot (on la distingue dans l'image vidéo de la figure précédente, sur la gauche).

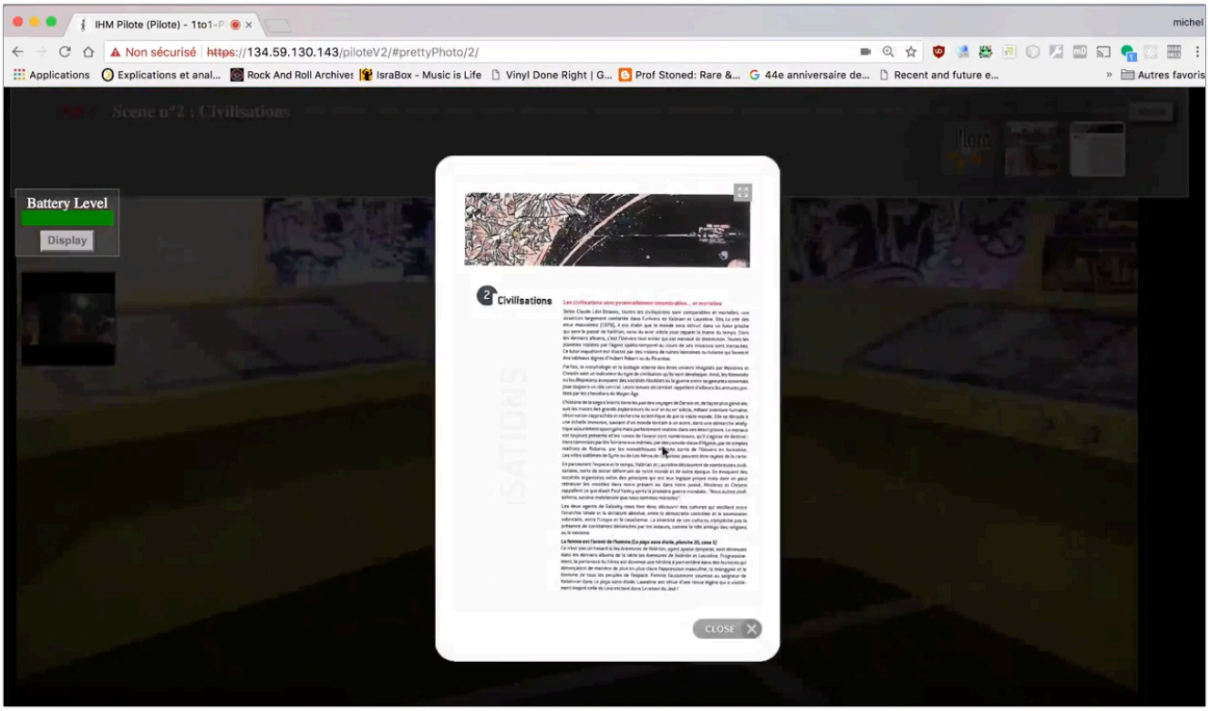


Figure 120. Un autre clic affichera une description détaillée à propos des civilisations, ou une vidéo YouTube, ou des informations issues de DBPedia.fr

8.6 Conclusion

Le projet est aujourd'hui en phase de pré-industrialisation. Les partenaires industriels étudient avec la BPI des modèles économiques pour exploiter un service de mise à

disposition de robots pour des visites à distance (visites automatiques, visites 1-1 type école mais aussi 1-N pour des visites *simultanées* dans plusieurs pays, en plusieurs langues, pour plusieurs publics)

Dans le cadre de l'accord entre le ministère de la culture et l'INRIA, une suite de ce projet est envisagée, notamment en poursuivant la collaboration avec les musées afin d'approfondir les scénarios d'utilisation, notamment en modélisant de nouvelles expositions et en effectuant de nouveaux tests utilisateurs des logiciels développés. L'objectif étant de proposer ces logiciels sous la forme d'une « boîte à outil open source » à destination des musées désirant décrire des visites robotisées.

L'utilisation d'une architecture connectée synchrone entre le robot et les visiteurs, mais aussi entre le robot et le triple store RDF distant, a montré la fiabilité et la réactivité qu'une telle solution peut apporter. Les technologies du Web sémantique se sont avérées bien adaptées à la modélisation de visites, et en augmentant les modèles, notamment avec des contraintes temporelles ou contextuelles, les capacités de raisonnement associées pourront permettre des visites reconfiguration dynamique plus souples (ex : on, prend du retard lors de la visite de la scène numéro 1, le robot peut suggérer d'aller directement à une autre scène, en fonction de la cible et du temps imparti on peut suggérer des parcours lors de la phase de création).

Le projet a permis de réaliser que les technologies du Web Sémantique sont viables et les outils suffisamment stables aujourd'hui pour des applications industrielles à forte contrainte (sécurité, disponibilité, réactivité). Par ailleurs, des modèles ontologiques simples, dans une architecture distribuée comme celle que nous avons mise en œuvre, remplacent avantageusement l'utilisation de bases de données classiques, tout en apportant extensibilité, souplesse et raisonnement. Le projet a été un excellent vecteur de promotion des technologies du W3C.

9. Web, Audio et sémantique pour l'interaction musicale (2017-)

Contexte

Les professionnels du streaming tels que Deezer, Spotify, Pandora ou Apple Music enrichissent l'écoute de musique avec par exemple la biographie des artistes et proposent des suggestions pour écouter d'autres chansons ou albums d'artistes identiques ou similaires. Un journaliste ou un présentateur radio utilise le Web pour préparer ses programmes. Un professeur d'une école d'ingénieurs du son utilisera des interviews de producteurs ou d'artistes, des retours d'expérience de producteurs célèbres, des logiciels d'analyse, des effets, des tables de mixage audio pour expliquer les techniques de production à ses étudiants.

Ces trois scénarios ont une chose en commun : ils utilisent des bases de connaissances allant de la plus empirique --- le résultat d'une recherche par mot-clé dans Google --- à la plus formalisée et utilisable par le biais d'APIs REST, comme celles proposées par Spotify, LastFM, MusicBrainz, DBPedia ou encore l'API d'analyse audio de The Echo Nest.

En tenant compte de cela, le besoin de bases de connaissances musicales plus précises et d'outils pour explorer et exploiter ces connaissances devient évident.

Contributions

J'ai initié le projet de recherche WASABI[105] (**W**eb **A**udio **S**emantic **A**ggregated in the **B**rowser for **I**ndexation) aujourd'hui financé par l'ANR. Avec l'équipe que je supervise chez WIMMICS, nous avons créé une base de connaissance de métadonnées musicales sur des chansons commerciales pop, rock, variété, reggae, blues, etc.

La base de connaissance contient aujourd'hui plus de deux millions de chansons, plus de deux cent mille albums et plus de soixante-dix-sept mille artistes.

Elle contient la plupart des chansons commerciales *chantées* connues, hors musique classique. Par exemple, tous les classiques du rock, pop, blues, reggae, rap, folk, variété etc. figurent dans la base. J'ai participé à la conception et au développement des différents outils logiciels mis en œuvre pour la constitution de cette base.

La base de données WASABI est hybride et conjugue NoSQL et graphes RDF. Nous sommes actuellement en train de définir les vocabulaires pour décrire de manière riche les contenus dont nous disposons (données culturelles mais aussi issues de l'analyse MIR audio et de l'analyse des paroles en langage naturel).

[105] Site web du projet : https://wasabihome.i3s.unice.fr, base de données en ligne : https://wasabi.i3s.unice.fr/

Collaboration, encadrement

Je suis le coordinateur national du projet ANR WASABI, qui a pour partenaires I3S, IRCAM, DEEZER, Parisson et Radio France.

Je supervise et coordonne une équipe de deux ingénieurs, un apprenti, quatre chercheurs et un thésard au sein de WIMMICS. Je conçois et développe aussi une partie importante des logiciels livrables de ce projet.

Publications principales ayant résulté de ces travaux

Note : ne figurent ici que les publications relatives à la partie « base de connaissance » du projet ANR WASABI. J'ai surtout publié pour WASABI dans le domaine des applications clientes WebAudio présenté dans les deux chapitres suivants de ce manuscrit. Nous sommes en train de finaliser la base de graphes RDF et les ontologies pour le projet. Des publications vont suivre. Le projet n'a que 15 mois (sur 42).

Gabriel Meseguer-Brocal, Geoffroy Peeters, Guillaume Pellerin, Michel Buffa, Elena Cabrio, et al. *WASABI: a Two Million Song Database Project with Audio and Cultural Metadata plus WebAudio enhanced Client Applications.* Web Audio Conference 2017 – Collaborative Audio #WAC2017, Aug 2017, London, United Kingdom. 2017. 〈hal-01589250〉

Michel Buffa, Amine Hallili, Philippe Renevier. "*MT5: a HTML5 multitrack player for musicians"*. - 1st Web Audio Conference, January 26-28, 2015 - IRCAM & Mozilla Paris, France, May 2015. 〈hal-01150455〉

Résumé de cette section

Mes travaux relatifs à la construction d'une base de connaissances hybride [36] (NoSQL et sémantique) ont pour cadre le projet ANR WASABI[106], démarré début 2017. Les partenaires de ce projet sont le laboratoire I3S de l'Université Côte d'Azur, IRCAM, DEEZER, et la société Parisson. D'autres collaborateurs sont Radio France (journalistes, archivistes), des compositeurs de musique, des musicologues, des écoles de musique et d'ingénieurs du son (classiques ou en ligne).

L'objectif principal de ce projet multidisciplinaire est de construire une base de connaissances de 2 millions de chansons qui contient des métadonnées recueillies à partir de bases de données musicales sur le Web (artistes, discographie, producteurs, année de production, etc.), mais aussi obtenues à partir de l'analyse des paroles de chansons (de quoi parlent-elles ? Des personnes ou des lieux sont-ils mentionnés ? Quelles sont les émotions qu'elles transmettent ? Quelle est la structure des paroles de la chanson - couplet, refrain, etc. ?), et obtenues à partir de l'analyse audio : genre, orchestration, tempo, mesures, tonalités, volume, accords, structure, détection de plagiats, séparation des sources / démixage, etc.

J'ai conçu et développé (avec l'équipe que je supervise) cette base de connaissance et les méthodes, modèles et outils qu'elle nécessite pour l'intégration de données. Nous avons

[106] Site web du projet : https://wasabihome.i3s.unice.fr/

commencé par collecter sur le Web plus de 2 millions de paroles de chansons commerciales, puis nous avons petit à petit intégré d'autres sources de données (MusicBrainz, DBPedia, iTunes, GoEar, EquipBoard, etc.). Une version de la base de données WASABI est déjà en ligne et est enrichie tout au long du projet[107].

Initialement cette base contenait uniquement des données structurées stockées dans une base NoSQL de type MongoDB/Elastic Search, mais nous sommes en train d'enrichir sémantiquement ces données en les dupliquant dans une base sémantique RDF de type Virtuoso, en les reliant à des ontologies et données liées existantes ou que nous développons au fur et à mesure de l'avancement du projet.

Nous disposons donc d'une base hybride permettant de répondre très rapidement à des requêtes simples ("*tous les disques de Michael Jackson*") mais également à des requêtes complexes exprimées en SPARQL dont le temps de réponse peut être beaucoup plus important, et qui peuvent faire appel à des inférences (ex: *"Quel est le rapport en RadioHead et Pink Floyd?", "Quelles sont les influences de Black Sabbath et qui ont-ils influencé ?", "Quels sont les événements de l'actualité qui apparaissent dans les chansons de U2 ?"*)

Mélanger le "culturel" et le "signal" audio

Le projet WASABI a l'originalité d'utiliser à la fois des algorithmes travaillant sur le signal audio, issus du domaine de recherche MIR (Music Information Retrieval), des algorithmes exploitant le Web de données pour agréger des données collectées depuis de multiples sources (DBPedia, MusicBrainz, etc.) et des algorithmes d'analyse de texte en langage naturel, pour produire une base de connaissances plus cohérente.

Ces algorithmes, et cela est une des originalités de ce projet, fonctionnent souvent ensembles (ex : détection de structure, détection d'émotions, détection de plagiat mélangent audio et texte, la séparation de sources/démixage exploite les données collectées sur le Web de données comme la composition d'un groupe de rock, les instruments utilisés etc.)

Des applications clientes sont alors développées pour les différents "utilisateurs cibles" du projet WASABI. Il est à noter que plusieurs scénarios impliquent des applications relevant du domaine de la MAO (Musique Assistée par Ordinateur), domaine quasiment absent du Web au moment du démarrage du projet.

Je me suis ainsi investi depuis 2013 dans le standard W3C WebAudio (je participe activement au groupe de travail WebAudio du W3C, voir les deux chapitres suivants de ce manuscrit pour le détail de mes contributions). Ainsi, j'ai conçu, avec les deux ingénieurs et l'apprenti recrutés par notre équipe sur ce projet, de nombreuses applications WebAudio interactives (table de mixage, instruments virtuels et effets audio temps réels pour musiciens/ingénieurs du son) qui profitent de l'énorme quantité de connaissances hétérogènes intégrées à la base de données WASABI, l'idée étant de relier des instruments/effets aux chansons de la base.

[107] https://wasabi.i3s.unice.fr/

Les bases de données musicales du Web peuvent être utilisées pour extraire des données culturelles, reliant la musique à des éléments tels que le producteur, le studio d'enregistrement, le compositeur, l'année de production, les membres du groupe, etc.

Les données en texte libre telles que les paroles de chanson ou le texte de pages web (blogs de passionnés, pages officielles d'artistes, etc.) liées à une chanson peuvent être utilisées pour extraire des données non explicites telles que les sujets, les lieux, les personnes, les événements, les dates ou même les émotions transmises. L'analyse du signal audio quant à elle permet d'extraire différents types de descripteurs pour la musique tels que le tempo, le volume, les accords, les émotions, le genre et la structure d'une chanson.

Ces extractions sont sujettes aux erreurs et aux incertitudes et nous visons à améliorer leur qualité en les combinant avec les connaissances recueillies à partir des sources de données du Web et avec les informations issues de l'analyse des paroles de chansons.

Par exemple la structure temporelle, la présence et la caractérisation de la voix, les émotions, la détection de plagiats, le démixage à partir d'une version stéréo de la chanson sont des données qu'on espère mieux calculer en associant le « culturel » et « l'audio ».

J'ai participé également, avec des membres de l'équipe, pendant six mois, à l'analyse des besoins et à l'identification des différents cas d'utilisation grâce à la présence d'utilisateurs de nos futurs résultats de recherche : Deezer, Radio France, journalistes musicaux, compositeurs, professeurs de musique et musicologues.

Je suis très impliqué dans le développement et l'intégration de briques logicielles open source et de web services pour :

- La *visualisation* des métadonnées audio (obtenues à partir d'algorithmes MIR - Music Information Retrieval), lecteur audio multipiste dans le navigateur à l'aide de l'API WebAudio (mixage en temps réel, effets audio, analyse audio),
- Le traitement automatique des paroles de chansons, reconnaissance et liaison des entités nommées, annotation et édition collaborative,
- L'accès à des services Web avec une API donnant accès à l'ensemble du corpus et permettant de manipuler l'ensemble des données. La modélisation à l'aide d'ontologies spécialisées permet/permettra d'effectuer des raisonnements (ex : "quelles sont les influences de David Bowie entre 1974 et 1976 ? Quels sont les artistes avec qui il partage ces influences ?"
- Les applications relevant de la Musique Assistée par Ordinateur (MAO).

Ces briques logicielles sont utilisées pour développer des démonstrateurs formalisés avec nos partenaires et collaborateurs en utilisant notamment le standard WebAudio et permettant le développement d'applications musicales accessibles au public à partir d'un navigateur Web.

Ce chapitre du manuscrit se concentrera principalement sur les points du projet que j'ai supervisés relatifs à la construction de la base de connaissance :

- Constitution de la base de données initiale par agrégation de données issues du Web de données : identification des sources, développement de scripts pour collecter ses données en très grand nombre, désambiguïsation, détection de conflits, curation, développement d'une API REST et utilisation conjointe d'ElasticSearch pour autoriser les recherches avec orthographe relâchée.
- Présentation de la structure logicielle de l'application développée (bases de données hybrides).
- Identification des ontologies utilisées (et nous en développons de nouvelles au cours du projet, le projet n'est pas encore à mi-parcours) et description des outils de migration MongoDB vers RDF dans un triple store VIRTUOSO, problèmes rencontrés.
- Perspectives futures (ajout du moteur sémantique CORESE pour les scénarios impliquant du raisonnement sur des demandes à la carte, notamment de la part de journalistes musicaux).

9.1 Construire une base de connaissance de chansons commerciales populaires

Les professionnels du streaming tels que Deezer, Spotify, Pandora ou Apple Music enrichissent l'écoute de musique avec leur biographie et proposent des suggestions pour écouter d'autres chansons ou albums d'artistes identiques ou similaires. Un journaliste ou un présentateur radio utilise le Web pour préparer ses programmes. Un professeur d'une école d'ingénieurs du son utilisera des outils analytiques pour expliquer les techniques de production à ses étudiants.

Ces trois scénarios ont une chose en commun : ils utilisent des bases de connaissances allant de la plus empirique --- le résultat d'une recherche par mot-clé dans Google --- à la plus formalisée et utilisable par le biais d'une API REST --- comme l'utilisation par Spotify de LastFM, MusicBrainz, DBPedia et des extracteurs audio de The Echo Nest.

Ensuite, le besoin de bases de connaissances musicales plus précises et d'outils pour explorer et exploiter ces connaissances devient évident.

Le projet WASABI (**W**eb **A**udio **S**emantic **A**ggregated in the **B**rowser for **I**ndexation), démarré début 2017, est un projet de 42 mois fondé par l'Agence Nationale de la Recherche (ANR) qui vise à répondre à ce besoin. Les partenaires de ce projet sont le laboratoire I3S de l'Université Côte d'Azur, l'IRCAM, DEEZER et la société Parisson. D'autres collaborateurs sont Radio France (journalistes, archivistes), compositeurs de musique, musicologues, écoles de musique, écoles de musique, écoles d'ingénieurs du son.

L'objectif principal de ce projet multidisciplinaire est de construire une base de connaissances de 2 millions de chansons qui contient des métadonnées recueillies à partir de bases de données musicales sur le Web (artistes, discographie, producteurs, année de production, etc.), à partir de l'analyse des paroles de chansons (De quoi parlent-elles ? Des lieux ou les personnes connues sont-ils mentionnés ? Quelles sont les émotions

qu'elles transmettent ? Quelle est la structure des paroles de la chanson ? Est-ce un plagiat ?), et de l'analyse audio (tempo, volume, accords, structure, détection de plagiait, séparation des sources / démixage, etc.).

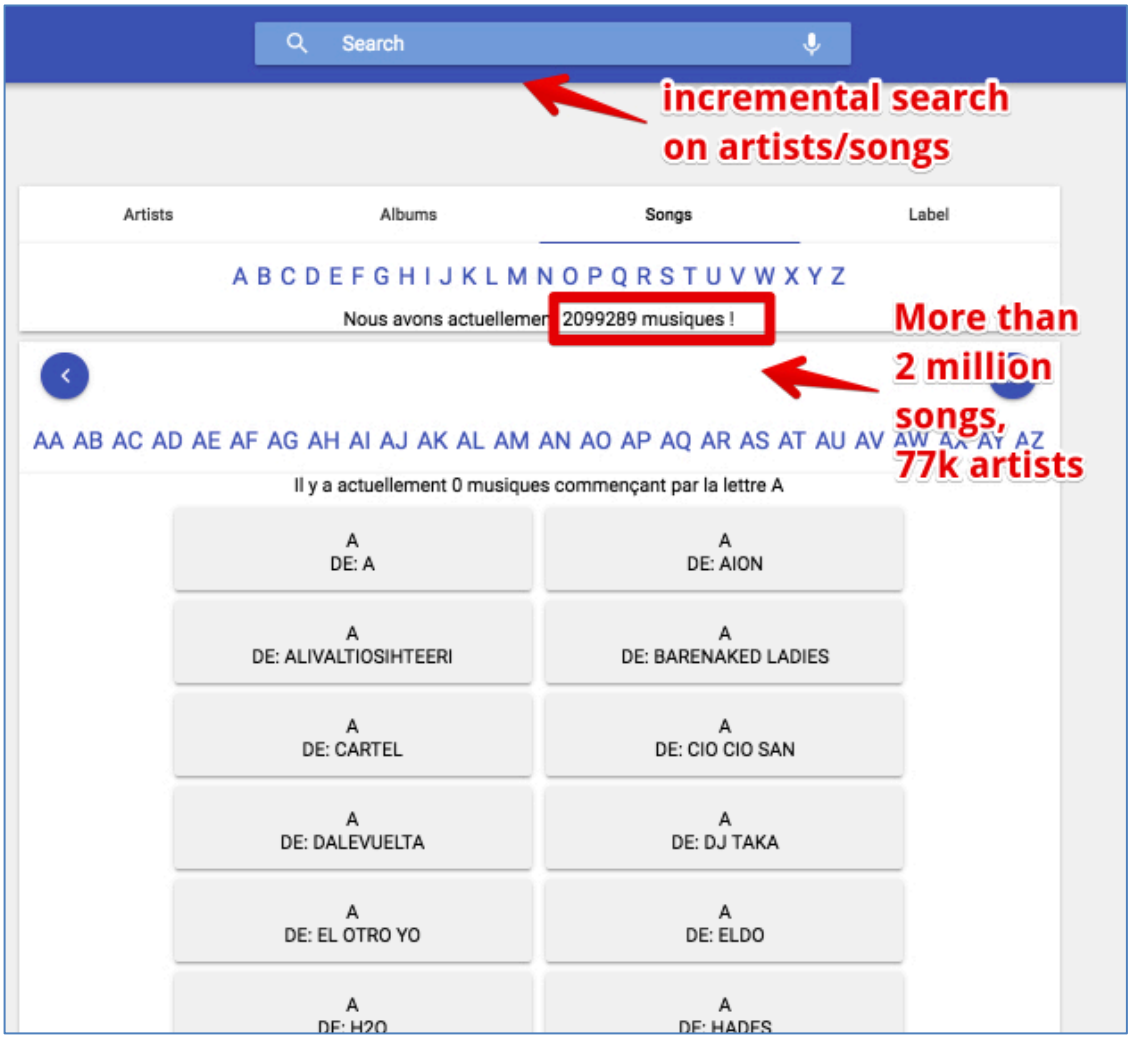


Figure 121. Interface de recherche de la base de données WASABI.

Une version préliminaire de la base de données WASABI est déjà en ligne et est enrichie tout au long du projet[108].

La Figure 121 et la Figure 122 donnent un aperçu de l'interface graphique principale d'exploration de la base de données WASABI (recherche par titre partiel de chanson et/ou d'artiste, navigation par facettes).

La base de données est livrée avec une API REST complète et comprend un SPARQL endpoint[109].

[108] https://wasabi.i3s.unice.fr/
[109] https://wasabi.i3s.unice.fr/apidoc/

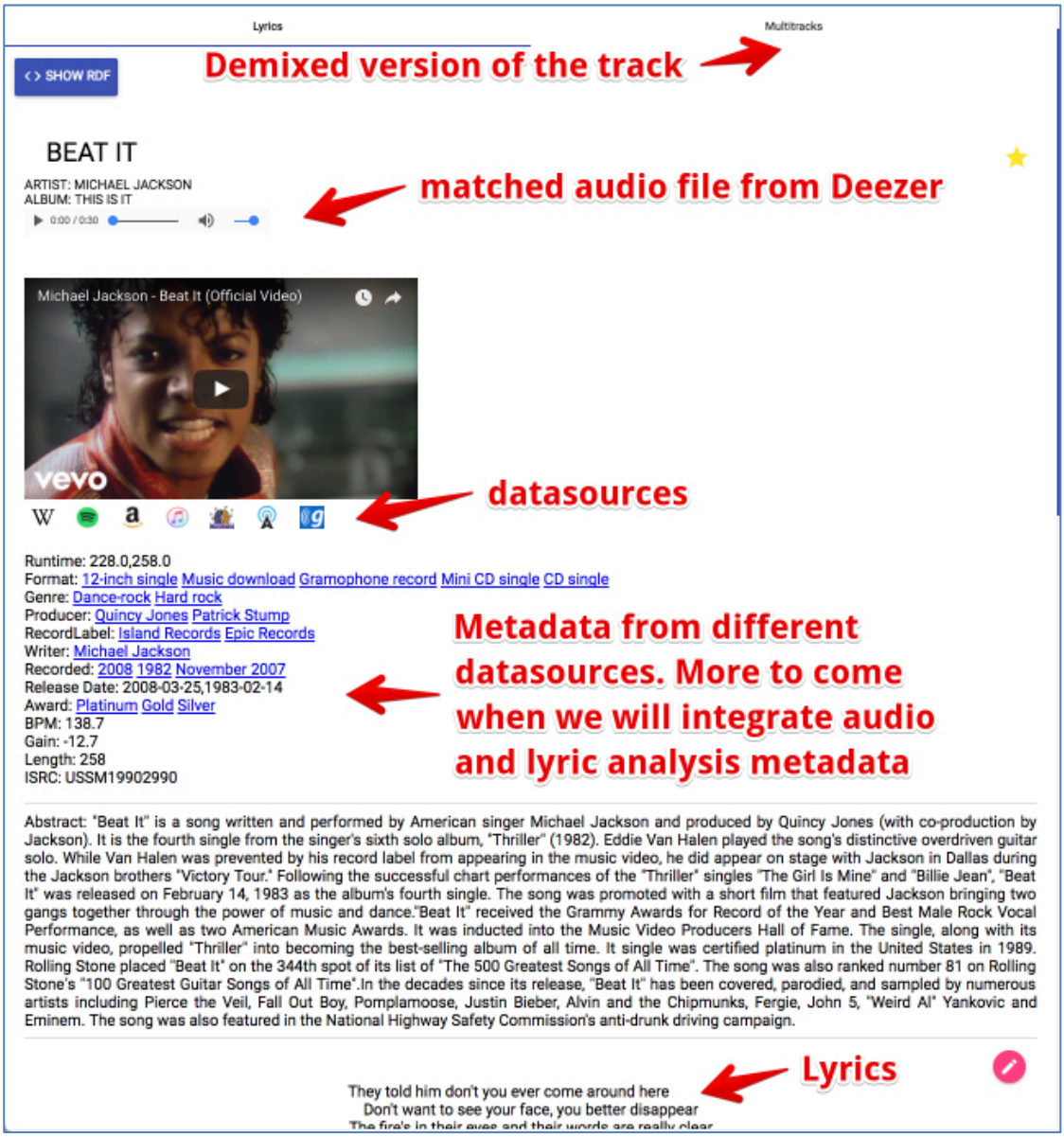


Figure 122. Un exemple de chanson dans la base WASABI (qui en contient plus de 2 millions)

Le projet WASABI a l'originalité (1) d'utiliser les algorithmes du domaine de recherche MIR (Music Information Retrieval) qui travaillent sur le contenu audio mais aussi d'exploiter aussi (2) les données collectées depuis le Web de données et (3) les algorithmes d'analyse de texte en langage naturel (NLP) sur les paroles pour produire une base de connaissances plus riche et cohérente. Les applications clientes WebAudio profiteront alors de cette énorme quantité de connaissances hétérogènes intégrées pour associer aux chansons, par exemple, des instruments préréglés avec des sons proches de ceux utilisés dans les chansons, une table de mixage pour remixer les différents instruments, ou des outils interactifs d'annotation pour musicologues.

La base de données est *NoSQL et RDF*. En exploitant les technologies du web sémantique pour typer et relier les métadonnées, il est possible d'extraire des données culturelles, reliant la musique à des éléments tels que le producteur, le studio d'enregistrement, le

compositeur, l'année de production, les membres du groupe, mais aussi des données impliquant des inférences comme des calculs d'influence par exemple.

Les paroles de chansons ou le texte des pages liées à une chanson peuvent être utilisées pour extraire des données non explicites telles que les sujets, les lieux, les personnes, les événements, les dates ou même les émotions transmises.

L'analyse du signal audio permet d'extraire différents types de descripteurs pour la musique tels que le tempo, le volume, les accords, les émotions, le genre et la structure d'une chanson.

Ces extractions sont sujettes aux erreurs et aux incertitudes et nous visons à les améliorer en combinant l'information extraite avec les connaissances extraites du Web sémantique et de l'analyse des paroles des chansons. Exemple de données que l'on pourra mieux calculer en associant audio et sémantique / données culturelles (à partir des paroles, à partir de l'audio) : la structure temporelle la présence et la caractérisation de la voix, les émotions, repérer les plagiats, faciliter le démixage (si on dispose d'une information telle que l'orchestration : qui joue sur la chanson et de quel instrument, nous supposons que cela pourra servir à améliorer les algorithmes de séparation des instruments audio). Une fois les pistes séparées, une analyse audio plus précise pourra être effectuée sur chaque piste et des données plus riches pourront être extraites (notes, instruments, type de réverbération, etc.).

Le projet WASABI spécifie les différents cas d'utilisation grâce à la présence d'utilisateurs de nos futurs résultats de recherche : Deezer, Radio France, journalistes musicaux, compositeurs et musicologues. Pour ce faire, WASABI propose(ra) une suite de progiciels libres et de services de données ouvertes pour les noms de domaine :

- La visualisation des métadonnées audio à partir d'analyses MIR, l'écoute de chansons multipistes dans le navigateur Audio (mixage en temps réel, effets audio, etc.),
- Le traitement automatique des paroles de chansons, reconnaissance des entités nommées, détection de structure, annotation et édition collaborative, etc.
- L'accès aux services Web avec une API offrant un environnement d'étude de similitude musicale combinant des analyses audio et sémantique.

Ces briques logicielles seront utilisées pour développer des démonstrateurs formalisés avec nos partenaires.

Le reste du document est organisé comme suit : la section 9.2 présente le contexte du projet WASABI et des travaux connexes, la section 9.3 présente les questions de recherche et les objectifs du projet, la section 9.4 présente les premiers résultats du projet et enfin nous conclurons par une discussion sur la suite du projet dans la section 9.6.

9.2 Contexte et état de l'art

Dans le domaine de l'extraction de l'information musicale à partir du signal audio, il y a peu de travaux antérieurs mélangeant, l'analyse des paroles, des métadonnées collectées sur le web de données et l'audio dans son ensemble.

Le projet Million Song Dataset a procédé à une classification basée principalement sur des données audio [Bertin et al. 2011 [171]], mais n'a pas profité des données structurées disponibles par exemple sur DBpedia, pour lever certaines incertitudes. Des informations telles que la composition de groupe ou l'orchestration peuvent être très pertinentes pour informer les algorithmes de démixage, mais ne sont disponibles que dans certaines sources de données (BBC, MusicBrainz…) et pour de nombreux artistes peu connus, cette information n'est pas disponible. C'est ici que la collaboration entre l'audio et la sémantique trouve son sens, l'un profitant de l'autre.

Des centres de recherche tels que le C4DM de l'Université Queen Mary de Londres ont développé des collaborations avec la BBC dans l'utilisation de RDF et des ontologies musicales proposées dans plusieurs domaines (y compris les effets audio et l'organologie - voir le projet semanticaudio.ac.ac.uk -). D'autre part, des sociétés telles que Spotify (ayant acquis The Echo Nest), Grace Note, ou Pandora (ayant récemment engagé plusieurs chercheurs dans le domaine de l'analyse audio), Apple Music (qui vient de créer son équipe de recherche audio à Londres) ont développé une certaine expertise dans ce domaine mais qui n'est malheureusement pas rendue au domaine public.

MusicWeb et MusicLynx (Allik et al. 2016 [172] [173]) relient les artistes musicaux au sein d'une application Web pour découvrir les connexions entre eux et fournit une expérience de navigation à l'aide de connexions extra-musicales ou tangentielles à la musique. Il intègre des métadonnées sémantiques liées ouvertes provenant de diverses sources de recommandations musicales et de données de médias sociaux, dont DBpedia.org, sameas.org, MusicBrainz, AcousticBrainz, Last.FM, YouTube, etc. ainsi que des informations dérivées du contenu. Le projet partage quelques idées avec WASABI, mais ne semble pas aborder la même échelle de données, et n'effectue pas d'analyses ad hoc sur le contenu audio (il réutilise les résultats disponibles sur AcousticBrainz) et les paroles.

Le projet WASABI est situé à une plus grande échelle que le Million Song Dataset (le double ! Les données audio sont fournies par Deezer). Il s'appuie également sur les résultats et les développements de deux projets antérieurs les projet ANR WAVES[110] et DIADEMS[111], ces derniers utilisant la plateforme musicale open source de Telemeta [174].

A noter que des collaborations avec des chercheurs du projet FAST (Johan Pauwels, Mark Sandler) sont en cours. Ce projet vise a pour but la « Fusion des technologies audio et sémantique » pour une production et une consommation musicales intelligentes ».

[110] http://wave.ircam.fr/
[111] https://www.irit.fr/recherches/SAMOVA/DIADEMS/

WASABI réutilise des ontologies crées par le projet FAST (par exemple la Multitrack Ontology [220], ou la Audio Effect Ontology [219]) et le projet FAST va utiliser des parties du dataset WASABI.

9.3 Questions et objectifs de recherche

9.3.1 Identification et évaluation des sources de données sémantiques et textuelles/culturelles/audio

9.3.1.1 Sources de données pour enrichir la base de données initiale

Cet objectif scientifique vise à alimenter les algorithmes MIR avec des données provenant du Web de données et permettant d'amorcer les paramètres initiaux (par exemple, si l'on connaît le producteur, le profil de l'artiste, les membres du groupe et les instruments qu'ils ont utilisés sur une chanson particulière, nous pouvons rechercher un signal de guitare directement

Le cœur du projet WASABI est sa base de connaissances, qui est enrichie tout au long du projet, à la fois avec de nouvelles métadonnées et avec de nouveaux modèles/ontologies.

Nous avons identifié plusieurs sources de données. Parmi elles, nous pouvons distinguer :

- Sources privilégiées en ligne : pour commencer la base de données Deezer est une agrégation de données provenant de différents labels musicaux commerciaux (40M chansons) ; elle est à jour mais souffre d'un manque de cohérence : le niveau de détails change beaucoup d'un label musical à l'autre, les variations sémantiques sont importantes, la classification est inexacte (les termes tels que " pop ", " rock ", " electro" sont trop génériques), et produit beaucoup de collisions de métadonnées (par exemple des homonymes entre artistes). On trouve également le « Million Song Dataset » qui est riche en métadonnées audio, mais contient très peu de métadonnées culturelles. Le Linked Data Catalog[112], annuaire en ligne de *datasets* fait également référence à d'autres ensembles de données mais qui sont pour la plupart datés et peu importants comparées aux deux sources précitées.
- Bases de données en ligne avec une API REST ou avec un SPARQL endpoint : DBPedia (différentes versions de chaque pays), musicbrainz.org (encyclopédie musicale gratuite, qui recueille des métadonnées musicales) et AcousticBrainz (données MIR calculées par *crowdsourcing*), last.fm et libre.fm, seevl.fm, musicmight.com (base de données hard rock/metal), discogs.com (l'une des plus grandes bases de données en ligne), soundfacts.org, equipboard.com (instruments et matériel utilisés par les artistes célèbres) etc.

Ces bases de données fournissent des métadonnées chansons/artistes avec beaucoup de variabilité dans leur contenu : certains genres musicaux sont plus représentés que d'autres, différents chapitres de DBpedia peuvent avoir des contenus complémentaires ou

[112] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/

conflictuels (il arrive que DBPedia.org propose plus de contenu que DBPedia.it même à propos d'un groupe italien).

Figure 123. Exemple d'entrée pour un artiste. Ici Metallica.
Les icones représentent les sources de données utilisées.

Un défi important dans ce projet est l'identification des métadonnées pertinentes, leur fusion et la résolution des conflits (noms de personnes avec différentes abréviations, etc.),

et leur conservation. Le projet WASABI prévoit de comparer régulièrement les résultats des analyses combinées audio/culturel avec les résultats publiés par certaines de ces sources de données (AcousticBrainz ou Million Song Dataset par exemple).

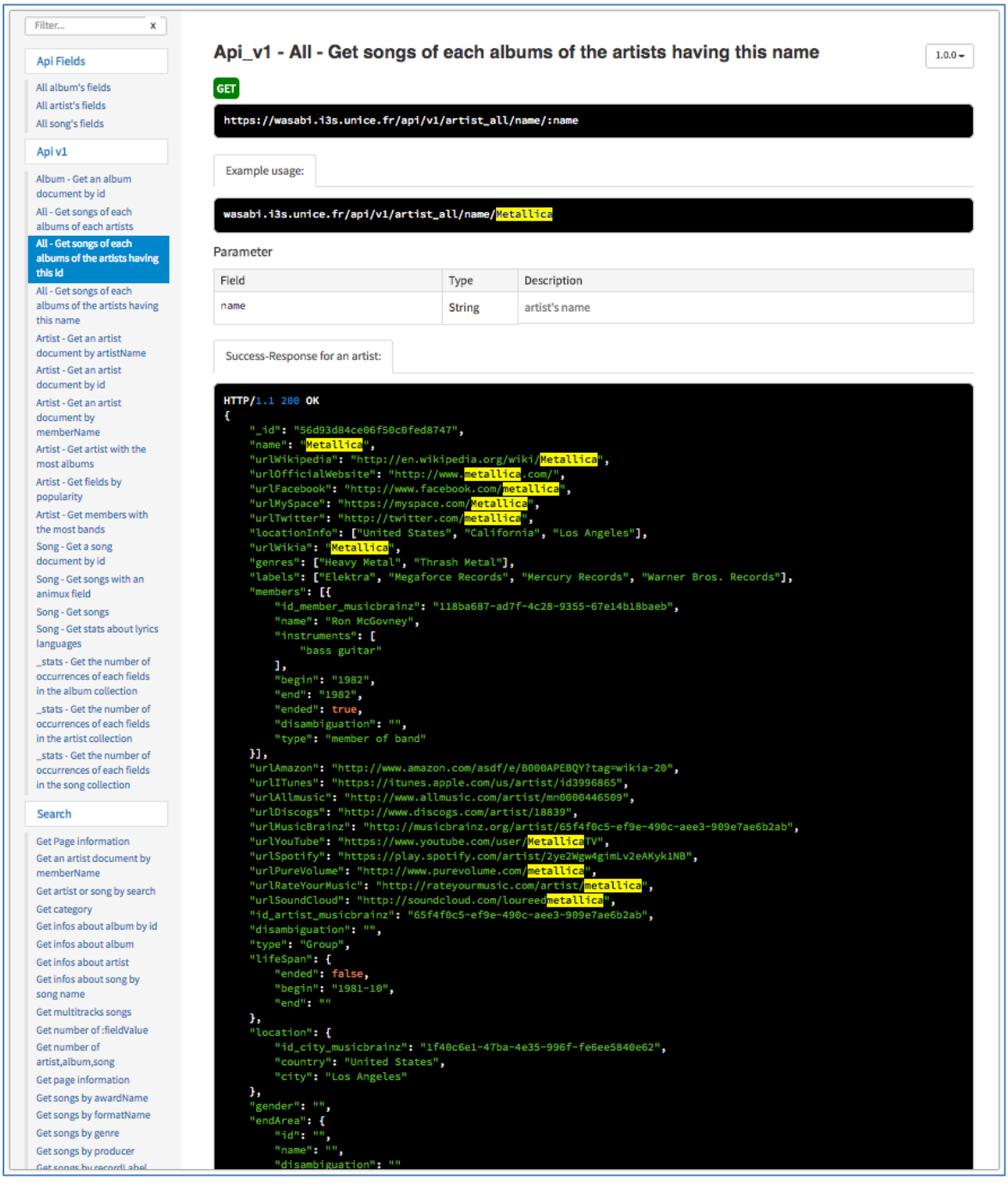


Figure 124. Exemple de résultats fournis par l'API (https://wasabi.i3s.unice.fr/api/v1/artist_all/name/Metallica).
Ici informations sur Metallica. On voit la liste des URIs des sources de données, l'origine géographique, etc.

9.3.1.2 Sources de données audio

Nous nous appuyons principalement sur les fichiers audio stéréo fournis par Deezer mais aussi sur un ensemble de morceaux multipistes provenant d'écoles d'ingénieurs du son,

de studios d'enregistrement et des archives de Radio France (des milliers d'artistes enregistrés en direct sur plusieurs pistes). Nous utilisons également les données audio de 7digital et de YouTube pour compléter le jeu de données.

L'appariement de ces titres audio (2 millions de pistes stéréo et environ 2 000 pistes non mélangées) avec les entrées actuelles de la base de données WASABI est une tâche permanente. Nous avons déjà mis en correspondance 87 % des chansons présentes dans la base WASABI avec les fichiers audio fournis par Deezer. Nos résultats préliminaires sont décrits plus en détail dans la section 9.4.

Les deux millions de chansons ont été choisies parmi le catalogue Deezer à partir de la disponibilité des paroles collectées depuis le site LyricWiki[113].

9.3.1.3 Vocabulaires

Le projet WASABI utilise un ensemble de vocabulaires / ontologies complémentaires, qui sera complété au fur et à mesure de l'avancée du projet.

Pendant les 12 premiers mois, nous avons conduit des entretiens avec les utilisateurs finaux, afin de définir leurs besoins, tandis qu'un état de l'art sur les ontologies existantes et sur les modèles utilisés dans l'industrie musicale (pas forcément sous forme d'ontologies) a été réalisé.

Nous adoptons une approche incrémentale axée sur les besoins des utilisateurs : nous intégrons de nouvelles métadonnées et mettrons à jour les vocabulaires que nous utilisons au fur et à mesure que de nouveaux besoins apparaissent (directement des utilisateurs, ou spécifiques aux outils que nous développons).

En particulier, nous avons déjà identifié les besoins suivants qui sont (ou vont être) modélisés dans le vocabulaire WASABI :

- La définition « culturelle d'une chanson » (auteur, compositeur, producteur, album, musiciens, date de sortie, URIs des sources de données etc.)
- La segmentation d'une piste musicale dans sa structure temporelle (vers, chœur, etc.),
- La segmentation d'une piste musicale en parties chantées et la caractérisation de sa qualité vocale,
- L'identification des plagiats,
- L'utilisation d'informations préalables (comme la composition organologique d'un groupe obtenu à partir du Web sémantique) pour aider les algorithmes de séparation des sources,
- L'utilisation de l'audio multicanal (soit propre ou estimé) pour améliorer l'estimation du contenu.
- La définition des données extraites de l'analyse des paroles (section suivante).
- La définition des effets audio utilisés ou de techniques d'enregistrement particulières.

[113] http://lyrics.wikia.com/

Ce projet est aussi l'occasion de travailler sur la définition d'une ontologie pour décrire efficacement une chanson dans le temps, et à notre connaissance, nous n'avons pas trouvé d'ontologies avec ce niveau de granularité dans les modèles de description.

Les articles d'Yves Raimond et al. de la BBC sont des précurseurs sur la façon dont on peut construire un jeu de données culturelles sur la musique [175]. Un important projet de recherche européen, le projet SAFE[114] aborde également ce sujet (Allik et al. [176] Thalmann et al. 2016 [177]. La spécification de la Music Ontology[115] fournit les principaux concepts et propriétés pour décrire la musique (i.e. artistes, albums, chansons, paroles, pistes), schema.org contient également un vocabulaire sur la musique, construit par les principaux fournisseurs de moteurs de recherche. Le projet SAFE a également créé une ontologie pour la classification des sons (chaud, flou, froid, etc.). L'IRCAM a développé plusieurs vocabulaires pour l'extraction d'éléments musicaux, en particulier pour la classification et l'étiquetage des genres, l'humeur, l'instrumentation/orchestration et l'analyse rythmique (métrique, tempo et segmentation temporelle). The Million Song Dataset a aussi développé des vocabulaires pour la classification des genres [178]

9.3.2 Analyse des paroles de chansons et consolidation avec les données de l'analyse audio

Les données textuelles telles que les paroles de chansons sont utilisées comme sources pour extraire des données implicites (comme la structure, les thèmes abordés par la chanson, les lieux, les personnes, les événements, les dates ou même les émotions transmises) à l'aide d'algorithmes de traitement du langage naturel (NLP). L'exploitation conjointe de ces connaissances, ainsi que l'information contenue dans le signal audio peut améliorer l'extraction automatique d'informations musicales, relatives par exemple à la détection de la structure d'une chanson, à la présence et à la caractérisation de la voix, à la détection émotions musicales, à l'identification de plagiats, ou même faciliter le démixage de la musique.

En ce qui concerne l'analyse des paroles de chansons, nous avons identifié les tâches suivantes que nous allons aborder en élargissant les approches existantes dans le traitement du langage naturel (NLP) et en les adaptant au domaine musical. Ce sont des tâches en cours, la première (détection de structure) a donné lieu à des publications de premier plan (Fell et al. [185] et Meseguer et al. [186])

1. **Détection de structure** : identification de la structure de la chanson (par ex. intro, couplet, refrain), suivant [179].
2. **Détection d'évènements** : analyse du texte de la chanson pour en extraire le contexte (références explicites ou implicites), ainsi qu'extraction d'entités nommées, d'emplacements géographiques et de références temporelles directement ou indirectement exprimées dans le texte. Par exemple, pouvoir lier le "We came down to Montreux" (de la chanson "Smoke on the Water" de Deep

[114] http://www.semanticaudio.co.uk/
[115] http://musicontology.com/specification/

Purple) au festival de jazz de Montreux. L'un des défis sera de traiter de l'utilisation abondante de la métaphore dans les paroles de texte, que nous aborderons en combinant des ressources les décrivant (dans la mesure du possible), et des approches NLP pour la détection des métaphores [180].

3. **Détection thématique** (*topic modelling*) : mise en œuvre de modèles probabilistes pour identifier des sujets ou des thèmes abstraits dans les paroles en établissant des relations entre un ensemble de documents et les termes qu'ils contiennent [179], [181].
4. **Détection d'émotions** (*sentiment analysis*) : classification des émotions dans les chansons, en utilisant à la fois la musique et les paroles des chansons d'une manière complémentaire. Nous testerons et adapterons des algorithmes de machine learning pour capturer des informations sur les émotions exprimées par le texte d'une chanson [182].
5. **Détection de plagiats** : mesure le degré de similitude des textes de deux chansons, et identification des structures récurrentes, paraphrases, plagiat multi-sources [183], [184]. Un aspect intéressant à explorer est la combinaison de mesures de similarité calculées à la fois sur le texte (similitude lexicale et syntaxique) et les caractéristiques audio en particulier pour détecter le plagiat entre les chansons dans différentes langues. Par exemple, s'il y a une grande similitude dans les caractéristiques audio, nous bénéficions de méthodes de traduction automatique pour calculer également la similitude textuelle entre les textes dans différentes langues. De plus, des ressources multilingues comme DBpedia pourraient être exploitées pour reconnaître la présence des mêmes entités dans les paroles.

9.3.3 Améliorer la recherche d'informations musicales en utilisant à la fois l'audio, les paroles et le Web sémantique.

L'une des principales questions scientifiques auxquelles WASABI tente de répondre est de savoir comment tirer profit de diverses sources de connaissances sur une piste musicale donnée pour en améliorer la description.

Dans WASABI, nous étudions l'utilisation de trois sources de connaissances : le contenu audio d'une piste musicale (souvent désigné par des informations sur le contenu), les paroles des pistes non instrumentales et les informations sémantiques tirées du Web (telles que DBpedia, MusicBrainz, Discogs ou autres).

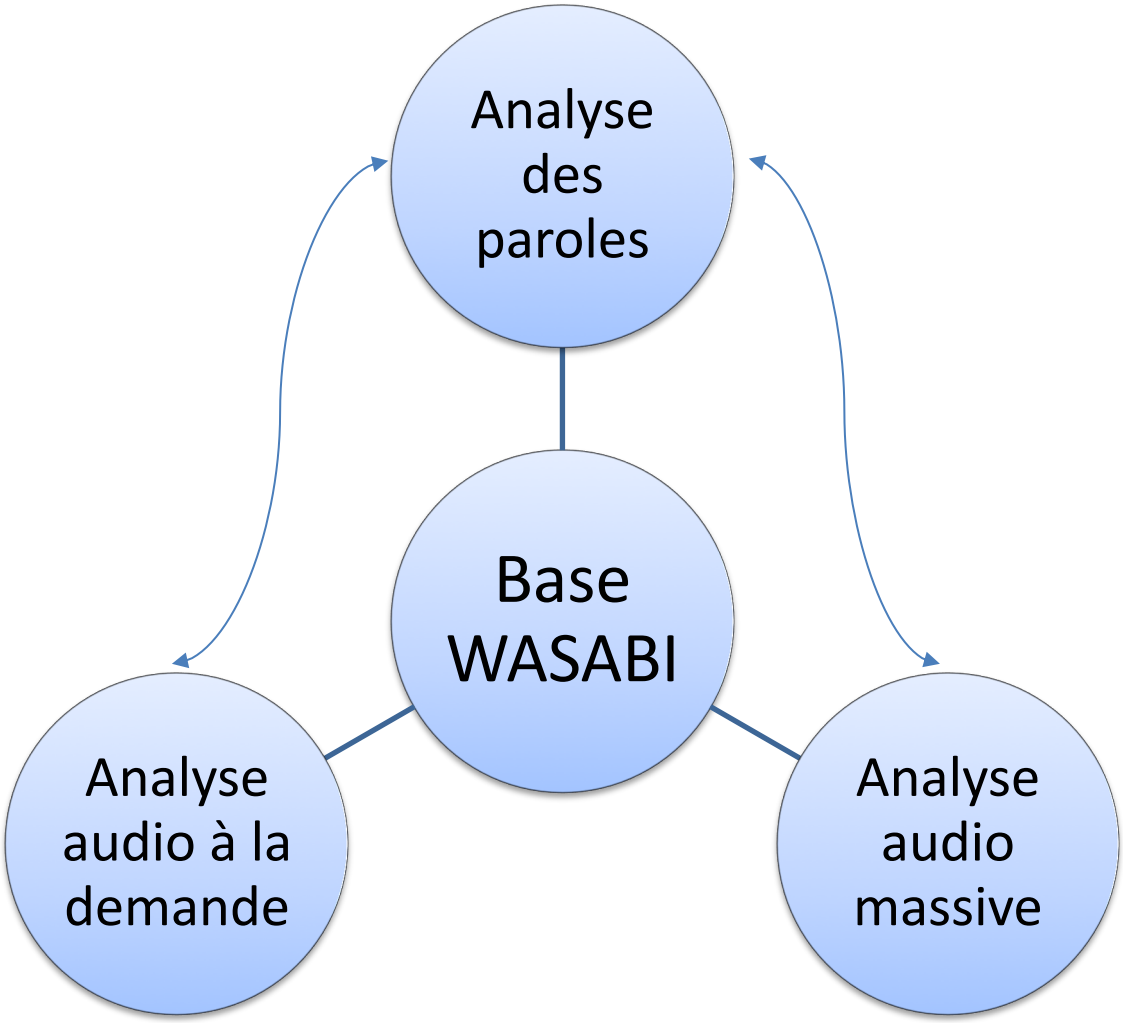


Figure 125. Ajout de métadonnées issues de l'analyse audio et de l'analyse des paroles. Certaines tâches d'analyse utilisent conjointement l'audio et les paroles.

Pour démontrer que l'utilisation des trois sources d'information est bénéfique, nous étudierons six cas différents d'utilisation de leur utilisation conjointe :

1. Estimation de la structure musicale (c'est-à-dire l'estimation de la position dans le temps du couplet, du chœur ou du pont) basée sur la répétition audio et la structure des paroles.
2. Segmentation de la voix au fil du temps et caractérisation basée sur le contenu audio (en utilisant les caractéristiques du chanteur) et la structure des paroles.
3. Estimation des émotions musicales à l'aide de l'audio, de l'analyse des paroles et du Web sémantique.
4. Identification de plagiat (lié à cette même détection à partir des paroles).
5. Séparation informée des sources audio (en utilisant les informations sur l'orchestration, les instruments joués, obtenues à partir du Web de données)

6. Reconnaissance d'instruments en utilisant des sources séparées (analyse audio sur pistes séparées)

De plus, l'ensemble de données non mixées (pistes audio séparées) fournit un environnement riche pour les problèmes MIR. A partir de ces pistes, il est possible d'avoir accès individuellement aux différents instruments ou même de spécifier des combinaisons de ceux-ci. Ce scénario est idéal pour améliorer des tâches telles que l'extraction de mélodie et d'harmoniques, l'extraction de voix chantée, etc. Ce sont des sous-problèmes de nos six cas d'utilisation.

9.3.4 Expérience utilisateur innovante

Nous avons de nombreux contacts avec des compositeurs, des musicologues, des journalistes de Radio France et de Deezer, qui souhaitent utiliser les technologies en cours de développement, aussi bien en back-office (pour les archivistes) qu'en front office (pour les utilisateurs finaux).

Parisson est en charge de la définition et l'utilisation d'interfaces multipistes pour les communautés de chercheurs et d'enseignants pour lesquelles il développe déjà des outils Web collaboratifs pour l'indexation audio automatique, semi-automatique et manuelle : la plateforme Telemeta (Figure 128). Cette plate-forme exploite le framework basé web TimeSide[116] pour l'analyse audio.

Wimmics/I3S, IRCAM et Parisson ont déjà publié ou présenté des démonstrations lors des WebAudio Conférences précédentes et ont produit des produits innovants avec cette technologie, qui seront inclus dans le projet WASABI (par exemple : le lecteur audio multipiste de la Figure 126 ou la simulation d'un amplificateur de guitare à lampes de la Figure 127). Les chapitres suivants de ce manuscrit décrivent les travaux que j'ai menés dans ce domaine depuis 2013.

[116] https://github.com/Parisson/TimeSide

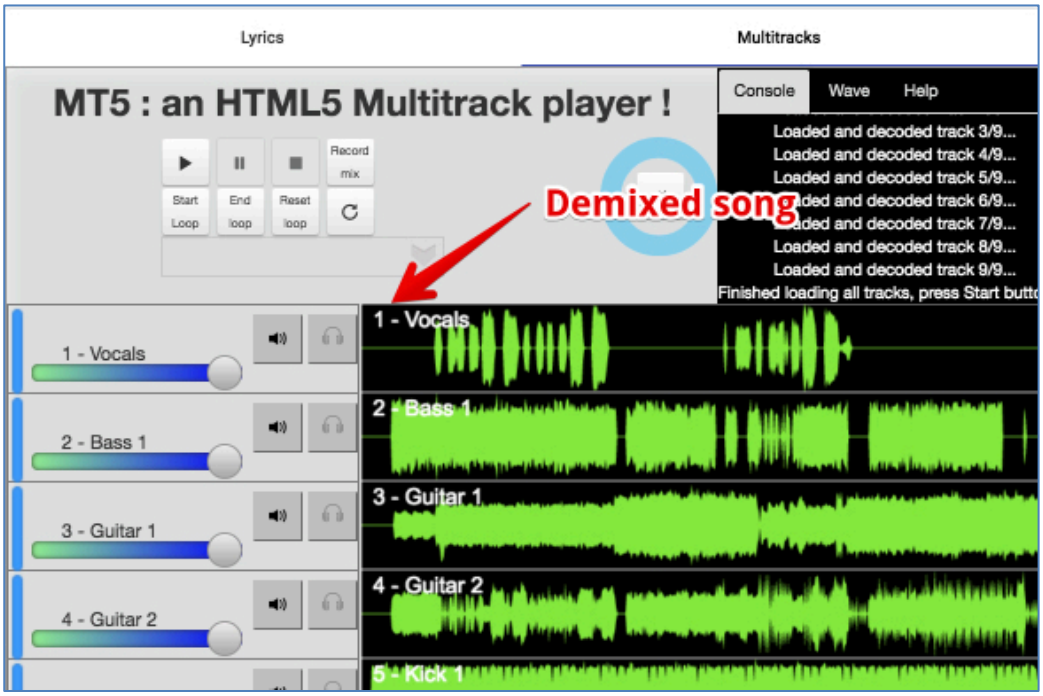


Figure 126. WASABI intègre des applications WebAudio, ici un lecteur audio multipiste.

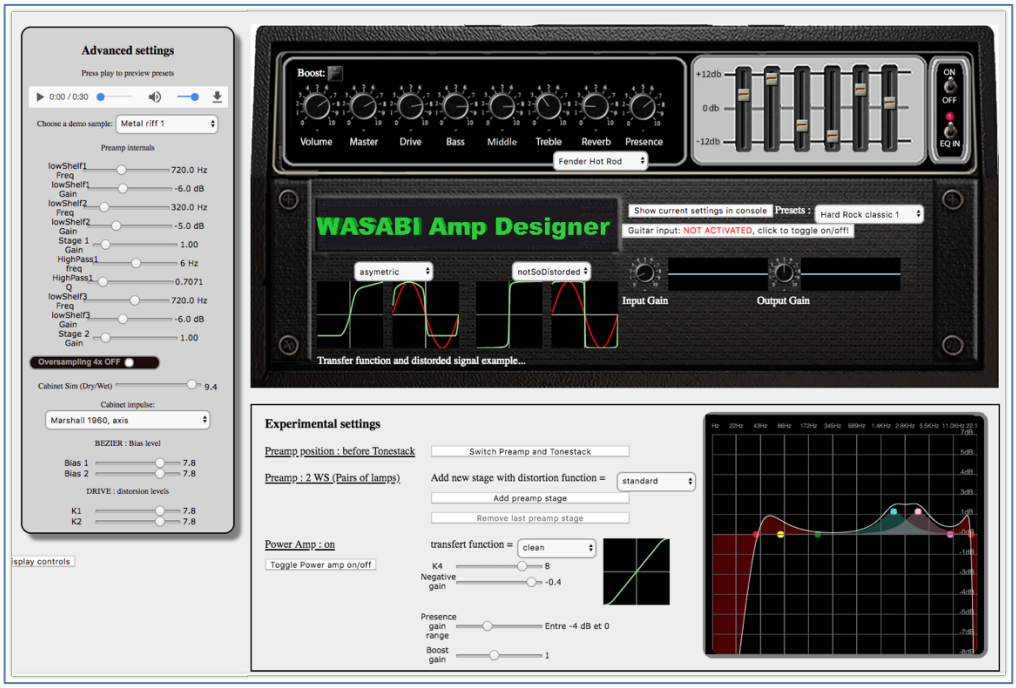


Figure 127. Interface graphique d'un simulateur d'amplificateur de guitare à lampes, conçu comme une application web et intégré aux outils WASABI.
Détails dans les chapitres 10 et 11 de ce manuscrit.

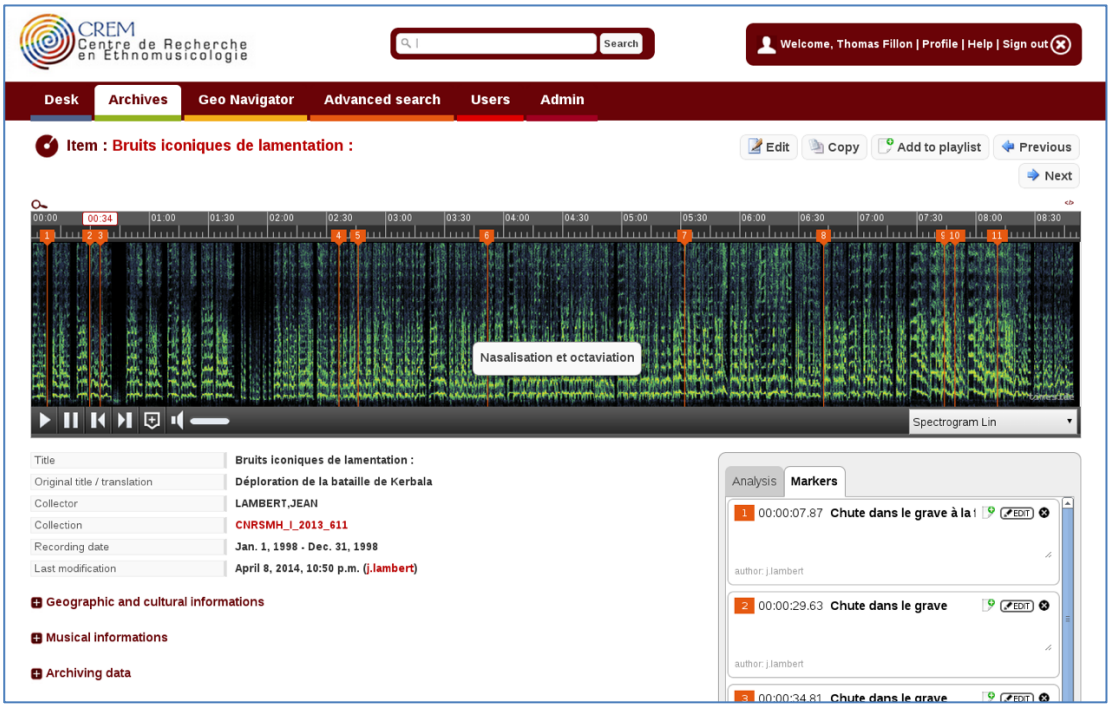


Figure 128. Plate-forme Telemeta déployée pour la gestion et l'annotation des archives sonores du projet CREM-CNRS.

Wimmics et Parisson participent également à la réalisation de démonstrateurs interactifs sous forme d'applications Web riches, capables de manipulations sonores en temps réel :

1. Applications Web pour parcourir les bases de connaissances musicales : un moteur d'analyse audio, un moteur de recherche, un navigateur de navigation à facettes avec la possibilité d'éditer / corriger manuellement les données textuelles (titre d'une chanson ou des paroles mal orthographiées, par exemple, corriger ou ajouter une annotation), mais aussi une API pour des programmes clients externes.
2. Démonstrateurs liés à des scénarios :
    - Des outils pour les écoles de musique,
    - Un outil pour aider à la composition musicale,
    - Un outil pour les data-journalistes,
    - Un outil pour l'analyse musicologique d'une œuvre,
    - Exemples interactifs intégrables dans les MOOCs.
3. Un ou plusieurs datasets contenant l'ensemble des données RDF décrivant le corpus final.
4. Des briques logicielles Open source permettant (1) de développer des applications exploitant cet ensemble de données : analyseurs audio, analyseurs, analyseurs, widgets WebAudio, etc. et (2) de faciliter l'interconnexion avec la base de données en ligne.

L'IRCAM et I3S sont impliqués dans la norme WebAudio API et dans le développement d'applications utilisant cette technologie (organisation de la première Conférence Web

WAC2015, publications sur le sujet, projet ANR WAVE, membres du groupe de travail W3C WebAudio qui fait la norme, etc.). Ces démonstrateurs tireront le meilleur parti de cette technologie, qui permet de développer des applications musicales de qualité proche de ce qui est connu dans le monde des applications natives, mais surtout en profitant des possibilités du Web : collaboration et hypermédia.

9.4 Premiers résultats

Actuellement, la base de données WASABI comprend 77K artistes, 200k albums, plus de deux millions de chansons. Nous avons rassemblé pour chaque artiste sa discographie complète, les membres du groupe avec leurs instruments, la chronologie, etc.

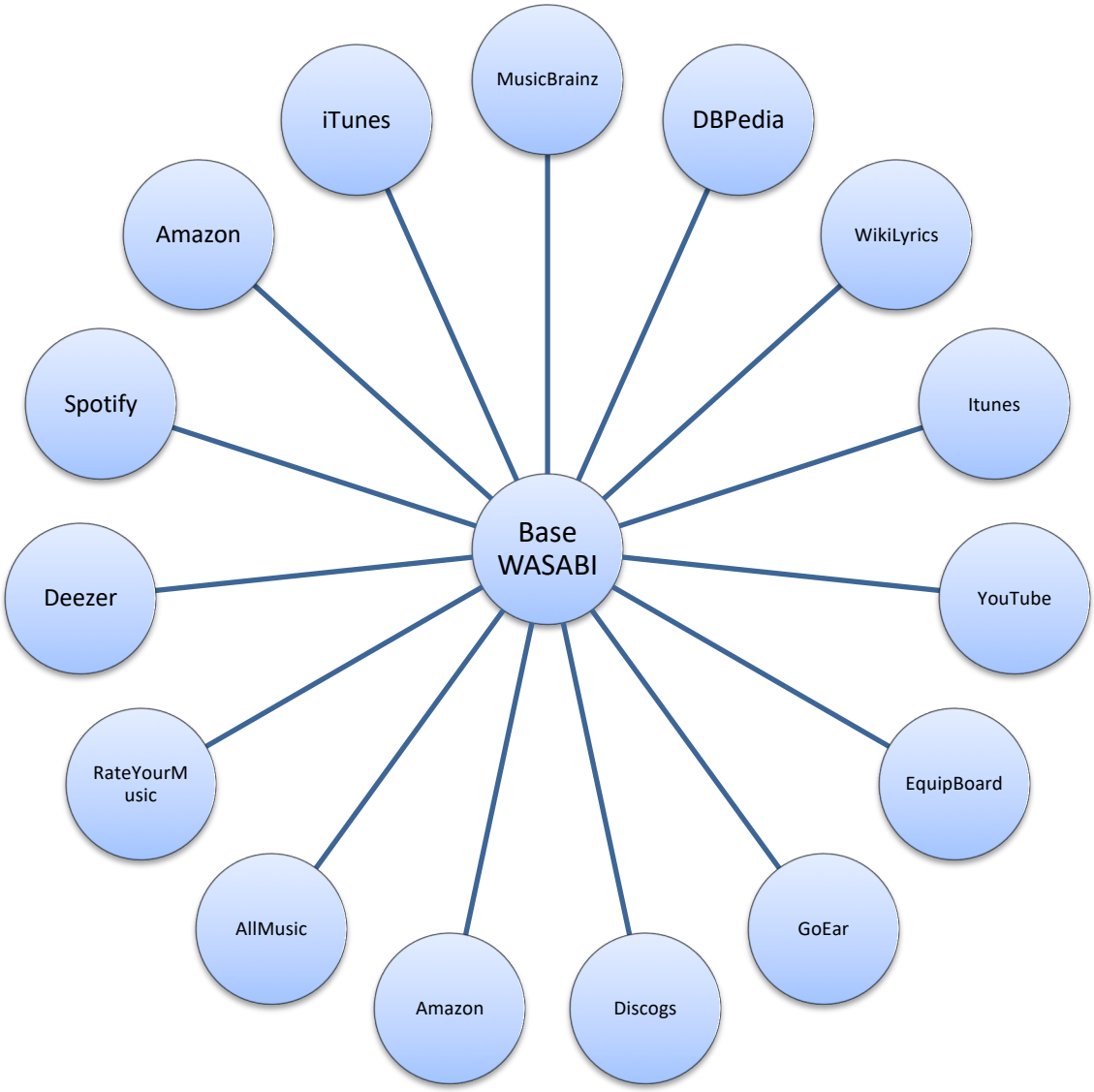


Figure 129. Les différentes sources de données utilisées pour la collecte de données « culturelles ».

Pour chaque chanson nous avons rassemblé ses paroles[117], les paroles synchronisées lorsqu'elles sont disponibles[118], le nombre de mots, les résumés DBpedia et les catégories auxquelles la chanson appartient, le genre, les tags, l'auteur, la date de sortie, les prix, les producteurs, l'artiste et/ou les membres du groupe, la piste audio stéréo de Deezer, lorsqu'elle est disponible, les pistes audio non mixées de la chanson, son ISRC, tempo (beats par minutes), la durée, etc.

Nous avons comparé les ids de chansons de la base de données WASABI avec les ids de MusicBrainz, iTunes, Discogs, Spotify, Amazon, AllMusic, GoHear, YouTube.

Nous sommes partis des entrés de WikiLyrics (deux millions de paroles de chansons, les paroles sont la « graine » de notre base de connaissance). Chaque chanson de WikiLyrics possède des liens vers d'autres sources de données, notamment MusicBrainz, Discogs, YouTube, Amazon, etc. Depuis les liens il est possible d'obtenir des les ids des chansons, artistes et albums sur ces différentes sources de données. Mais pas les ids Deezer, et nous avons absolument besoin de ces ids pour matcher les données audio. Certains ids sont également faux (vidéo YouTube qui ne correspond pas à la version de la chanson par ex.)

Deezer possède dans sa propre base de données des ids vers MusicBrainz mais aussi vers d'autres sources.

L'appariement se fait à l'aide de critères multiples afin d'éviter les fausses correspondances (titre de la chanson, de l'album, nom de l'artiste, ids MusicBrainz, YouTube, etc.)

Par ailleurs, nous avons eu aussi besoin de trouve les entrées DBPedia correspondantes. Nous avons utilisé ici plusieurs heuristiques : 1) interroger les instances de DBPedia de différents pays en fonction de la nationalité des artistes, 2) utiliser un service comme sameas.org qui permet de matcher des ids MusicBrainz et WikiPedia (et donc déduire des URIs DBPedia).

Nous n'avons pas récupéré toutes les métadonnées disponibles sur DBPedia car cette source a pour particularité de n'avoir qu'une seule page par chanson, et non pas une page pour toutes les versions de la chanson. Par exemple, « Stairway to Heaven » de Led Zeppelin, est une entrée unique dans DBPedia, mais il existe des dizaines de versions par de multiples artistes, avec des producteurs différents à chaque fois. Une étude au cas par cas sur des dizaines d'exemples de chansons connues a permis de choisir quelles métadonnées sont pertinentes sur telle ou telle source de données.

Enfin, munis des ids des chansons sur de multiples sources, nous avons transmis un dump de la base à Deezer. Ainsi nous avons réussi à faire correspondre 87 % des chansons WASABI avec les chansons de la base de données Deezer (c'était essentiel, car nous avions besoin des pistes audio correspondantes).

[117] Source : http://lyrics.wikia.com/
[118] Source : http://usdb.animux.de/

Nous avons utilisé de multiples critères : métadonnées collectées auprès de DBpedia et MusicBrainz, mais aussi, lorsque nous avions les identifiants de cette chanson à partir d'autres sources de données (iTunes, Spotify, etc.), nous avons utilisé ces informations car Deezer avait également collecté ces identifiants.

La base WASABI contient 1732169 chansons avec des paroles, 73444 qui ont au moins un résumé sur DBpedia, 7227 qui ont été identifiées comme des "chansons classiques" (elles ont été numéro un, ou ont reçu un Grammy award, ou ont beaucoup de reprises, etc. Nous avons fait correspondre 2k chansons avec leur version multipiste. Ces versions multipistes seront utilisées comme référence pour les algorithmes de séparation des sources que l'IRCAM exécutera sur l'ensemble des chansons.

Au final, il reste encore des erreurs qui ont été corrigées à l'aide d'outils de détection de conflits ou de perte de cohérence : même métadonnée mais valeurs différentes selon la source, métadonnée censée être unique mais valeurs multiples, détection de variantes orthographiques dans les noms. Nous avons également organisé des « marathons WASABI » pendant lesquels plusieurs personnes naviguaient manuellement dans la base à la recherche d'erreurs. Des erreurs « systématiques » ont ainsi pu être corrigées et les données recollectées.

La base de données est en passe de recevoir les résultats des premières analyses audio menées chez Deezer à l'aide des outils développés par l'IRCAM (Octobre 2018).

Des résultats très encourageants sur la détermination de la structure des chansons par analyse conjointe des paroles et de l'audio, sur un sous ensemble de la base WASABI ont été publiés et vont être également intégrés et poursuivi sur l'ensemble du dataset dans les mois qui viennent [185].

9.5 Architecture logicielle

WASABI s'appuie sur de multiples moteurs de bases de données : actuellement, nous fonctionnons sur un serveur MongoDB avec une indexation par ElasticSearch. Cette base de données est accompagnée d'une API REST assez complète[119].

La base NoSQL, couplée à l'outil d'indexation flou ElasticSearch fournit une API RESTful permettant des requêtes simples mais très rapides, comme « chercher une chanson ou un artiste » avec un système d'auto-complétion très réactif, ou bien « connaître tous les disques d'un artiste », « tous les détails d'une chanson, d'un disque, d'un artiste », « les catégories d'une chanson », etc.

Les données sont régulièrement exportées vers un triple store RDF de type Virtuoso, à l'aide des outils de mapping NoSQL/RDF développés par Franck Michel dans sa thèse [221]

Dans un premier temps nous avons aligné les métadonnées culturelles « classiques » avec l'ontologie MusicBrainz, elle-même réutilisant les vocabulaires de The Music Ontology

[119] https://wasabi.i3s.unice.fr/apidoc/

[175], et nous avons effectué la description des pistes dans le cas de chansons ayant une version démixée avec la Multitrack Ontology [220]. D'autres ontologies sont en cours d'évaluation (Studio Ontology, AudioEffect Ontology, etc.) pour la définition d'autres métadonnées dont nous disposons.

Nous sommes également en train de mener des entrevues avec les utilisateurs finaux de WASABI et les ontologies qui correspondront aux différents besoins et cas d'utilisation qui ne sont pas couverts par des ontologies existantes (notamment pour tout ce qui est relatif à l'analyse des paroles en langage naturel). Le projet est à mi-parcours et il est normal que de nombreuses tâches soient encore inachevées.

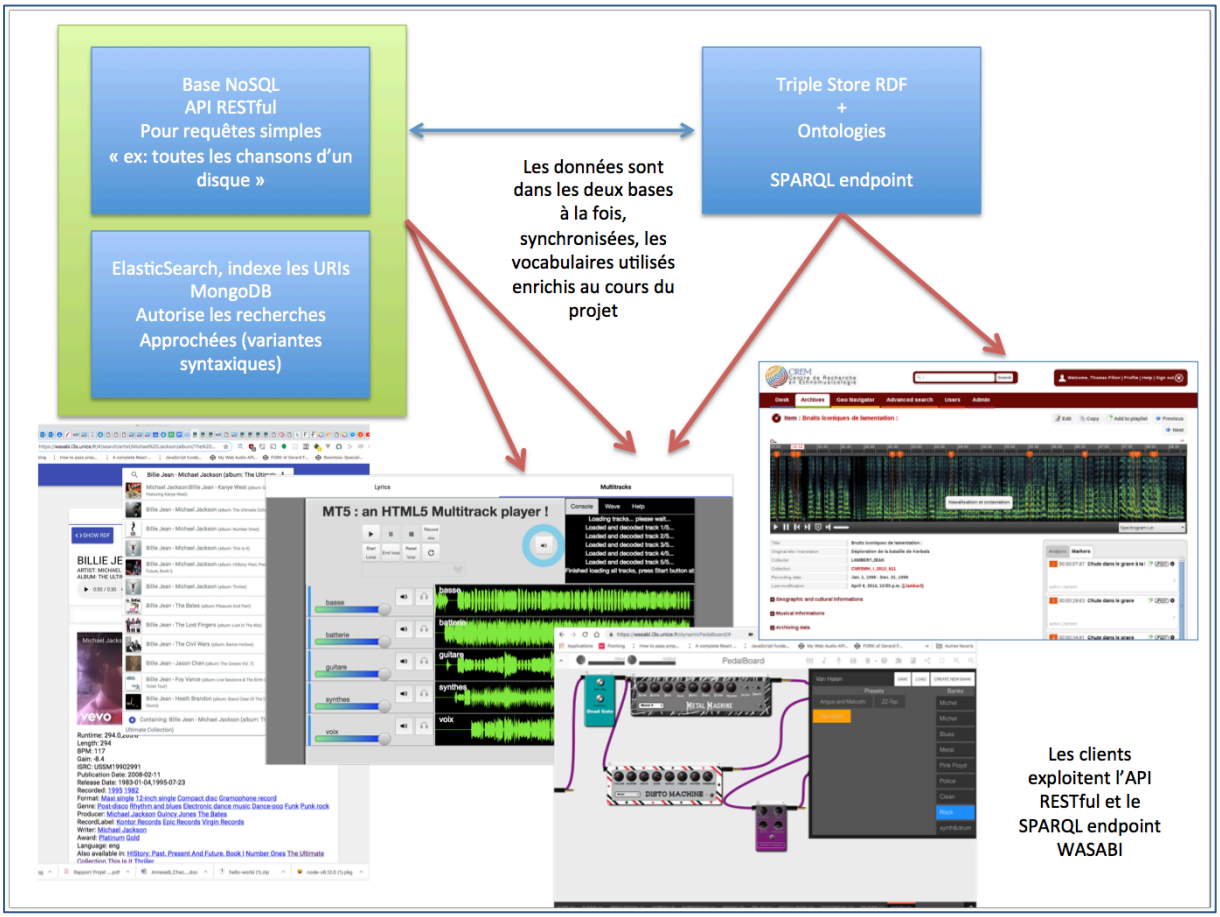


Figure 130. Architecture des bases de données WASABI. Une base NoSQL (MongoDB) est utilisée avec un outil d'indexation acceptant des orthographes rapprochées (ElasticSearch) pour les requêtes « rapides ». Un triple store Virtuoso est synchronisé et héberge les mêmes données en RDF. Les vocabulaires utilisés sont enrichis pendant le projet.

9.6 Conclusion

Le projet WASABI, lancé en janvier 2017, propose une base de métadonnées de 2 millions de chansons qui contient les métadonnées issues de multiples sources du Web de données (WikiLyrics, DBpedia, MusicBrainz, Deezer, etc.), les paroles et les fichiers audio correspondants.

Parmi les défis que nous avons relevés :

- Collecte et fusion de données multi-sources : désambiguïsation, curation, fusion, vérification de cohérence, mise en correspondance métadonnées / audio.
- Analyse des besoins des utilisateurs finaux (diffuseurs de musique, compositeurs, écoles de musique ou d'ingénieurs du son, musicologues, journalistes).
- Identification de vocabulaires / ontologies pour la description des métadonnées.
- Implémentation avec base de données hybride NoSQL/RDF et définition d'une API RESTful, d'un SPARQL endpoint.
- Développement d'applications clientes (dans mon équipe, mais aussi à l'IRCAM).

Tout au long du projet, nous ajoutons d'autres sources de données, nous exécutons des algorithmes MIR sur les fichiers audio et des algorithmes de traitement du langage naturel. Des algorithmes agissant conjointement sur les trois types de données (culturelles, paroles et audio) ont déjà permis d'améliorer l'état de l'art (par exemple pour le calcul de la structure d'une chanson).

L'accès à la base de données est public[120] par le biais de l'interface Web WASABI ou par le biais d'une API RESTful et d'un SPARQL endpoint.

La base de données est exploitée par des applications client WebAudio telles qu'un lecteur audio multipistes, l'application Telemeta pour l'indexation audio automatique, semi-automatique et manuelle, ou encore et des outils de MAO pour le Web exploitant le standard WebAudio. Par exemple : simulations d'amplificateurs de guitare, multiples effets audio en temps réel, table de mixage, analyseurs de fréquences, oscilloscopes, audiogrammes et autres.

Les chapitres suivants de ce manuscrit présentent de telles contributions relevant de la MAO pour le Web et de la création de nouveaux standards.

[120] Nous voulons dire ici un accès complet aux métadonnées. Il n'y a pas d'accès public aux données protégées par le droit d'auteur, comme les paroles et les fichiers audio complets. Des extraits audio de 30 secondes sont néanmoins disponibles pour presque toutes les chansons.

10. Participation au standard WebAudio, simulation d'amplificateur à lampes pour guitare et autres effets audio (2015-)

Contexte

Le standard du W3C WebAudio apporte la "Musique Assistée par Ordinateur" (MAO) dans le navigateur. Notre objectif initial était d'évaluer les limites du standard WebAudio en termes de performances et d'API et de voir s'il était possible de concevoir des effets et instruments « temps réel » capables de rivaliser avec l'état de l'art existant dans le domaine des applications "natives" développées depuis les années 90. Ce travail s'intègre naturellement au projet ANR WASABI qui cible entre autres : les compositeurs, les écoles de musique, les écoles d'ingénieurs du son et les musicologues. Les applications WebAudio vont être intégrées à la base de données WASABI.

On pourra par exemple, choisir une chanson, prendre sa guitare et un clavier et jouer par-dessus la version multipiste, apprendre à régler un amplificateur de guitare ou un synthétiseur virtuel dans son navigateur, etc.

Par ailleurs, je participe, depuis 2015, avec le W3C, au développement du standard WebAudio et nous verrons que j'ai contribué -avec d'autres chercheurs- à l'ajout d'options pour des traitement à faible latence et au développement d'applications (preuves de concept) pour valider ces possibilités.

Contributions

J'ai d'abord développé seul depuis 2015, puis à partir de 2017 avec Jérôme Lebrun, la simulation fidèle d'un d'amplificateur de guitare à lampes existant : le Marshall JCM 800 (Figure 131), utilisé par de nombreux artistes célèbres (AC/DC, Guns and Roses, Deep Purple, etc.).

Ce travail consiste à émuler les différentes parties du circuit électronique de cet amplificateur à l'aide de WebAudio, en implémentant des algorithmes de traitement du signal connus à l'aide de l'API disponible, et en trouvant des solutions adéquates pour contourner certaines contraintes propres à l'environnement dans lequel nous nous trouvons (le navigateur Web). Enfin, nous avons comparé (quantitativement et qualitativement) notre réalisation avec l'état de l'art, c'est-à-dire des simulations natives, pour la plupart commerciales, écrites en C++, et n'ayant pas les contraintes propres aux webapps.

Ce travail nous a valu le "*best paper presentation award*" à la conférence WebAudio Conference 2017 et un prix de l'Université Côte d'Azur en 2017.

Depuis, nous avons poursuivi ce travail en affinant les modèles utilisés dans la simulation, puis en créant un framework permettant de reproduire différentes architectures électroniques présentes dans les amplificateurs à lampes que l'on trouve dans l'équipement de nombreux musiciens. On peut maintenant simuler par exemple un amplificateur Fender, Vox, Mesa Boogie, etc. ou même créer des designs originaux. Ces outils ont été testés par des guitaristes professionnels, et sont utilisés par des écoles de musique à titre expérimental[121].

Figure 131. Un des premiers prototypes de la simulation temps réel sous forme d'application web, de l'amplificateur de guitare à lampes Marshall JCM800.

En parallèle, j'ai développé, avec deux ingénieurs que je supervise dans l'équipe WIMMICS sur le projet ANR WASABI, un écosystème logiciel pour intégrer d'autres instruments et effets audio temps réel, en plus des simulations d'amplificateurs.

Ce travail a obtenu une "*honorable mention*" lors des démos de la conférence TheWeb Conf 2018 qui s'est tenue à Lyon, classé rang A.

Le projet WASABI n'est pas encore à mi-parcours, mais certains logiciels sont dans un état très avancé et ont été présentés le 31/5/2018 lors des journées "*Digital Technologies*" organisées par l'Université Côte d'Azur pour présenter aux entreprises les travaux de chercheurs pouvant faire l'objet de transferts industriels.

La société imusic-school.com présente à Sophia-Antipolis collabore avec nous depuis quelques mois et devrait intégrer une partie de ces logiciels dans son école de musique en ligne.

[121] Voir par exemple ces vidéos : https://www.youtube.com/watch?v=DUM99xQt5fg (Roger Larthaud, professeur à l'école de musique de Biot), https://www.youtube.com/watch?v=dKJnUsyWkz0&feature=youtu.be (Luc Botta, professeur au conservatoire de Gap), ou encore https://www.youtube.com/watch?v=lIfm9ZMtG-I

Collaboration, encadrements

J'encadre deux ingénieurs et coordonne le travail des chercheurs impliqués dans le projet WASABI chez WIMMICs autour du développement des applications WebAudio. J'ai également encadré trois stages et de nombreux projets de Master 2 relatifs au travail présenté ici.

Je participe depuis 2015 à la création du standard WebAudio du W3C (je fais partie du WebAudio Working Group) et à l'organisation de la WebAudio Conference, seule conférence spécialisée dans le domaine.

Je donne régulièrement des tutoriaux sur WebAudio dans des conférences internationales (TheWeb Conference 2018, WWW 2016, WAC 2018 par ex.).

J'ai écrit un MOOC sur la plateforme edX avec un chapitre dédié à WebAudio, ayant réuni plus de 50.000 étudiants[122].

Publications principales ayant résulté de ces travaux

Remarque : la WebAudio conférence (WAC) n'a pas encore de "rank", car le domaine de recherche a émergé en 2015, mais demeure la première conférence réunissant tous les acteurs du domaine. Voir les lettres de Samuel Goldsmith (IRCAM) et de Mark Sandler (Queen Mary University of London), présidents de deux éditions de cette conférence.

Buffa, M., **Lebrun, J.** *WebAudio Virtual Tube Guitar Amps and Pedal Board Design.* WebAudio Conference 2018, Berlin. (2018). 〈hal-01893781〉

Buffa, M. and Lebrun, J. *"Guitarists* will *be happy: guitar tube amp simulators and FX pedal in a virtual pedal board, and more!"*, WebAudio Conference 2018. Demo Paper. Berlin. (2018). 〈hal-01893681〉

Buffa, M., Lebrun, J. *Real-Time Emulation of a Marshall JCM 800 Guitar Tube Amplifier, Audio FX Pedals, in a Virtual Pedal Board*. WWW2018 - TheWebConf 2018 : The Web Conference, 27th International World Wide Web Conference, Apr 2018, Lyon, France. 2018, 〈hal-01721463〉

Buffa, M., Lebrun, J. *"Real time tube guitar amplifier simulation using WebAudio."* Web Audio Conference 2017 – Collaborative Audio #WAC2017, Aug 2017, London, United Kingdom. 2017. 〈hal-01589229〉 **Best paper presentation award**. (no best paper at this conference, this is equivalent).

Buffa, M., Lebrun, J. "*Web Audio Guitar Tube Amplifier vs Native Simulations"*. Web Audio Conference 2017 – Collaborative Audio #WAC2017, Aug 2017, London, United Kingdom. 2017, 〈hal-01589330〉

Michel Buffa, Maxime Demetrio, Nouriel Azria. "*Guitar pedal board using WebAudio"*. Web Audio Conference, Apr 2016, Atlanta, United States. 2016, 〈hal-01400910〉

Résumé de cette section

Les simulations numériques d'amplificateurs de guitare sont devenues populaires avec des appareils tels que la série POD de Line6 au début des années 2000, ou plus récemment

Ce ch[122] "HTML5 Apps and Games", Week 1, sur edX.org

avec le modélisateur d'amplis Axe FX-II de Fractal Audio Systems : un processeur numérique de préampli/effets tout-en-un qui contient un vaste inventaire virtuel de centaines d'amplis de guitare vintages et modernes. Il s’agit ici de matériel hardware embarquant des circuits DSP (Digital Sound Processor) programmables, exécutant le code de la simulation numérique.

En 2002, sort Amplitube, un plug-in audio commercialisé par IK multimédia, c’était le premier simulateur d'amplis *100% logiciel* sur le marché, suivi bientôt par Guitar Rig de Native Instruments. Aujourd'hui, nous pouvons trouver plusieurs plug-ins logiciels natifs (commerciaux ou gratuits, quelques-uns seulement sont open source) pour les stations de travail audionumériques, qui simulent des amplificateurs de guitare existants, ou qui sont basés sur un design original de leurs auteurs.

Ces simulations d'amplificateurs logiciels sont pratiques dans les situations nomades, ou lorsque le budget de production pour l'enregistrement est faible, car elles sont moins chères et plus flexibles que leurs équivalents matériels.

Depuis 2015 j’ai d’abord travaillé seul, puis avec Jérôme Lebrun, sur la création de simulations temps réel d’amplificateurs de guitare à lampes, tournant dans un navigateur web et exploitant l’API WebAudio. Nous présenterons dans ce chapitre les résultats obtenus et montreront qu’il est possible de rivaliser avec l’état de l’art des simulations dans le monde natif. En effet, à ce jour*, les travaux que je présente ici sont encore uniques en leur genre* [37, 41].

Aujourd’hui nous disposons d’un outil interactif permettant de créer et tester interactivement des simulations temps réel de plusieurs modèles d’amplificateurs du marché ou originaux. On peut ainsi régler finement la conception des différentes parties d’un amplificateur (préamplificateur, amplificateur de puissance, égalisation basse, médium et aigus, simulation des haut-parleurs etc.), en modifiant la topologie et la nature des éléments qui les composent (lampes, filtres, gains etc.). L’interface utilisateur de cet outil développé dans le cadre du projet ANR WASABI est présentée Figure 132.

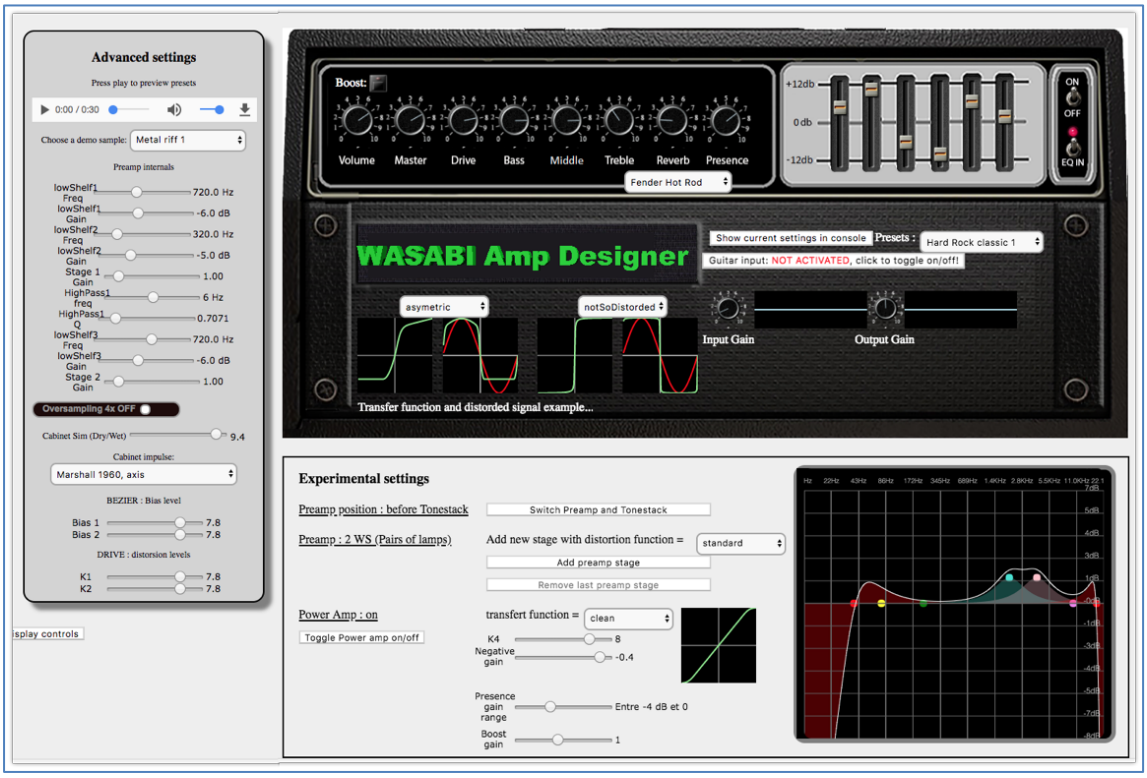


Figure 132. Interface utilisateur du « WASABI Amp Designer », un outil permettant de créer des simulations de plusieurs modèles d'amplificateurs.

Cet outil produit un fichier de configuration qui sert alors à générer une version simplifiée, plus ergonomique, que les guitaristes non experts peuvent utiliser. Voir Figure 133.

Figure 133. Trois amplificateurs ayant une conception différentes, générés par l'outil de la figure précédente. Le premier produit des sons approchant le son d'une guitare acoustique, le second des sons « classic rock » et le troisième des sons « métal ».

Nous présenterons les principes de fonctionnement des amplificateurs à lampes et un état de l'art sur les différentes manières de les simuler, en insistant sur les particularités de la plateforme Web, sur ses contraintes et sur la manières dont nous avons pu les contourner. La simulation de chaque étage composant un amplificateur sera détaillée et nous concluront par des tests utilisateurs effectués par plusieurs guitaristes amateurs et professionnels.

Nous sommes aujourd'hui les seuls à proposer des simulations de ce type fonctionnant en temps réel sur la plateforme Web, et la qualité des simulations est comparable à ce qui se fait dans le monde natif, en témoignent les tests en aveugles que nous avons effectués avec des guitaristes amateurs et professionnels, et les mesures quantitatives obtenues.

Dans le chapitre 11 nous verrons qu'il est même possible de faire fonctionner *telles quelles* ces simulations dans des station de travail audionumériques natives du marché (Ableton Live, Reaper, Cubase, etc.) et ainsi les comparer à leurs concurrentes écrites en langage natif (C/C++), dans un même environnement d'exécution.

10.1 Introduction

Les guitaristes jouant de la guitare électrique sur scène utilisent dans leur grande majorité un ou plusieurs « amplificateur(s) ». Le rôle de ces amplificateurs est de transformer le signal électrique très faible sortant de l'instrument (généré par le mouvement des cordes en face d'aimants situés sur les micros de la guitare), de l'ordre de quelques millivolts, en centaines de volts capables de mettre en mouvement des haut-parleurs de parfois 20 ou 30 centimètres de diamètre. Pendant ce processus d'amplification l'objectif aujourd'hui n'est pas de reproduire le signal initial avec un maximum de fidélité (comme en hi-fi) mais au contraire de « sculpter » le son pour qu'il corresponde à différents timbres et dynamiques liés à certains styles de musique. Beaucoup de distorsion « froide » pour le hard rock ou le métal, un léger « crunch » (une distorsion légère) pour le blues, des sons « chauds et clairs » pour le Jazz, etc. Les guitaristes modernes choisissent leur amplificateur en fonction du son et de la « dynamique » qu'il procure (les « sensations de jeu »).

Ce chapitre parle principalement de l'amplificateur Marshall JCM800 (Figure 134) rendu célèbre par les groupes de « classic rock » comme Led Zeppelin, Deep Purple, AC/DC, Guns and Roses etc. C'est l'amplificateur emblématique de ce style de musique.

Dans le cadre du projet ANR WASABI nous avons réalisé la première simulation d'un amplificateur de guitare à lampes fonctionnant dans un navigateur web, et exploitant le standard émergeant WebAudio. Dans ce projet, nous développons un standard de composants réutilisables (pour des « plugins audio » semblables à ceux qui existent dans le marché des applications natives, voir chapitre suivant), un ensemble d'instruments virtuels (pianos, synthétiseurs, boîtes à rythmes etc.) et d'effets audio temps réel (simulation d'amplificateurs de guitare, réverbération, delay, chorus, flanger etc.) ainsi que des logiciels capables d'assembler ces éléments ou de jouer des morceaux multipistes en accompagnement. Ces éléments de « MAO pour le web » vont être associés aux chansons de la base de données WASABI et sont destinés aux écoles de musique, aux écoles d'ingénieurs du son, aux compositeurs.

Nous verrons que dans la simulation du JCM800 que nous avons développée, nous avons également ajouté des options pour pouvoir simuler d'autres types d'amplificateurs. Un des avantages des simulations numériques est que lorsqu'on désire modifier la conception

de l'amplificateur simulé, il est plus facile de rajouter/modifier des éléments virtuels que d'assembler de véritables composants électroniques !

La plus grosse difficulté étant bien sûr d'obtenir « un bon son » et une « bonne dynamique », c'est à dire que guitare en main, les sensations lorsqu'on joue doivent être le plus proche de ce que l'on ressent quand on utilise un véritable amplificateur. Par exemple, avec un amplificateur à lampes, si on joue doucement (si on gratte à peine les cordes), on aura un son clair, mais si on attaque les cordes, alors on aura un son plus distordu. Ce comportement est dû aux caractéristiques non linéaires et à des effets de contre-réaction de certains composants électroniques lorsqu'ils sont surchargés (par un signal de trop grande amplitude). Cette non linéarité rend la simulation de ces composants et de leurs interactions particulièrement complexe.

Figure 134. Véritable amplificateur de guitare Marshall JCM800, 1m50 de hauteur, poids : 45 kilos.

Les simulations numériques d'amplificateurs de guitare sont devenues populaires dans un premier temps avec des appareils hardware embarquant des circuits DSP programmables, comme le « POD » de Line6 au début des années 2000 (Figure 135), ou plus récemment avec le modélisateur d'amplis Axe FX-II de Fractal Audio Systems (Figure 136), un processeur numérique tout-en-un qui contient un vaste inventaire virtuel de centaines d'amplificateurs de guitare vintage et moderne et d'effets audio.

Figure 135. Multi effets / simulateur d'amplificateur POD XT produit par la société Line6. Trente centimètres de large. Poids : 1 kilo.

Figure 136. Axe FX-II de fractal Design

En 2002, Amplitube, un plugin audio commercialisé par IK Multimedia (10.3), a été la première simulation 100% logicielle d'amplificateurs de guitare à lampes mise sur le marché avec succès, suivie bientôt par Guitar Rig de Native Instruments (Figure 138).

Ce sont aujourd'hui deux logiciels très réputés dans le monde de la MAO, utilisés en studio pour l'enregistrement de nombreux disques commerciaux.

Figure 137. Le plugin logiciel Amplitube 1, sorti en 2002 (IK Multimedia).

Figure 138. Guitar Rig 1 par Native Instruments. Le pédalier MIDI associé se branche sur un port USB de l'ordinateur et permet de commander le logiciel.

En 2018, nous pouvons trouver des centaines de plugins natifs (commerciaux ou gratuits, quelques-uns seulement sont open source), qui simulent des amplificateurs de guitare existants, ou qui sont basés sur un design original de leurs auteurs. Ces plugins fonctionnent dans des stations de travail audionumériques (*DAWs* ou *Digital Audio Workstations* en anglais) qui dominent le marché : Logic Audio (commercialisé par Apple), Ableton Live (Figure 139), Cubase (Steinberg), ProTools (Avid), GarageBand (Apple) etc.

Figure 139. La station de travail audionumérique Ableton live avec des plugins (sur la gauche). A droite : les pistes d'enregistrement sur lesquelles on va appliquer les traitements audio proposés par ces plugins.

Les simulations logicielles d'amplificateurs sont populaires dans les situations nomades, ou lorsque le budget de production pour l'enregistrement est faible, car elles reviennent moins cher et sont plus flexibles que leurs équivalents matériels analogiques ou numériques.

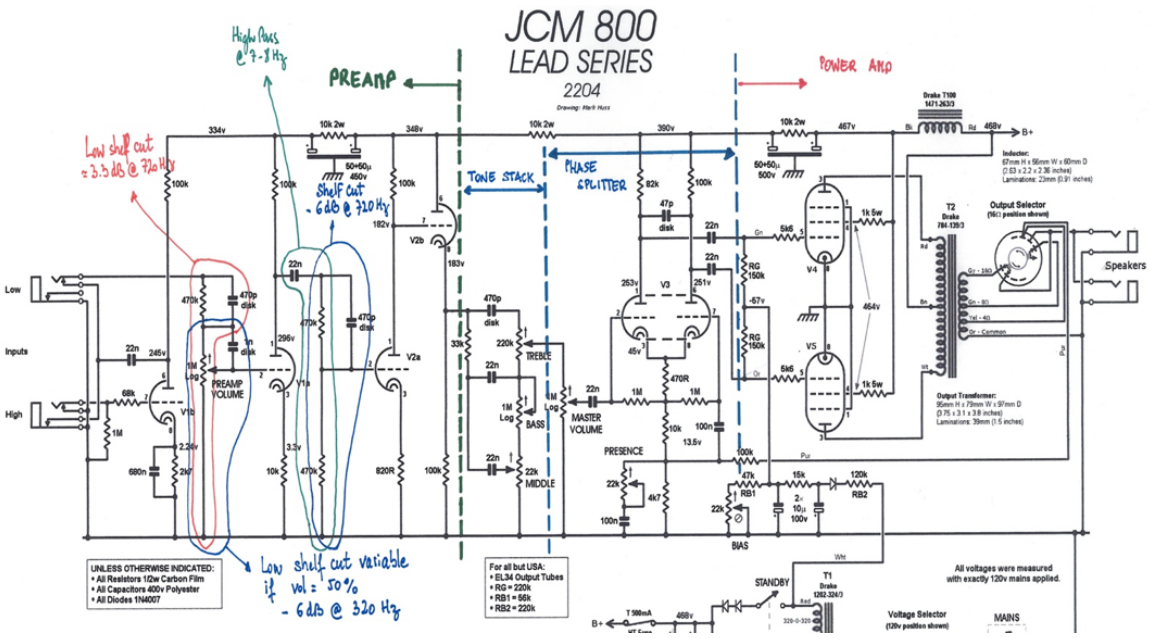


Figure 140. Schéma électronique de l'amplificateur de guitare à lampes Marshall JCM 800. On a découpé verticalement les différents « étages ». De droite à gauche : préamplificateur, tonestack, amplificateur de puissance. Suivent le circuit de réverbération du son et les haut-parleurs (qu'il faut également simuler).

10.2 État de l'art de la MAO sur le Web

En 2012, Google Chrome a proposé pour la première fois l'accès à une entrée audio de l'ordinateur (microphone ou carte son externe). L'implémentation pour Mac OSX autorise le traitement du son en temps réel avec une très faible latence (ce qui n'est pas encore le cas sous Windows). Aujourd'hui les autres navigateurs Web du marché ont également implémenté cette fonctionnalité qui s'appuie désormais sur l'API Media Capture and Streams[123] du W3C.

La démo "Input Effects" de Chris Wilson[124] a été l'une des premières à montrer des effets de traitement du son en temps réel et à proposer des implémentations d'effets célèbres tels que Delay, Distortion, WahWah, Chorus, Phaser, etc. Cette démonstration ne permettait pas de chainer des effets mais prouvait qu'il était possible d'obtenir un traitement à faible latence à partir de code JavaScript tournant dans un navigateur. Cependant, se rapprocher du son et de la dynamique d'un véritable amplificateur de guitare est un vrai défi que les exemples de Chris Wilson n'ont pas abordé.

De nombreux articles ont été écrits sur la modélisation des amplificateurs de guitare à lampes [42][43], et sur les particularités des effets de distorsion linéaires et non linéaires adaptés à la guitare [44][45][46][47]. Certains ouvrages, comme celui écrit par James J. Clark « *Advanced programming techniques for modular synthesizers* » [48], ne se concentrent pas uniquement sur la guitare mais couvrent en profondeur les différentes approches pour obtenir un effet de distorsion sur un signal.

123 https://www.w3.org/TR/mediacapture-streams/
124 https://webaudiodemos.appspot.com/input/index.html

L'approche commune consiste à modéliser les différentes parties d'un amplificateur de guitare. Wikipedia donne une assez bonne description du design de haut niveau d'un tel matériel : « *En général, les amplificateurs de guitare ont deux étages de circuit d'amplification et ont fréquemment des circuits de mise en forme du son, qui comprennent habituellement au moins des réglages de basses et d'aigus (le tonestack), et qui fonctionnent de la même manière que les commandes équivalentes sur une chaîne hi-fi domestique. Certains amplificateurs de guitare ont un égaliseur graphique, commandé par des sliders verticaux, pour contrôler plusieurs bandes de fréquences. Le premier étage d'amplification est le préamplificateur (il peut y en avoir plus d'un), qui amplifie le signal de guitare à un niveau que peut piloter l'étage de puissance (on passe de millivolts –sortie des micros de la guitare- à une dizaine de volts). L'amplificateur de puissance ou étage de sortie produit un signal de courant plus élevé pour piloter un haut-parleur afin de produire un son que le guitariste et le public peuvent entendre* ».

Il existe principalement deux approches pour simuler les différentes parties d'un amplificateur de guitare : l'une s'appelle la *modélisation virtuelle analogique* (en anglais : *virtual analog* ou *physical modeling*) (Figure 141) et consiste à reproduire le schéma électronique dans un logiciel de simulation de circuits analogiques, par exemple le logiciel SPICE[125], standard de l'industrie électronique, puis demander au logiciel de traduire ce circuit en équations à résoudre. Ces équations sont typiquement des équations algébriques différentielles non linéaires.

SPICE peut également produire du code C++ prêt à être exécuté. Cependant, il est souvent nécessaire de faire des simplifications et des optimisations pour obtenir une solution permettant un traitement en temps réel. C'est particulièrement le cas de la modélisation des lampes utilisés dans les amplificateurs de guitare et de leurs interactions avec d'autres parties du circuit (voir [42] et [49] pour une synthèse des techniques courantes d'optimisation).

[125] http://www.ecircuitcenter.com/AboutSPICE.htm

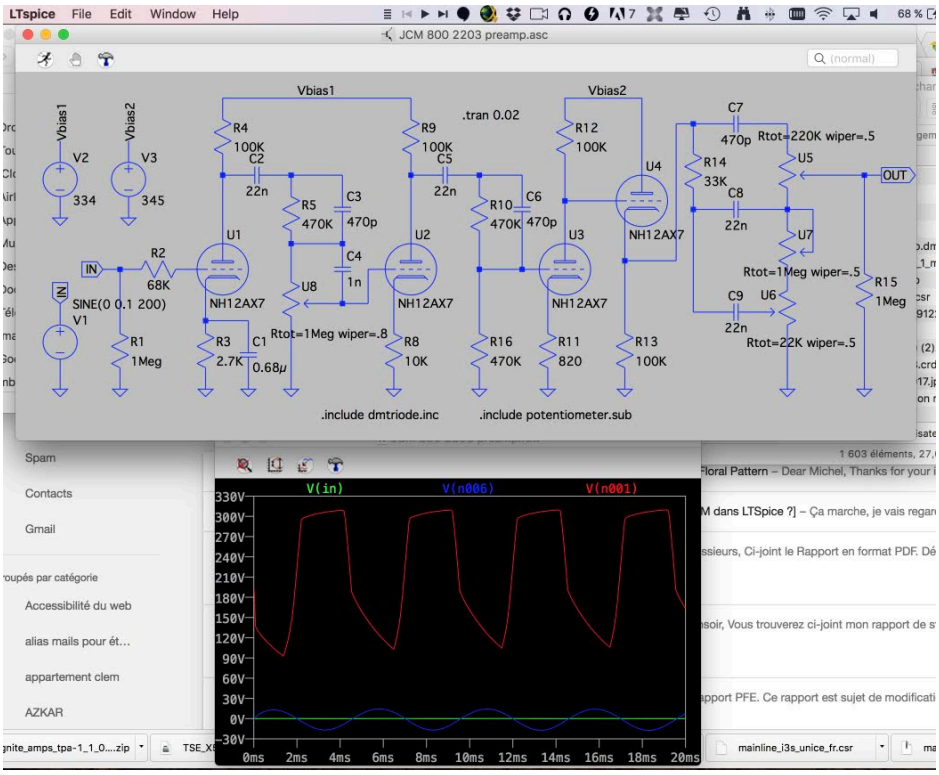


Figure 141. Le schéma électronique du préamplificateur du JCM800 dans le logiciel LTSPice (version gratuite de SPICE).

Une autre technique consiste à effectuer une émulation de plus haut niveau, dans laquelle les parties « logiques » sont identifiées (filtres, lampes, etc.) et émulées manuellement à l'aide de modèles séparés (Figure 142). Ceci est en théorie moins précis car certains effets et interactions tels que l'effet de contre-réaction de lampes surchargées ou l'action de l'impédance des haut-parleurs sur l'amplificateur de puissance peuvent ne pas être pris en compte.

Cependant, cette approche est aussi plus simple et plus adaptée à WebAudio et à ses limites actuelles (le traitement du signal par défaut est limité à des blocs de 128 échantillons à la fois, et il n'était pas possible jusqu'à très récemment de faire des traitements à l'échantillon près sans introduire des *glitches*[126] et de la latence).

WebAudio est une API standard du W3C qui propose de construire un « graphe audio » en connectant des nœuds entre eux. Le signal parcourt le graphe et chaque nœud traversé peut modifier le signal (ex : un nœud filtre pourra couper les sons aigus). Certains nœuds particuliers peuvent servir de source sonore (fichier audio, générateur d'onde, lien vers un élément HTML5 `<audio>` ou `<video>`).

[126] Terme employé pour décrire un craquement ou une coupure dans le son.

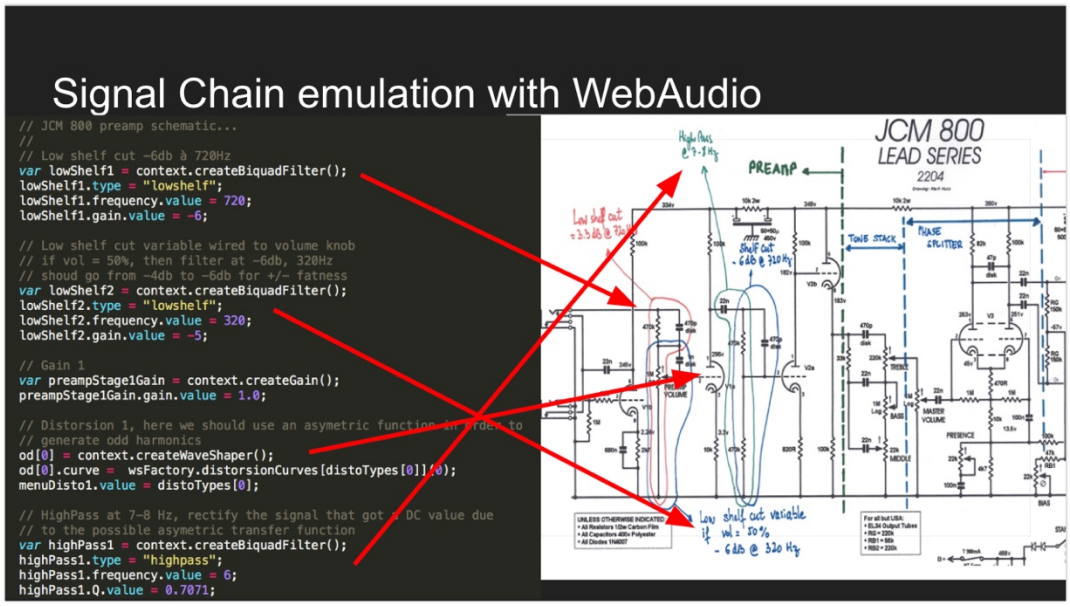


Figure 142. Simulation de haut niveau. On identifie les lampes, les filtres, et on simule partie par partie à l'aide de code utilisant l'API WebAudio.

WebAudio propose un ensemble de nœuds « de haut niveau » (comme le nœud WaveShaper[127] et le nœud de BiquadFilter) qui peuvent être utilisés pour modéliser des des lampes et des filtres, et il a été démontré que des techniques de *wave shaping* correctement utilisées, associées à un sur-échantillonnage et à un filtrage approprié, peuvent donner de bons résultats[128]. Le célèbre processeur d'effets POD XT de Line6 utilise de telles techniques [42].

A notre connaissance, il n'existe pas de travaux de recherche antérieurs qui ont essayé de simuler un amplificateur de guitare *complet* en utilisant WebAudio.

Pedals.io[129] est une recréation JavaScript de certaines pédales d'effets audio classiques pour guitaristes (delay, chorus, overdrive, etc.), et nous retrouvons presque les mêmes implémentations de ces effets dans de nombreuses bibliothèques JavaScript basées sur WebAudio telles que toneJS[130], tunaJS[131], pizzicatoJS[132], etc. Le logiciel GuitarStack[133] lui aussi reprend les mêmes modèles de pédales et recrée un amplificateur de manière très simple. Les travaux les plus avancés que nous avons trouvé sont (1) une application Google Chrome nommée GuitarFX[134] qui propose des modèles d'amplificateurs simples avec un ensemble de pédales d'effets audio, mais ne recrée pas en détail chaque étape d'un

[127] Le nœud WaveShaper représente une fonction de transformation non linéaire du signal. Il utilise une courbe pour appliquer une distorsion plus ou moins importante sur la forme d'onde au signal. En fonction de la forme de la courbe, les harmoniques générées dans le signal sortant diffèrent et peuvent donner un son plus chaud ou plus froid. On appelle cette technique le *wave shaping*, en anglais.
[128] G2 Workshops and tutorials : Waveshaping and Distortion.
https://rhordijk.home.xs4all.nl/G2Pages/Distortion.htm
[129] https://pedals.io
[130] https://github.com/Tonejs/Tone.js/
[131] https://github.com/Theodeus/tuna
[132] https://alemangui.github.io/pizzicato/
[133] https://github.com/lucaong/guitarstack
[134] https://tinyurl.com/ljdhuqh, plus maintenue.

véritable amplificateur (un seul nœud WebAudio WaveShaper est utilisé par exemple, aucune modélisation de l'étage de puissance). Les stations de travail audionumériques commerciales basées web comme soundrap.io et bandlab.io proposent aussi des simulations d'amplificateurs, mais une vue du graphe WebAudio dans le debugger de Google Chrome ou de Mozilla Firefox montre des modèles très simplifiés (on ne dispose pas du code source, ce sont des applications commerciales).

10.3 Modélisation des différents étages

10.3.1 Préamplificateur

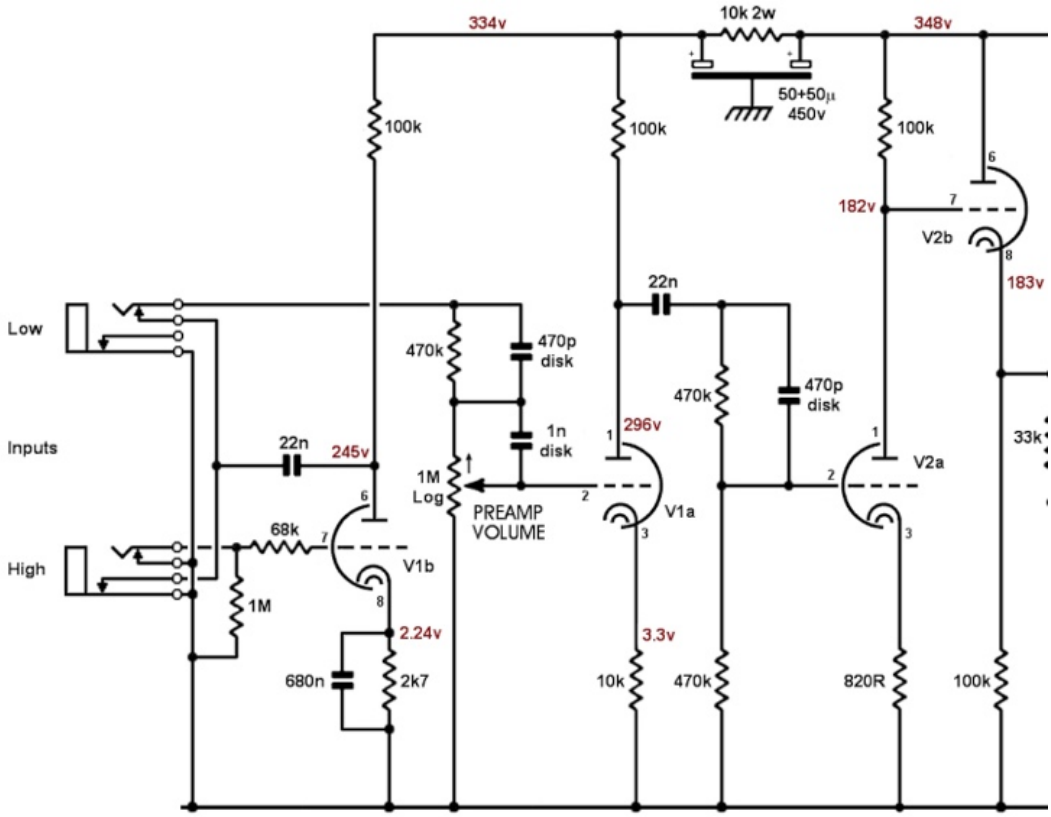


Figure 143. Partie « préamplificateur » du JCM800. Les filtres sont des couples capacité/résistance, les lampes sont des demi-cercles (V1a, V1b, V2a, V2b).

Analyse et simulation du schéma électronique

La Figure 143 présente le schéma électronique du préamplificateur du Marshall JCM800. Il est composé de plusieurs filtres et de deux doubles triodes (V1 et V2, typiquement deux lampes de modèle 12AX7). La seconde double triode, qui se compose de deux lampes v2a et v2b, située à la fin de du trajet suivi par le signal dans le préamplificateur (à droite de la figure), limite l'écrêtage et agit comme un driver linéaire de l'étage suivant : le tonestack (voir section 10.3.2). La partie la plus intéressante est constituée des filtres et de la première double triode V1a et V1b. De la gauche de la chaîne de signal vers la droite, nous avons un filtre passe-bas avec un gain de -3.3dB à 720Hz, puis un autre passe-

bas à -6dB 320Hz, puis le premier étage de triode nommé V1b, suivi d'un filtre passe-haut à 6-7Hz. C'est la première partie du préamplificateur.

Les filtres *low shelf* qui suivent coupent les fréquences gênantes générées par la guitare, et la partie V1 génère des harmoniques impaires et régulières (V1a est essentiellement un gain, V2 amplifie et introduit des harmoniques).

Les harmoniques paires sont importantes pour percevoir le son comme « chaud, *bluesy* », alors que les harmoniques impaires donnent un son plus « dur, *gritty* ». Le mélange d'harmoniques paires et impaires est connu pour être le secret du « *warm punchy sound* » que l'on trouve dans le « classic rock » et dans le « blues ».

Pour modéliser les deux parties de cette première triode, nous avons utilisé un nœud WebAudio de type WaveShaper avec une fonction de transfert asymétrique décrite dans l'article de Pakarinen-Yeh [42], présentée Figure 144 (dont les origines proviennent d'un brevet détenu par Doidec et al. 1998), illustrée à gauche de la Figure 145.

$$f(x) = \begin{cases} 0.630035, & 0.320018 \le x \le 1 \\ -6.153x^2 + 3.9375x, & -0.08905 \le x < 0.320018 \\ -\frac{3}{4}\left\{1 - \left[1 - (|x| - 0.032847)\right]^{12} + \frac{1}{3}(|x| - 0.032847)\right\} + 0.01, & -1 \le x \le 0.08905 \\ -0.9818 & x < -1 \end{cases}$$

Figure 144. Fonction de distorsion asymétrique (Pakarinen, Yeh)

Cette fonction clippe (écrête) différemment les parties négatives et positives du signal (« hard clipping » sur la partie négative, tandis que la partie positive de l'onde subit un « soft clipping »). Cette asymétrie permet de simuler le rapport-cyclique de modulation d'un amplificateur à tube en saturation. Elle ajoute des harmoniques paires et impaires, résultant en une sonorité plus riche qui caractérise les amplis à lampes « vintage » (années 60/70). On observe qu'après le passage dans cette lampe à fonction de transfert asymétrique le signal de sortie n'est plus centré car l'asymétrie ajoute à la valeur DC (la Figure 145 montre dans sa partie gauche la fonction de transfert asymétrique et juste à droite un exemple de sinusoïde pure transformée par cette fonction ; en vert le signal résultant, il présente bien une dissymétrie verticale et un décentrage).

Le filtre passe-haut à 6-7Hz qui se trouve juste après a pour fonction de « redresser / rectifier » (recentrer) ce signal et élimine le bourdonnement qui aurait pu être amplifié.

La seconde partie du préampli est constituée d'un autre filtre passe-bas à 720Hz avec un gain de -6dB. Cette fois, pour le deuxième étage de triode V2, nous avons utilisé une fonction de transfert symétrique (dans l'exemple, réalisé avec une fonction *tanh*) pour générer des harmoniques plus uniformes.

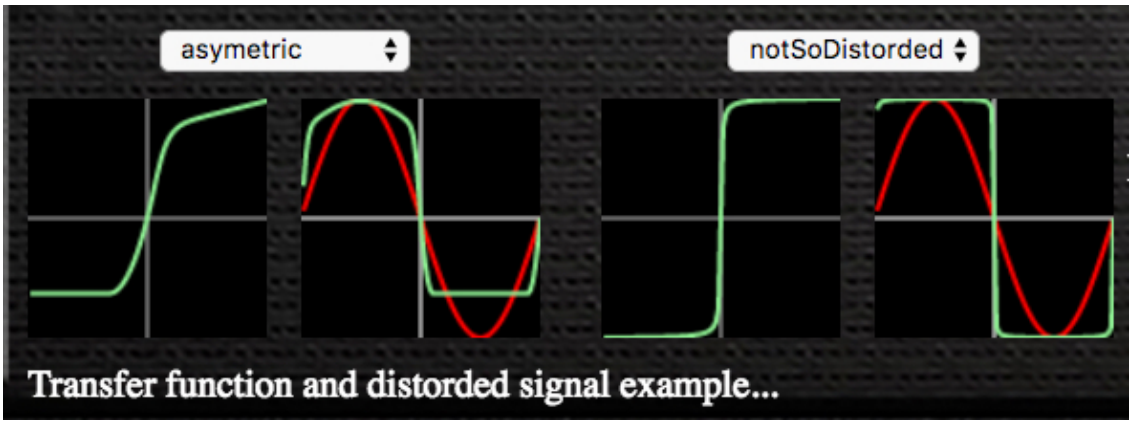


Figure 145. Fonctions de transfert utilisées pour les deux premiers tubes du préampli JCM800 (v1a et v1b)

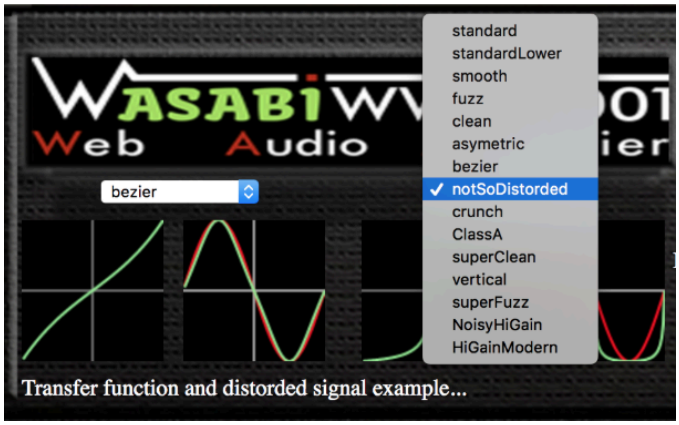


Figure 146. Nous proposons 15 fonctions de transfert différentes.

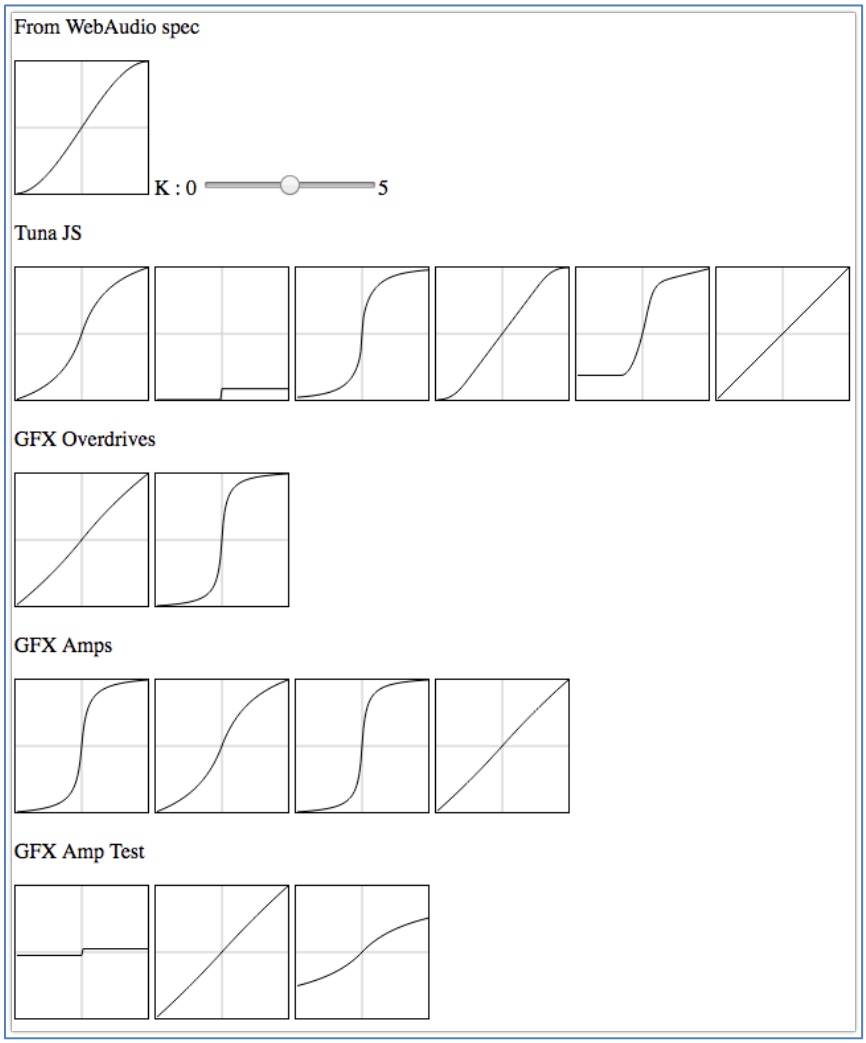


Figure 147. Exemple de fonctions de transfert paramétrables que nous avons testées et inclues dans la simulation des lampes[135].

Dans notre implémentation, nous avons inclus 15 fonctions de transfert différentes issues de la littérature consacrée aux modèles de lampes et à leurs approximations par les techniques *de wave shaping* (Figure 146 et Figure 147), qui peuvent être associées aux différents tubes de préampli (V1 ou V2), donnant ainsi plus de polyvalence au design original du préampli JCM800. Celles montrés dans la Figure 145 produisent un son riche en harmoniques paires et impaires, avec un haut niveau de distorsion/overdrive, qui peut être entendu dans les vidéos avec les guitaristes qui ont testé l'ampli simulé[136]. Ce sont les fonctions qui produisent le traitement le plus proche de la configuration classique du JCM800.

Tous les filtres ont été implémentés à l'aide de filtres biquad standards inclus dans l'API WebAudio.

[135] L'application que nous avons développée permettant de « voir » ces fonctions et de les paramétrer est disponible en ligne sur : http://jsbin.com/dijifi/edit?html,output

[136] https://wasabi.i3s.unice.fr/AmpSim3/userEvaluation.html, chercher aussi « WebAudio guitar » sur YouTube pour voir de nombreuses autres vidéos.

Il faut faire attention : pendant l'écrêtage, ces fonctions de transfert peuvent produire des harmoniques qui dépassent la moitié de la fréquence d'échantillonnage et qui peuvent créer des réplicas spectraux dans les hautes fréquences, provoquant des artefacts d'aliasing par repliement dans le spectre du signal [42]. Certaines méthodes, comme le *sur-échantillonnage*, permettent de supporter une plus large bande de fréquences avant que l'aliasing n'apparaisse. Le nœud WebAudio WaveShaper possède une propriété nommée « oversampling », qui malheureusement augmente la latence (en fonction de sa valeur "2x" ou "4x"). Une autre approche pour limiter l'aliasing est connue sous le nom de *distorsion multi-bandes* et consiste à diviser le signal en plusieurs bandes de fréquences séparées et à appliquer différents types de distorsion dans chaque sous-bande [48]. Nous avons implémenté cette approche dans une version précédente de la simulation [187], qui produisait en effet de très beaux sons propres. Cette version basée sur un design original, non inspiré d'un amplificateur existant, peut être essayée en ligne[137].

Le préamplificateur que nous avons mis en place pour le travail présenté dans ce chapitre intègre le sur-échantillonnage en option, mais nous avons également coupé les fréquences de repli / artefacts à l'aide de filtres et mesuré/constaté que, l'étage de simulation de haut-parleur que nous avons implémenté (voir section 10.3.5) coupe lui aussi la plupart des hautes fréquences qui peuvent résulter de l'aliasing (les convolutions simulant les haut-parleurs sont aussi des filtres déguisés). Ces mesures ont été confirmées lors d'évaluations qualitatives, car aucun de nos testeurs humains ne pouvait entendre une différence, avec ou sans sur-échantillonnage activé.

Vers un préamplificateur personnalisable

Dans un premier temps nous nous sommes contentés de tester la reproduction fidèle du JCM800, mais par la suite, nous avons repensé la simulation développée afin de pouvoir supporter plusieurs conceptions de préamplificateurs. D'un modèle à l'autre, le nombre et le type de lampes et de filtres, peut changer. De même, les lampes possèdent un « réglage du bias » qui peuvent affecter la manière dont elles vont modifier le signal. Ce réglage se traduit par une modification subtile de la fonction de transfert associée. Nous avons ainsi modifié l'implémentation de l'étage de pré-amplification afin qu'il soit aisément personnalisable, soit pour reproduire ce que l'on trouve sur d'autres modèles d'amplificateurs existants, soit pour explorer des conceptions qui n'existent pas (car trop onéreuses –avec un nombre important de lampes, ou trop expérimentales –produisant des sons très typés).

Le nombre et le type de tubes / filtres associés est personnalisable : nous avons ajouté la possibilité de jouer interactivement avec le nombre d'étages à lampes en cascade (Figure 148) dans le préampli, de modifier le type de tube utilisé (triode, beam-tetrode, pentode, etc.) en changeant les courbes de transfert du WaveShaper. Les courbes peuvent être symétriques, asymétriques, coupées ou même paramétriques de style Bézier avec

[137] Multiband distortion preamp in a WebAudio amp sim: https://mainline.i3s.unice.fr/AmpSim

contrôle de bias [194] [195] [189]. En règle générale, plus de lampes signifie un son plus « épais ».

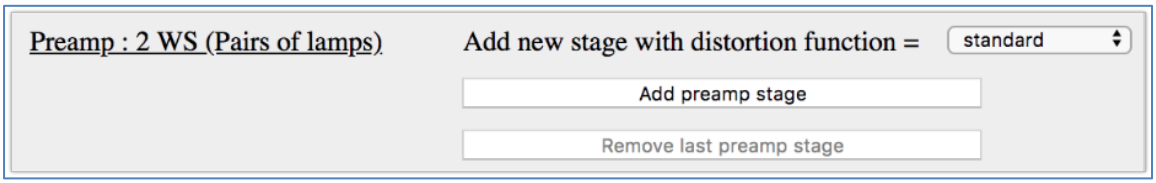


Figure 148. Le nombre de lampes et fonction de transfert par lampe est personnalisable.

Introduction d'un éditeur de fonction de transfert : les courbes des nœuds WaveShaper utilisés dans la simulation des lampes peuvent maintenant être ajustées (à l'aide de la souris ou des doigts), même en jouant, permettant un feedback instantané - vous pouvez entendre le résultat de vos modifications en temps réel (ce qui est intéressant pour la pédagogie) et vous pouvez voir l'impact sur les harmoniques générées grâce à certains analyseurs de fréquence en temps réel que nous avons intégrés dans l'interface graphique (on les voit dans la Figure 131 page 273). Ces analyseurs permettent de sonder le signal à différents endroits du circuit de l'amplificateur.

Basé sur les courbes de Bézier, l'éditeur interactif de fonction de transfert permet d'ajuster le bias (il suffit de cliquer et de faire glisser le point de contrôle central pour rendre la courbe asymétrique), et offre de nombreux raccourcis clavier pour « régler la pente », « forcer la courbure », etc. Des contraintes sont prises en compte afin de rester dans la plage des fonctions réalistes de transfert de tubes [189]. Une version de test de cet éditeur peut être essayé en ligne[138].

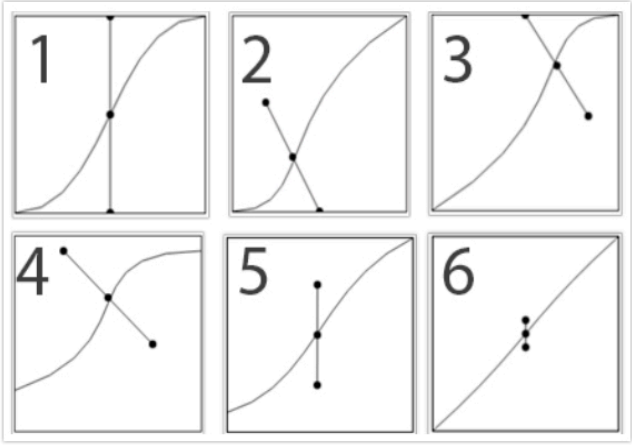


Figure 149. Editeur interactif de courbes de transfert des Wave Shapers.

Support de différentes topologies de préamplificateurs : la simulation permet de passer d'un modèle entier de préamplificateur à un autre. Par exemple, on peut passer du préamplificateur du JCM800 illustré par la Figure 150 (graphe WebAudio correspondant au schéma électronique de la Figure 141 page 283), à un préamplificateur « multi-

[138] Démonstration en ligne de l'éditeur de courbes de transfert : https://output.jsbin.com/zotaver

bandes » qui donne des couleurs sonores radicalement différentes, des sons plus cristallins que nous n'aurions pu obtenir autrement [48] (Figure 151). Le premier met en œuvre des nœuds WaveShaper et des filtres en série et reprend une topologie classique que l'on retrouve chez Fender, Marshall, Vox, Mesa Boogie etc., là où l'autre effectue des traitements en parallèle, par bande de fréquence, et permet d'obtenir des sons très clairs proches d'un amplificateur Roland jazz Chorus par exemple. D'autres topologies seront ajoutées à l'avenir.

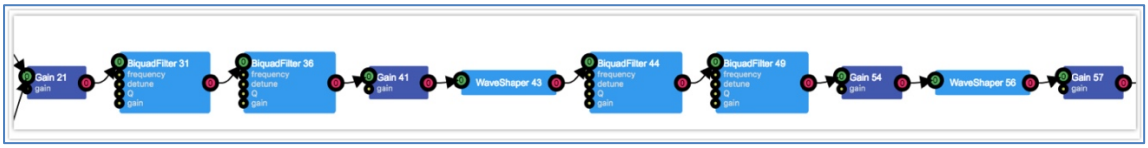


Figure 150. Préamplificateur avec topologie en série. Ici un exemple de préamplificateur à deux étages, chacun composé d'un nœud WaveShaper (bleu clair, fin), de deux filtres (blues clairs, épais) et d'un gain (bleu foncé).

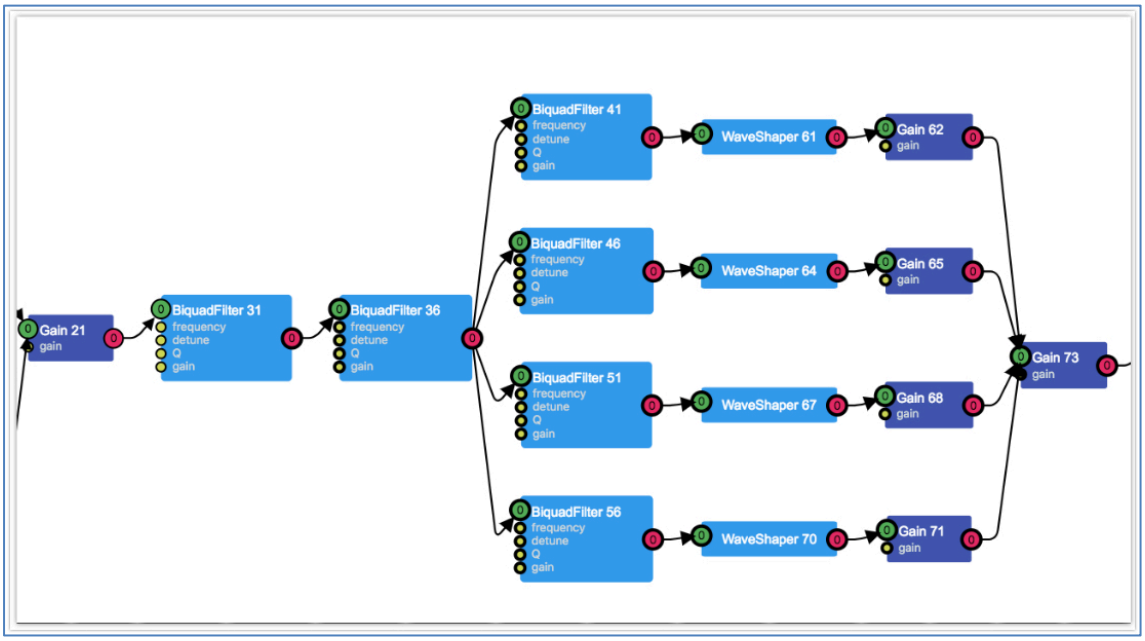


Figure 151. Préamplificateur multi-bandes (filtres + WaveShapers en parallèle) pour limiter les effets d'intermodulations.

Quatrième amélioration - la pente des courbes de la fonction de transfert peut être pilotée en temps réel par la variation d'amplitude de l'enveloppe du signal d'entrée : l'amélioration de *la dynamique* de la réponse en temps réel dans les simulations est cruciale pour les rendre plus réalistes. Ceci se traduit par de meilleures sensations guitare en main : lorsque la dynamique est bonne, on « contrôle » mieux la variation de la distorsion présente dans le son en fonction de l'attaque des cordes de la guitare électrique. Les amplificateurs à lampes sont réputés pour avoir une meilleure dynamique que les amplificateurs à transistors, moins onéreux.

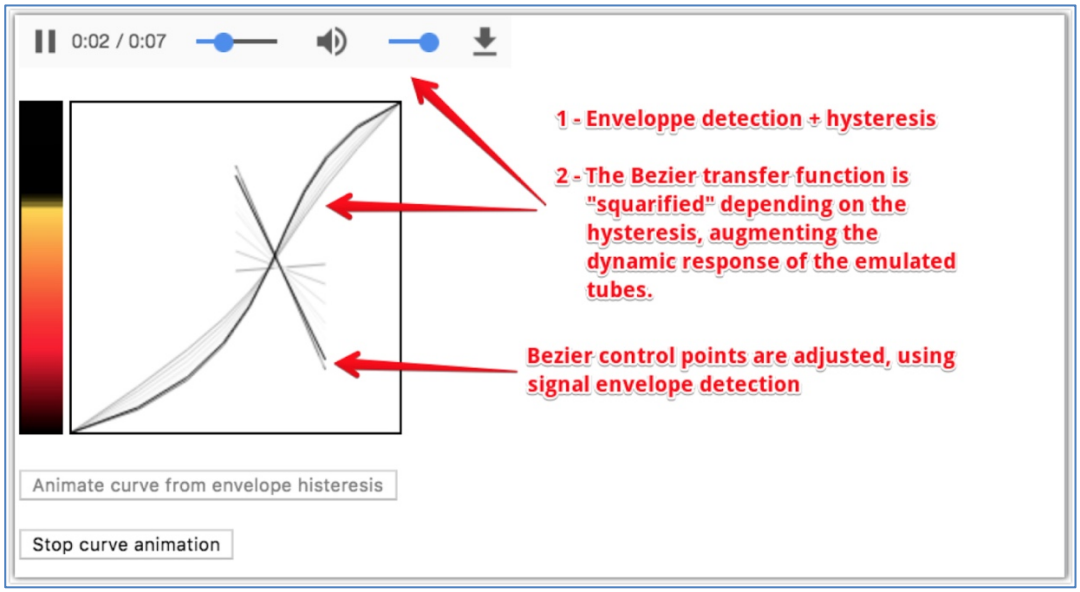


Figure 152. Augmenter la réponse dynamique des lampes émulées, en utilisant la détection de la puissance d'enveloppe pour changer la pente des courbes de transfert.

En effet, lorsqu'ils sont poussés à leurs limites par de grands transitoires, les amplificateurs de guitare à lampes - contrairement à ceux à transistors -, ont une tendance à la saturation harmonique avec compression dynamique (cet effet est appelé *« sag »* dans le jargon) et introduisent un léger retard temporel (cet effet est appelé *« squish »*).

L'effet *sag* est consécutif à une chute de la tension d'alimentation dans le préamplificateur en réponse à *des attaques* transitoires importantes[139]. Quant à lui, l'effet *squish* est lié à une hystérésis temporelle induite par un décalage temporel dans la boucle de rétroaction - également en réponse à de grands transitoires lors de l'effet *release*.

Ces effets sont assez délicats à émuler en temps réel[140]. Par conséquent, pour approcher l'effet *sag* dans la réponse, nous avons implémenté des changements dynamiques en temps réel dans les pentes des fonctions de transfert. En ajustant correctement le seuil auquel on obtient l'effet *squish*, seuls les transitoires les plus importants de l'amplitude de l'enveloppe du signal d'entrée déclencheront un changement de pente contrôlé. La Figure 152 montre une courbe animée en temps réel avec comportement d'hystérésis dans la fonction de transfert déclenchée par l'enveloppe du signal d'entrée. Le logiciel que nous avons développé pour tester cette idée est disponible en ligne[141].

[139] Plus de détails surs : http://www.aikenamps.com/index.php/what-is-sag
[140]Exemples : http://www.diale.org/tube_emulation.html
[141] https://output.jsbin.com/zotaver. Voir aussi cette vidéo de l'utilisation de cette technique dans la véritable simulation : https://www.youtube.com/watch?v=zBhn7odezUQ

10.3.2 Tonestack (partie des réglages basses-médiums-aigus)

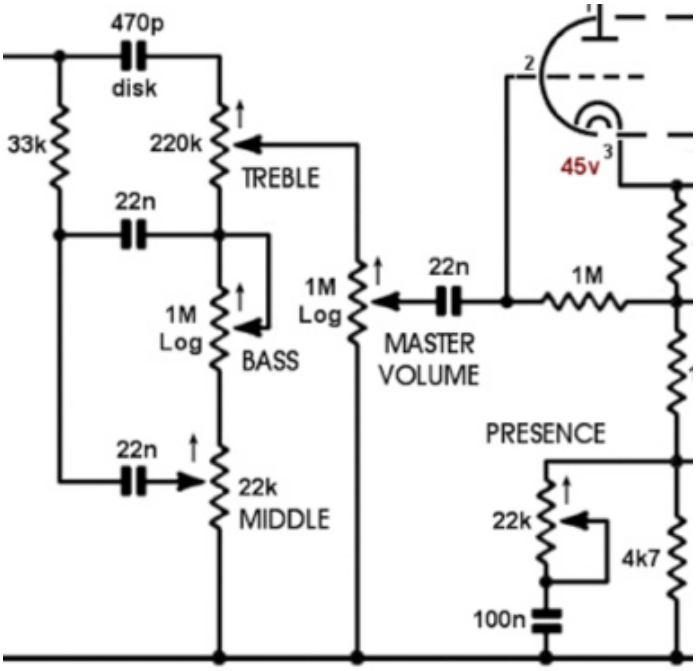


Figure 153. Le schéma du tonestack du JCM800

Dans un article de référence sur la modélisation de l'amplificateur Fender Bassman [6], David Yeh et Julius Smith expliquent que le tonestack (circuit qui gère les réglages de basses, médiums et aigus), situé avant le préamplificateur du Bassman ne peut pas être simulé avec précision par des filtres en série ou en parallèle. Outre sa simplicité et son faible coût de fabrication, ce circuit conçu par Jim Marshall, se traduit par un filtre complexe, chaque élément influençant l'autre.

Yeh et Julius ont analysé symboliquement ce schéma en utilisant la technique *virtual analog*, et ont proposé une solution exacte : un filtre qui répond aux commandes de l'utilisateur de la même manière que le prototype analogique hardware.

De nombreux amplificateurs classiques produits par Fender, Vox ou Marshall partagent cette conception du tonestack (peu onéreux à produire en série), avec quelques variantes. L'emplacement dans le circuit de l'amplificateur peut varier d'un modèle à l'autre - avant le premier étage de pré-amplification sur les amplis Fender, après sur les amplis Marshall par exemple -, ou bien on retrouve des valeurs de résistances et capacités légèrement différentes.

Dans notre amplificateur simulé en WebAudio, nous avons dans un premier temps transcompilé l'implémentation C++ originale du code de Yeh[142] en JavaScript, en utilisant l'outil *emscripten* [188], et nous avons utilisé un nœud WebAudio ScriptProcessor pour reproduire le tonestack exact du JCM800. Cependant, les expérimentations ont montré qu'au minimum 4ms de latence sont introduits, c'est pourquoi nous avons décidé d'explorer d'autres alternatives. Par ailleurs, nous avons parfois rencontré des « glitches

[142] Disponible ici : http://quitte.de/dsp/caps.html - ToneStack

audio » - des coupures de son pendant que les utilisateurs utilisaient les potentiomètres de réglage des basses, mediums et aigus de l'interface graphique de l'amplificateur - (le code des nœud ScriptProcessor fonctionne dans le thread de la GUI), ce qui nous a conduit à supprimer temporairement cette implémentation. Nous avons adopté dans un premier temps, comme solution alternative, un ensemble de filtres biquad en série : le filtre aigu (de type « high shelf », avec une fréquence de coupure de 6.5 KHz) entre dans le filtre médium (de type « peaking », 1.7KHz) puis dans le filtre des basses (« low shelf », 100Hz). Nous avons ajusté les types de filtres et leurs paramètres afin qu'ils approchent la réponse en fréquence du vrai tonestack du JCM 800 (nous avons utilisé l'outil *tonestack calculator*[143] pour comparer notre approximation avec la réponse en fréquence du véritable tonestack, voir Figure 154). Le résultat n'est pas une réplique fidèle du circuit original, mais s'en approche : nos testeurs l'ont trouvé facile à utiliser et ont réussi à façonner leur son rapidement.

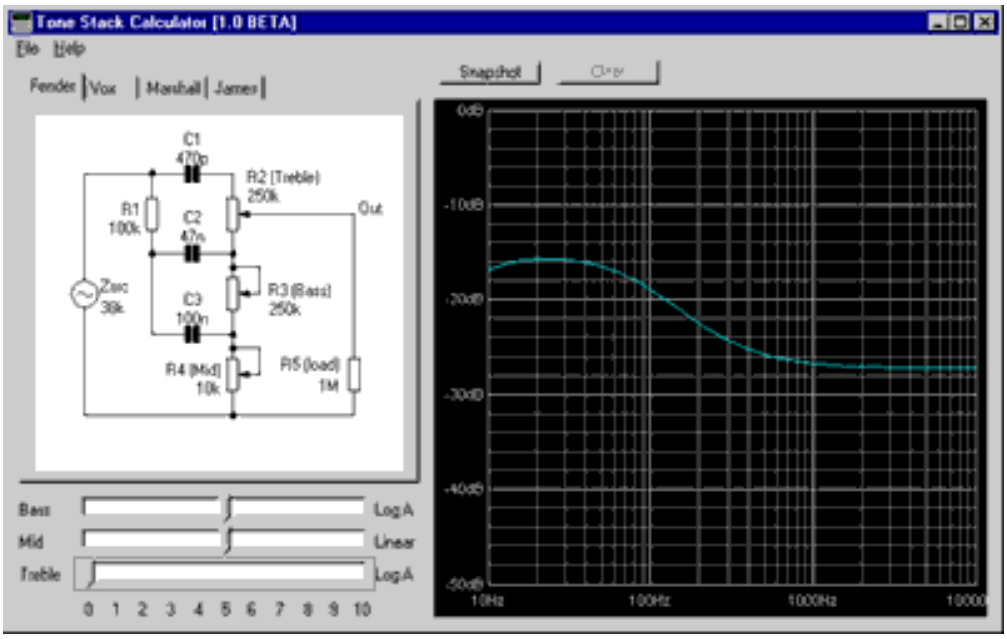


Figure 154. L'outil *tonestack calculator* permet d'étudier la réponse en fréquence des tonestacks des principaux amplificateurs du marché.

WebAudio possède un nœud spécialement adapté à des traitements bas niveau censé remplacer le nœud ScriptProcessor, mais sans ses défauts : le nœud AudioWorklet [196]. Ce dernier nœud, contrairement au ScriptProcessor, s'exécute avec une priorité plus grande que le thread de la GUI du browser et théoriquement, doit résoudre les problèmes de performance et de latence. Les premières implémentations datent de Mai 2018 et nous avons rapidement pu recréer le tonestack originel.

La dernière version du tonestack fonctionne maintenant en WebAssembly à l'intérieur d'une AudioWorklet et est entièrement personnalisable : nous avons profité du standard de plugins WebAudio que nous avons développé (voir chapitre 11 de ce manuscrit) pour intégrer un ensemble de variantes du tonestack de la Figure 153 codée

[143] http://www.duncanamps.com/tsc/

en FAUST et trans-compilées en AudioWorklet/WebAssembly. FAUST est un langage spécialisé pour la programmation d'algorithme de traitement du signal (DSP), et peut se compiler vers diverses cibles [197]. Nous avons travaillé avec les auteurs de ce langage pour générer directement des plugins WebAudio dans le format que nous avons développé en commun, appelé WAP (WebAudio Plugin) [198].

Cette approche permet de jouer avec un tonestack qui est maintenant fidèle à la version originale basé sur le design présenté dans la Fig. 8 (Marshall, Fender etc.). Avec cette approche, on peut aussi expérimenter facilement de nouvelles topologies pour le réseau de filtres composant cet étage de l'amplificateur.

Deuxième amélioration - l'interface graphique permet de choisir « à la volée » parmi les différentes implémentations / réglages de ce tonestack : nous pouvons maintenant sélectionner un tonestack Fender, Marshall, Vox ou autres…, ou utiliser un modèle différent comme celui que nous avions dans nos premières expérimentations, avec des filtres en série.

Troisième amélioration – le tonestack et le préamplificateur peuvent être échangés dans la topologie de la chaîne de signal : à savoir, plusieurs amplificateurs de guitare matériels localisent le tonestack *avant* le préamplificateur et non *après* comme le Marshall JCM800. Lorsque le préamplificateur est saturé, placer le tonestack avant permet d'obtenir des sons « plus durs / métalliques modernes » en boostant certaines fréquences avant cette première amplification, et des sons plus propres et plus étincelants quand le préampli n'est pas saturé[144]. Un tonestack placé après le préampli « lisse » l'effet de ce dernier.

[144] Quelques vidéos pour voir comment les sons à gain élevé peuvent être obtenus en échangeant les positions du préampli et du tonestack: https://youtu.be/-NdMdJQx2Bw?t=4m11svoir aussi https://youtu.be/PiOD7n3g-Qs

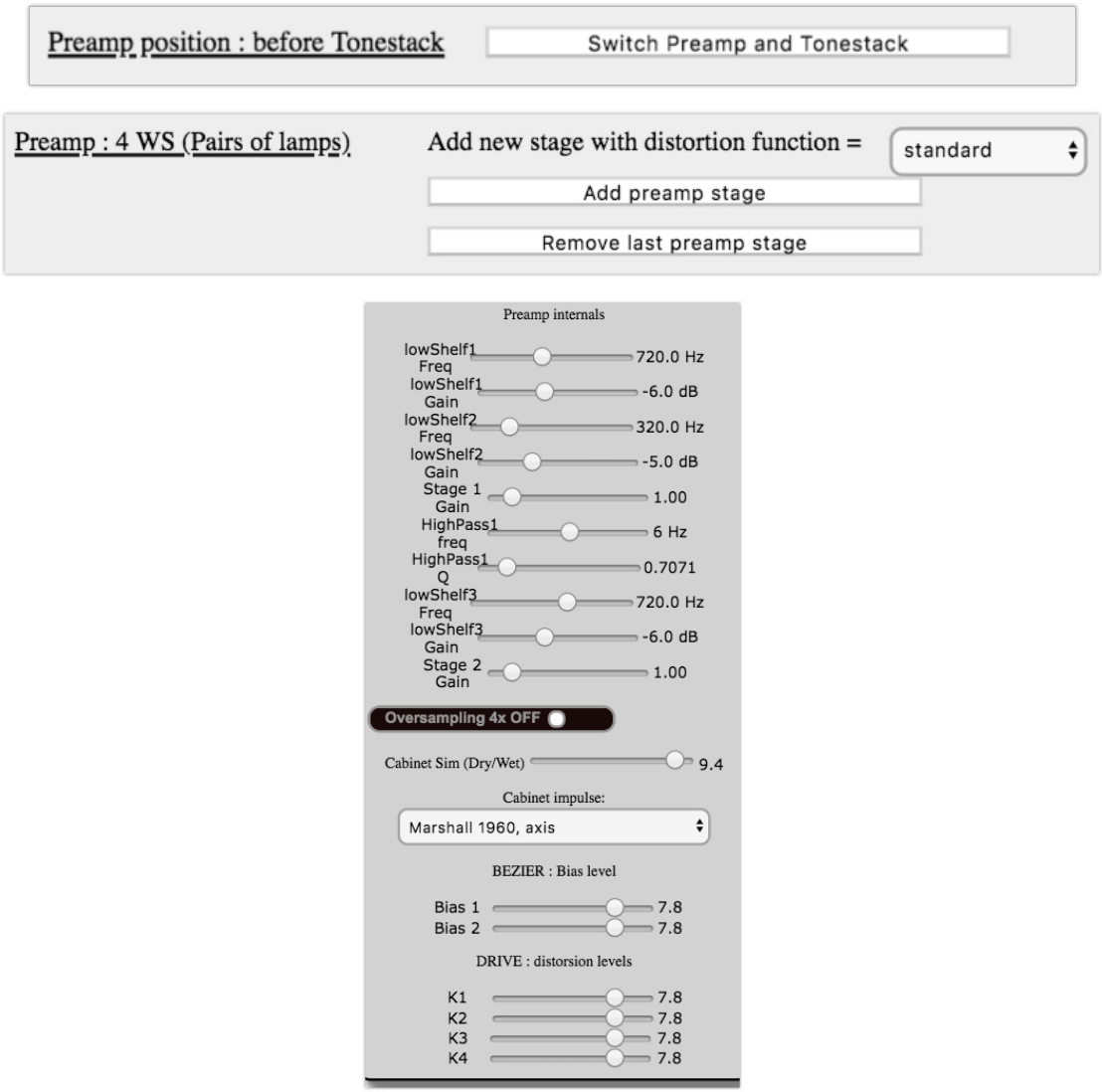


Figure 155. Ensemble des réglages « avancés » permettant de « concevoir » un préamplificateur personnalisé. Ne figurent pas ici les menus permettant d'ajuster les fonctions de transfert (Figure 146 et Figure 149).

10.3.3 Réverbération

Nous avons utilisé un nœud WebAudio de type « Convolver », permettant d'appliquer des convolutions sur le signal. Ce nœud utilise des « fichiers d'impulsions » de réverbérations, sortes d'empreintes digitales permettant de reproduire des effets que l'on obtiendrait en écoutant un son dans un endroit spécifique, par exemple « un hall » ou « une petite pièce ». On obtient ces fichiers en enregistrant un même son joué dans une chambre anechoïque puis dans une pièce produisant de la réverbération. En observant les différences obtenues, on obtient un fichier qui caractérise la pièce. De même si on compare un son produit par un amplificateur de guitare avec et sans l'effet de réverbération activé, on obtient la caractéristique de cet effet. Idem pour caractériser un micro ou un haut-parleur. On peut donc trouver dans le domaine public ou commercial des fichiers d'impulsions correspondant aux principaux micros, haut-parleurs, circuits de réverbération, lieux (hall, église, etc.). Nous avons utilisé des fichiers libres de droits

correspondant aux circuits de réverbération équipant quelques amplificateurs classiques, dont le Marshall JCM800.

Nous avons inclus ce nœud convolver dans un graphe audio classique wet/dry permettant de régler la quantité de réverbération que l'on désire ajouter au signal[145]. Voir Figure 156.

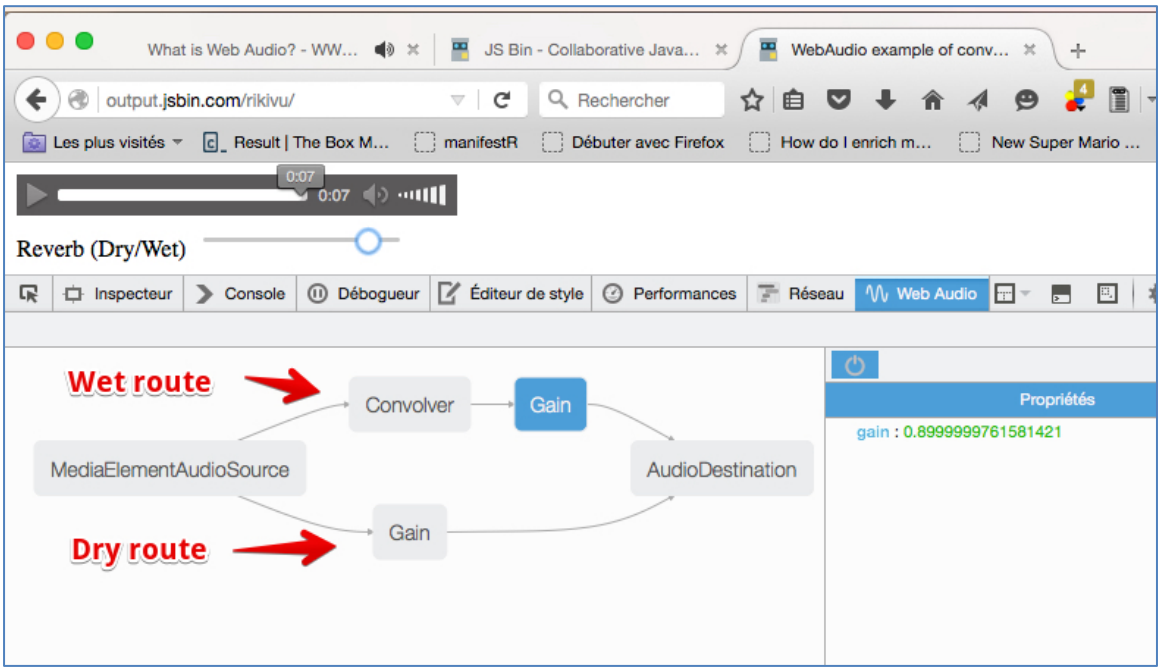


Figure 156. Schéma classique wet/dry pour l'implémentation d'une réverbération réglable. En haut le signal est réverbéré, en bas il est inchangé. La présence de deux gains (volumes) permet de mixer la quantité de signal réverbéré que l'on désire en sortie.

Afin d'incorporer les aspects non linéaires de la réverbération, d'autres approches sont prévues, comme par exemple par filtrage de Volterra ou l'utilisation de l'algorithme de Schroeder qui permet de générer des réverbérations très fortement paramétrables (voir [222] pour une synthèse de ces méthodes).

10.3.4 Amplificateur de puissance

L'étage amplificateur de puissance du JCM800 (Figure 157) repose sur un séparateur de phase alimentant un circuit push-pull classique de pentodes (deux lampes de type EL34 ou 6550).

La 2ème harmonique (et toutes les harmoniques paires en général) sont naturellement annulées par un push-pull bien équilibré [189]. De plus, par rapport aux triodes, l'utilisation de pentodes améliore les troisièmes harmoniques (et toutes les harmoniques impaires) [190]. Nous nous retrouvons ainsi avec un spectre composé principalement d'harmoniques impaires qui pourraient être raisonnablement simulées à l'aide de fonctions de transfert symétriques[146]. Cependant, si l'on veut réintroduire des harmoniques paires

[145] Testable ici : https://jsbin.com/belide/edit?html,js,console,output
[146] Millet P. "The sound of distortion", online tutorial. http://www.pmillett.com/file_downloads/ThesoundofDistortion.pdf

pour obtenir une version plus chaude du JCM800, il faut plutôt utiliser des fonctions de transfert asymétriques, ce qui produit une bonne simulation de l'utilisation classique de lampes (pentodes) légèrement mal assorties pour obtenir des harmoniques égales à partir d'un push-pull déséquilibré. Cette technique « d'appairage approximatif des lampes » était prisée par certains guitaristes comme Jimi Hendrix, pour « salir le son ». Nous avons ajouté un gain faible (< 1) avant notre étage d'amplification de puissance pour reproduire l'effet d'overdrive "light" attendu. Comme pour la partie puissance des amplis analogiques réels, en plus de l'overdrive, une influence sur la tonalité est liée à l'impédance des haut-parleurs de l'enceinte. Cette impédance est due à la charge des lampes de puissance et donc "façonne" le son. Ce document [191], écrit par le créateur des amplificateurs à lampes Randall, donne une idée générale de l'impédance d'un haut-parleur et comment la modéliser.

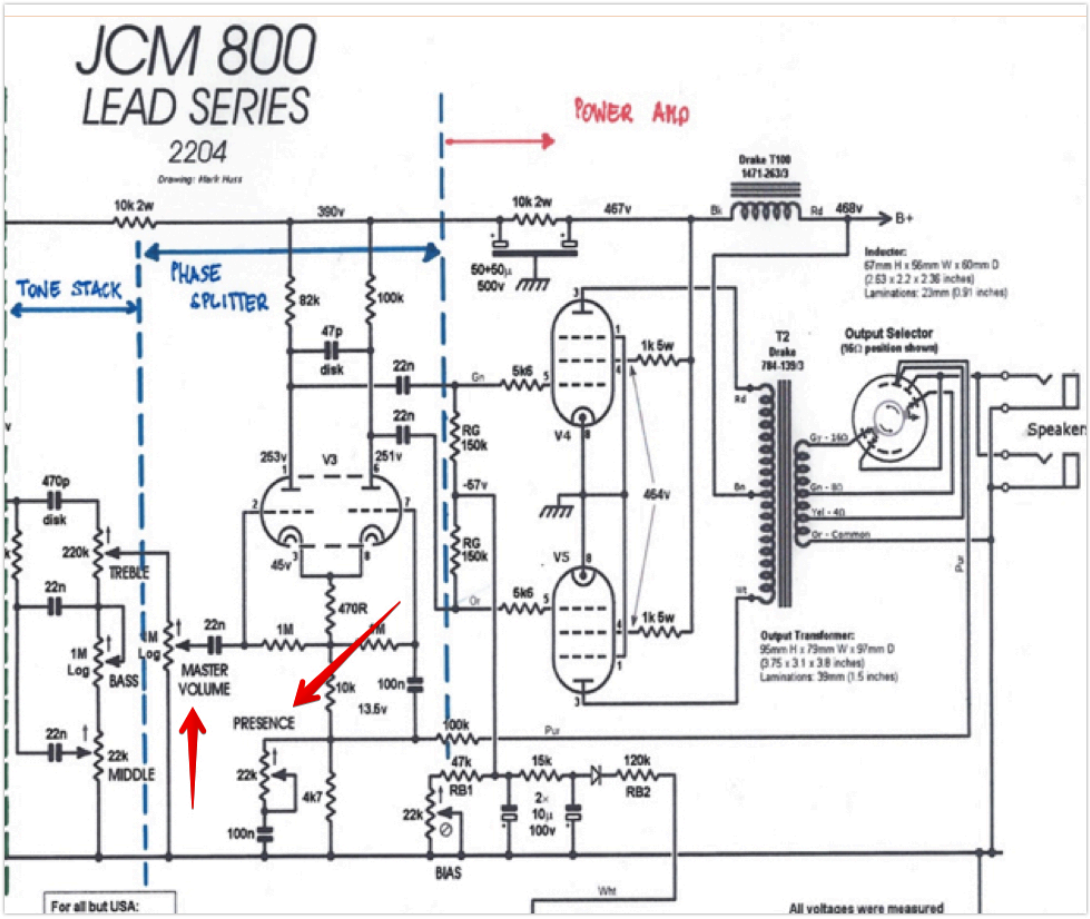


Figure 157. Amplificateur de puissance du Marshall JCM800. Les flèches rouges montrent le réglage du Master Volume en entrée de l'étage de puissance (Power Amp) et le filtre de « Présence », qui figure dans une boucle de feedback négatif destinée à diminuer ou augmenter certaines harmoniques présentes dans le signal d'entrée.

Dans un premier temps, nous avons opté pour une simulation simplifiée de l'étage de puissance avec un modèle composé d'un gain (le Master Volume) et d'un wave shaper symétrique en série (pour simuler les lampes de puissance). Le filtre de « présence » qui fait partie du Power Amp a été simplement rajouté en série sous la forme d'un filtre de

type « peaking » à 3900 Hz. Les résultats étaient très satisfaisants, l'amplificateur simulé réagissait bien, sonnait bien, les réponses en fréquence assez proches de l'amplificateur réel (section 10.4.2) et les tests utilisateurs positifs (voir section 10.6).

Cependant, comparé à la meilleure simulation de notre comparatif (Guitar Rig) ou au véritable amplificateur, les réglages de Master Volume et de Présence, deux des potentiomètres qui agissent sur l'amplificateur de puissance, ne répondaient pas de manière fidèle, et ne permettaient pas de régler finement la réponse dynamique de l'amplificateur.

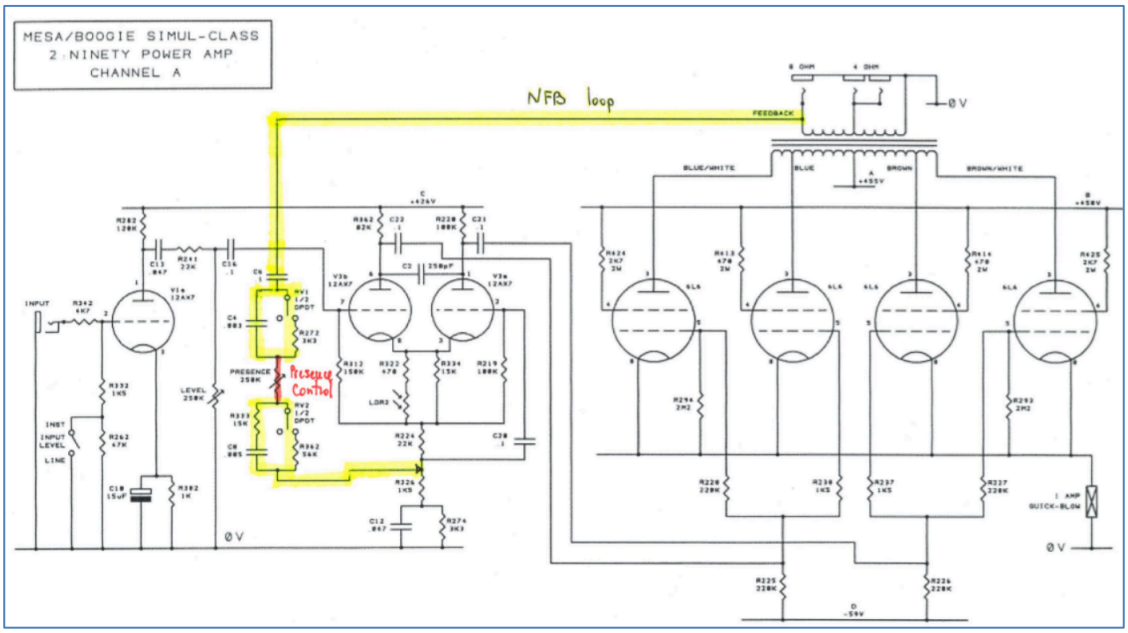


Figure 158. L'étage de puissance du Mesa/Boogie Simul-Class 2:90. La boucle de rétroaction négative (*Negative Feedback Loop* ou NFB) est en surbrillance.

Nous allons maintenant expliquer pourquoi l'amplificateur de puissance est une étape critique à émuler et comment nous y sommes parvenus, malgré les inconsistences identifiées dans les implémentations de l'API WebAudio que nous avons rencontrées dans les différents navigateurs Web.

L'étage de puissance, contrôlé par les boutons Master Volume et Présence, a un impact critique sur le son et sur la dynamique globale, car le type de son que vous pouvez obtenir uniquement à partir de l'étage de pré-amplification reste pauvre en termes de distorsion. Si on se réfère à Kuehnel [195] et Denton [189], quand on monte le volume d'un ampli à lampes en utilisant le bouton Master Volume, les tubes de puissance distordent de plus en plus et le son devient de plus en plus épais et moins contrôlé. Et au fur et à mesure que l'on se rapproche de la saturation de puissance (écrêtage des tubes de puissance et/ou saturation du noyau du transformateur de sortie jusqu'au fameux effet *grinding*, avec son mélange de saturation/oscillations), la dynamique de sortie se restreint également, entraînant des distorsions *muddy* et un son *plus lourd*. Le jargon anglais est difficile à traduire. On parlerait plutôt en français de son « épais », « lourd », « métallique », « abrasif » (traduction de *gritty*), « riche ».

Boucle de Feedback Négatif (NFB) et Présence : dans la plupart des amplificateurs, de puissance, pour le prix d'un gain total plus faible, une boucle de feedback négatif (NFB) est ajoutée afin de nettoyer le son de sortie de certaines distorsions indésirables. La NFB est obtenue en prélevant une partie du signal en sortie du transformateur et en le renvoyant avec un déphasage de 180 degrés dans l'étage de séparation (voir le tracé en surbrillance de la Figure 158). Les amplis sans NFB peuvent être plus puissants (gain plus élevé) avec plus de distorsion, mais ils deviennent aussi plus imprévisibles[147] lorsqu'on les pousse à la saturation. La boucle NFB lisse le son en réduisant le niveau de distorsion dans les parties du spectre où il est le plus dérangeant (typ. mid-range) [195].

Maintenant, dans les amplis de guitare avec NFB, le contrôle de présence vise à apporter un moyen de régler la brillance et de la netteté du son dans l'étage de puissance. Le comportement de cette commande est très différent de celles présentes dans le tonestack (graves, mediums, aigus), qui ne fait que façonner la sortie du préampli. La présence fonctionne comme un contrôle holistique de la brillance du timbre par la modification du signal dans la boucle de rétroaction négative. Les effets du feedback négatif sont réduits à partir d'une certaine fréquence (déterminée par le réseau RC dans la boucle). Avec moins de contre-réaction négative dans les hautes fréquences, le son devient plus fort, plus vif et dynamique. Contrairement à un contrôle des aigus classique qui est soustractif, le contrôle de présence est pseudo-additif en ce qu'il limite l'aspect soustractif de la boucle de NFB.

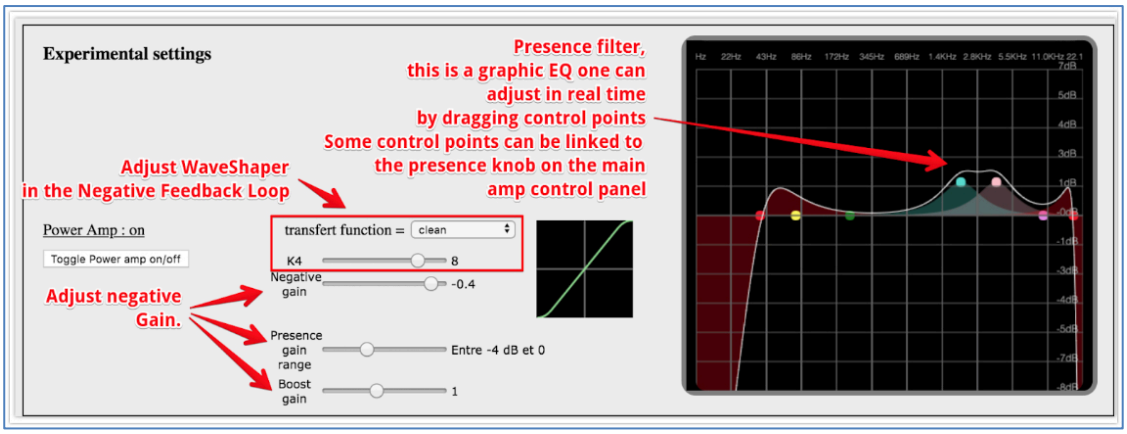


Figure 159. Les réglages de l'amplificateur de puissance NFB peuvent être modifiés/ajustés dynamiquement à la volée sans problème.

Dans l'implémentation actuelle, le filtre de présence est entièrement personnalisable et peut être contrôlé en temps réel à l'aide d'un égaliseur graphique (Figure 159) pour choisir la plage de fréquence du filtre. Les valeurs min et max du gain négatif dans la NFB peuvent également être ajusté à l'aide d'un curseur. Ces outils (NFB et Présence) sont plutôt sensibles (les larsens peuvent facilement apparaître, c'est pourquoi nous fournissons des contrôles dans la GUI pour les ajuster/restreindre) mais le contrôle de présence est aussi très puissant et spectaculaire pour façonner le son final. Nous

[147] On pourrait dire « difficiles à apprivoiser » car les réglages ne montrent aucune linéarité dans comportement, la manipulation des boutons de réglages est contre intuitive….

conseillons fortement au lecteur de visionner cette vidéo YouTube[148] qui montre bien les différences dans le son et la dynamique avec et sans cette boucle (power amp activé / désactivé) dans notre simulation.

Découverte de bugs d'implémentation du graphe WebAudio dans les navigateurs, et limitations de l'API

Les implémentations WebAudio de la NFB et de la Présence ont été difficiles à réaliser en raison de certaines limitations de l'API WebAudio et des divergences/bugs dans la façon dont les navigateurs gèrent les graphes WebAudio possédant des boucles.

Les spécifications de l'API WebAudio exigent que toute boucle dans le graphe possède au moins un nœud de type Delay[149]. Ainsi, sans la présence d'un tel nœud, Firefox stoppe tout traitement, tandis que Google Chrome ne se plaint pas mais ajoute, en coulisses, un délai de 3ms. De plus, avec WebAudio, le signal est toujours traité par paquets de 128 échantillons, ce qui signifie que la valeur minimale pour un délai est de 128 / taux d'échantillonnage, soit environ 128/44100 = 3ms. Pour une bonne mise en œuvre de la NFB, un délai d'un échantillon aurait suffi.

Maintenant, quand on ajoute un Delay de 3ms dans la boucle pour se conformer aux spécifications, cela donne une coloration du son légèrement différente avec Firefox ou Chrome. Nous avons signalé ces erreurs et discuté avec les implémenteurs (Raymond Toy de chez Google et Paul Adenot de chez Firefox) mais pour le moment, rien n'a été corrigé (Septembre 2018).

Comme perspective, nous prévoyons de ré-implémenter la boucle de NFB avec un nœud AudioWorklet pour permettre une NFB de latence inférieure à 3ms[150], typiquement en codant l'ensemble des traitements filtres, gain et wave shaper, en FAUST.

10.3.5 Simulation des haut-parleurs

Nous avons utilisé un nœud convolver avec des impulsions de « baffles » d'amplificateurs de guitare. Un « baffle » est une boîte en bois avec plusieurs haut-parleurs à l'intérieur. La Figure 160 montre le baffle Marshall 1960 équipant le JCM800 (la tête de l'amplificateur est posée dessus dans la configuration complète). Parfois cet amplificateur est équipé de deux baffles de ce type, posées l'une sur l'autre.

[148] https://www.youtube.com/watch?v=-NdMdJQx2Bw
[149] https://webaudio.github.io/web-audio-api/ - rendering-loop
[150] 128 échantillons/44100Hz = 3ms. C'est la limite basse de WebAudio actuellement, si on se restreint à l'utilisation de nœuds standards.

Figure 160. A gauche : la baffle Marshall 1960, équipée de quatre haut-parleurs de 12 pouces. A droite : la tête du Marshall JCM800.

Nous avons utilisé des fichiers d'impulsions correspondant au baffle du Marshall JCM800 (ainsi que quelques autres d'amplis Fender, Vox etc.) issus de l'ensemble commercial Redwirez[151]. Nous l'avons implémenté en utilisant un graphe audio wet/dry classique pour rendre l'effet ajustable, similaire à celui utilisé pour simuler l'effet de réverbération. Les tests et les mesures ont montré que la qualité des impulsions affecte la tonalité globale du son et permet de supprimer les réplicas spectraux introduits par les WaveShapers par filtrage passe-bas[37].

10.4 Évaluations

10.4.1 Latence

Paul Adenot de Mozilla a donné un tutoriel à la WebAudio Conference 2016 qui couvrait toutes les facettes des performances de l'API WebAudio. Ce tutorial incluait une section sur la latence [192], c'est-à-dire le temps que met le signal pour parcourir toute la chaine de traitement, depuis la sortie de la guitare jusqu'à la sortie du son par les haut-parleurs. Une latence trop élevée interdit de jouer de la guitare « en temps réel », la sensation devenant de plus en plus désagréable pour le musicien au fur et à mesure que la latence augmente. Avec un amplificateur physique, cette latence est inférieure à 1ms. Avec les simulations audionumériques, on lit souvent « qu'au-delà de 10ms » le système devient peu adapté au jeu temps réel. Nous verrons que ces valeurs sont sujettes à discussion, cette valeur, de 10ms que l'on trouve dans la littérature correspond en général à la taille du buffer audio divisée par la fréquence d'échantillonnage, et n'inclut pas d'autres éléments susceptibles de « ralentir le signal ».

En effet, la latence dépend de la carte son / du pilote logiciel de cette carte / de la taille du buffer audio / des ports USB, du système d'exploitation / du pilote audio supporté par le navigateur[152], de l'implémentation de WebAudio par le navigateur (par exemple, la

[151] http://www.redwirez.com/

[152] Par exemple, le traitement à faible latence n'est pas encore possible sous Windows car le seul pilote audio visible par les navigateurs est celui de base de l'OS (WASAPI driver) et n'est pas adapté à des traitements à faible latence.

conception multi processus de Google Chrome ajoute de la latence lors de la communication inter processus). La latence est également introduite par certains nœuds WebAudio : le nœud Delay (évidemment), le nœud DynamicCompressor ajoute une latence fixe de 6ms, les nœuds filtre biquad ajoutent une latence de deux échantillons qui est négligeable, les nœuds WaveShaper, lorsque le sur-échantillonnage est activé, ajoutent de la latence, le nœud Panner (pour régler le son 3D) ajoute de la latence en fonction de certains paramètres d'azimut et d'élévation, etc.

Dans notre implémentation, nous avons essayé de maintenir la latence aussi faible que possible : nous n'avons pas utilisé de fonctions/nœuds qui pourraient ajouter de la latence.

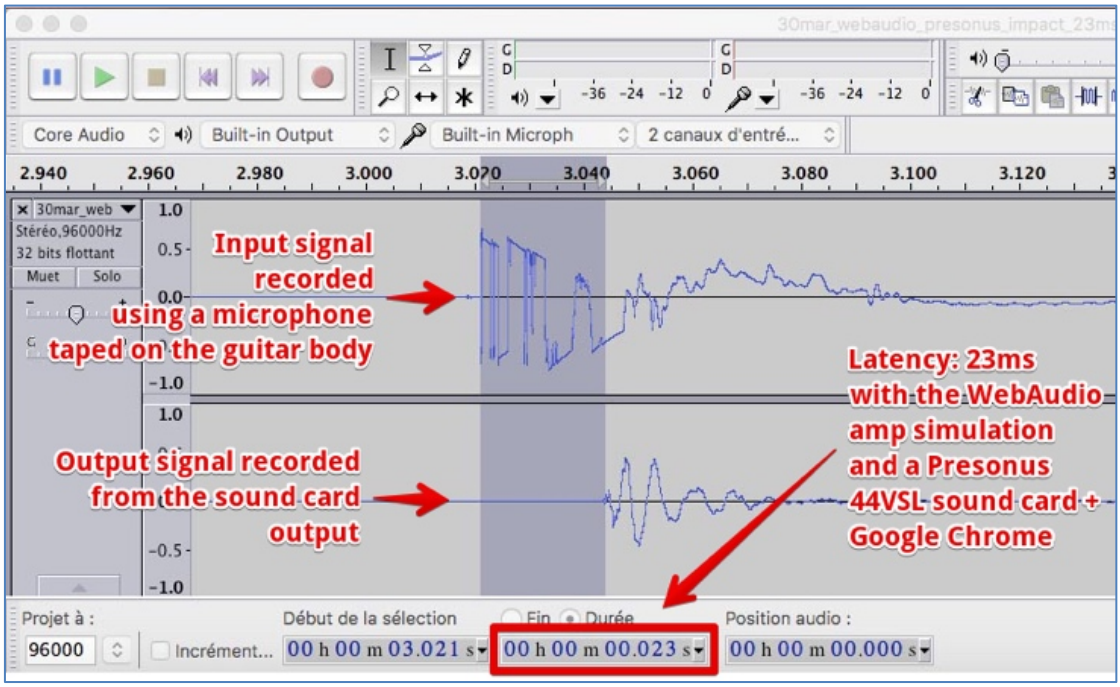


Figure 161. Exemple de mesure dans Audacity, ici l'ampli WebAudio avec Google Chrome et une carte son Presonus 44VSL.

En Juin 2018 nous avons mesuré la latence avec un Mac Book Pro 2016, Mac OSX Sierra, 16 Go de RAM, équipé d'une carte son Presonus 44VSL, mais aussi d'une carte son Apogee Jam. Nous avons lancé l'amplificateur WebAudio sur un navigateur "propre" (pas d'autres onglets et processus en cours d'exécution) et testé avec Chrome et Firefox.

Nous avons effectué nos tests sur les versions suivantes des navigateurs : Google Chrome version 59.0.3053.3 dev (64-bit) et Firefox Nightly version 55.0a1 (2017-04-01) (64-bit) pour Mac OS.

Nous avons utilisé la taille du buffer audio par défaut de 256 échantillons en double buffer, ce qui correspond à 11ms de latence intrinsèque (confirmée par la valeur de la propriété `baseLatency` du contexte WebAudio). Nous avons également mesuré la latence obtenue avec la simulation du JCM800 proposée par le plugin natif Guitar Rig (utilisé comme plugin dans la station audionumérique GarageBand), et avec le plugin Clean Brit qui est fourni avec GarageBand (également une réplique du JCM800). La taille

de la mémoire tampon est verrouillée à 512 échantillons avec GarageBand (versions >= 10), la même valeur utilisée par WebAudio dans nos configurations de test.

En outre, nous avons comparé aussi avec une pédale « Mod Duo[153] » fonctionnant sous Linux (et proposant la simulation d'amplificateur Guitarix, écrite en C++ et open source) et avec un amplificateur de guitare hardware Yamaha THR10 comme référence (Figure 162).

Figure 162. A gauche, la pédale ModDuo (Linux) et à droite l'amplificateur YamahaTHR10.

Nous avons mesuré la latence de la guitare au haut-parleur, en utilisant un microphone à électret Panasonic WM61A fixé sur le corps de la guitare, à côté de la corde de mi aiguë (Figure 163), et avec un jack branché à la sortie des cartes son. Nous les avons rejoints en utilisant un fil Y branché dans un enregistreur numérique Sony PCM-M10 à 96kHz/24bit.

[153] https://www.moddevices.com/

Figure 163. Configuration matérielle de la mesure de latence.

Nous avons ensuite frappé plusieurs fois le corps de la guitare avec une baguette métallique, et enregistré le signal résultat en passant par différentes simulations de cartes son et d'amplis (avec des réglages qui n'ont pas trop altéré le signal). La Figure 161 montre (dans le logiciel Audacity) un exemple de mesure après le chargement d'un fichier .wav que nous venons d'enregistrer. La sélection à la souris donne une assez bonne approximation de la latence. Cependant, nous avons confirmé ces mesures en utilisant notre propre script Matlab qui analyse les différentes impulsions en forme de Dirac pour calculer la valeur max/min et la valeur moyenne de la latence ainsi que les estimations du retard du groupe. Les mesures et les fichiers.wav sont disponibles en ligne[154].

Un résumé de nos mesures de latence est présenté dans le tableau 6. Nous avons comparé les latences avec un tampon audio de 256 échantillons en double buffer (eq. 512) à la fois avec WebAudio et avec des plugins natifs dans GarageBand. Il est intéressant de noter que même avec des valeurs de latence légèrement plus élevés dans les configurations WebAudio, les guitaristes n'ont pas trouvé la latence gênante avec Google Chrome (ce n'était pas le cas avec Firefox Nightly qui avait une latence plus élevée que Google Chrome au moment des mesures[155], voir section suivante). La communauté MAO estime généralement que l'on peut sentir la latence quand elle est supérieure à 10ms, et que les guitaristes expérimentés pourraient même la sentir au-dessus de 5ms. Presonus, fabriquant reconnu de cartes sons, d'un autre côté, déclare[156] « *qu'une latence de 9.7 ms est encore en dessous du domaine de la perception humaine, et cela ne devrait pas*

[154] https://wasabi.i3s.unice.fr/AmpSim3/latency.html
[155] Nous avons utilisé Google Chrome version 59.0.3053.3 dev (64-bit) et Firefox Nightly version 55.0a1 (2017-04-01) (64-bit) pour Mac OS.
[156] https://www.presonus.com/community/Learn/The-Truth-About-Digital-Audio-Latency

affecter la performance ». Voir le livre « Audio Anecdotes » [193] pour une démystification de ce qu'est *la latence audio perçue*.

Avec le Yamaha THR10, un amplificateur de guitare hardware avec modélisation numérique, utilisant un circuit DSP dédié, nous avons mesuré une latence < 1ms.

La pédale Mod Duo utilise un système d'exploitation Linux et supporte les plugins audio natifs (écrits en C++) au format LV2 pour définir des effets et simulations d'amplificateurs. Ce matériel fonctionne sur un processeur double cœur ARM A7 1.0GHz, et inclut une carte son Cirrus Logic dédiée. La Mod Duo utilise une mémoire tampon audio fixe de 128 échantillons à 48Khz. La latence sans plugins, donnée par le constructeur, est de 9ms. Nous l'avons utilisé avec le plugin natif LV2 Guitarix JCM800 et un simulateur de baffles adapté.

Tableau 6. Latences guitare-haut-parleur mesurées

	Guitar Rig JCM800	**Garage Band JCM 800**	**Guitarix JCM800**	**Web Audio Amp (Chrome/FF Nightly 03/2017)**
Presonus 44VSL	17ms	19ms		23/32ms
Apogee Jam	19ms	26ms		26/35ms
ModDuo Pedal (Linux)			21ms	

On voit que les latences « réelles », mesurés de bout en bout (du son émis par le corps de la guitare jusqu'à la sortie des haut-parleurs) est loin de la valeur de 10ms cité souvent en référence comme le seuil maximal au-delà duquel la latence est gênante pour un jeu en temps réel. Le logiciel leader du marché comme Guitar Rig de Native Instrument donne une latence entre 17 et 19ms, et les simulations WebAudio, avec 23 et 26ms sont très proches. A noter que ces latences sont obtenues avec un buffer audio de 512 échantillons. Il sera sans doute possible de régler plus finement la taille de ce buffer dans les prochaines versions de l'API WebAudio, ce qui permettra de réduire encore plus la latence. Avec les valeurs actuelles, nous allons le voir, la latence n'a pas gêné les guitaristes (y compris professionnels) qui ont testé nos simulations (section 10.6). Ceci confirme bien des résultats récents sur la perception psycho-acoustique de la latence. Nous renvoyons à ce sujet le lecteur vers un article de référence par Vroomen et al. [223].

10.4.2 Réponses en fréquence et formes d'onde

Nous avons utilisé des ondes sinusoïdales à différentes fréquences et amplitudes comme signal d'entrée, et avons tracé les réponses en fréquence et la forme d'onde à différents stades de l'amplificateur WebAudio. Nous avons ensuite comparé avec la sortie d'un vrai amplificateur JCM800 hardware et avec la sortie du plugin natif Guitar Rig JCM800, avec des réglages similaires et la même sinusoïde en entrée.

Les réponses en fréquence étaient proches, mais la fonction de transfert asymétrique utilisée dans la simulation de la première lampe du préamplificateur (section 10.3.1), a introduit plus d'harmoniques que l'amplificateur réel et que Guitar Rig avec des réglages

pour un son « crunch »[157]. La Figure 164 montre la réponse en fréquence d'un signal d'entrée et du signal après l'étage de préampli (avec les deux fonctions de transfert de la Figure 145). En blanc, nous voyons les harmoniques paires et les pics entre eux sont les harmoniques impaires. Cela correspond à un son chaud, riche en harmoniques. Nous avons ensuite ajusté principalement les réglages du préamplificateur pour que les résultats soient les plus proches possibles de ceux que l'on obtient avec le véritable JCM800 (Figure 165).

A noter que ces mesures ne rendent pas du tout compte de l'aspect « dynamique » de l'amplificateur, c'est-à-dire la manière dont il répond à une attaque sur les cordes de la guitare. Elles montrent juste comment, pour un signal identique en entrée, la simulation est capable de générer le même type d'harmoniques que l'amplificateur réel. La signature « dynamique » d'un amplificateur dépend beaucoup de la boucle de rétro action négative de l'amplificateur de puissance, décrite dans la section 10.3.4.

[157] A noter que le plugin Guitar Rig produisait lui aussi des réponses en fréquences différentes du véritable amplificateur.

Figure 164. Avec la simulation : réponses en fréquence et onde Sin à 440Hz. A l'entrée et après la phase de préampli.

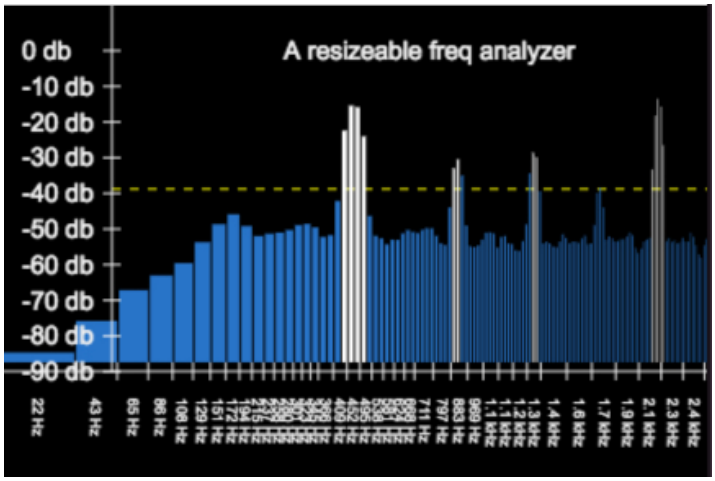


Figure 165. Réponse mesurée sur le vrai JCM800 en envoyant en entrée le signal sinusoïdal de la figure précédente. Réglages identiques sur la simulation et sur le vrai amplificateur. Le son en sortie a été mesuré avec un micro Shure SM58 placé devant le baffle du Marshall et re-routé dans l'ordinateur vers le même analyseur de fréquences.

10.5 Quelques mots sur l'interface graphique et sur le support MIDI

La version actuelle de l'interface graphique de l'ampli s'appuie sur un ensemble de WebComponents appelés "webaudiocontrols". Ils proposent des boutons photo-réalistes, des curseurs, des leds, des interrupteurs/commutateurs et ressemblent à des widgets populaires dans le monde des plugins audio natifs. Les *spritesheets* (ensemble d'imagettes contenant toutes les positions possibles des potentiomètres, etc.) dont ils dépendent ont été créées à l'aide d'un générateur en ligne (WebKnobMan), une version Web de l'outil KnobMan populaires chez les développeurs de plugins natifs. Nous avons contribué à cet ensemble de widgets en ajoutant un menu contextuel « midi learn » à chacun d'entre eux,

permettant le contrôle de la GUI à partir de périphériques MIDI externes. Nous utilisons par exemple l'ampli WebAudio avec un pédalier MIDI Behringer FCB1010 (pour changer les presets ou pour commander des effets[158], Figure 167) et avec un *pad* de contrôle Novation Launcher (qui vient avec un ensemble de boutons rotatifs que nous avons mappés aux potentiomètres de notre interface graphique, Figure 166), mais l'ampli peut être contrôlé en utilisant n'importe quel autre dispositif MIDI.

Figure 166. Un contrôleur MIDI permet facilement de modifier les réglages des instruments et effets WebAudio virtuels, sans quitter son instrument.

Figure 167. Pédalier midi utilisable avec les instruments et effets WebAudio.

[158] Vidéos :https://photos.app.goo.gl/HQfGATF6SgewJ3Ra9 et https://photos.app.goo.gl/XwNfbDa4iV62hbvZ9

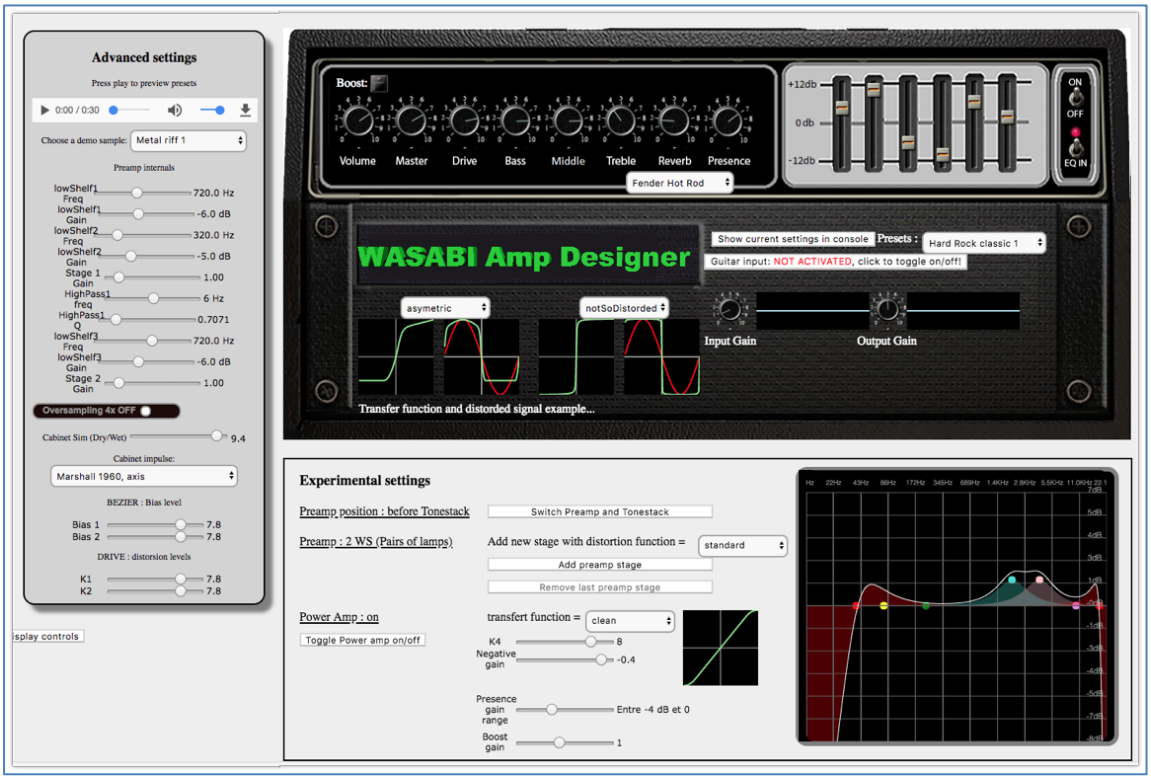


Figure 168. Interface développeur, permet de configurer dans le détail chaque étage de l'amplificateur simulé.

Nous avons conçu plusieurs interfaces graphiques, une pour les « concepteurs d'amplificateurs » (Figure 168), et d'autres pour les utilisateurs finaux (les musiciens, voir Figure 170). A partir de l'interface développeur on peut configurer tous les détails de tous les étages de la simulation, puis générer un fichier JSON qui contient tous les paramètres de la configuration actuelle de l'amplificateur (Figure 169).

Ce fichier sert ensuite à configurer un « plugin d'amplificateur » à la GUI très simplifiée (juste les réglages que l'on retrouve sur un ampli classique), qui sera intégré dans une station de travail audionumérique.

```json
{
    "name":"Heavy metal 1",
    "boost":false,
    "LS1Freq":720,
    "LS1Gain":-6,
    "LS2Freq":320,
    "LS2Gain":-5,
    "gain1":1,
    "distoName1":"asymetric",
    "K1":"7.8",
    "HP1Freq":6,
    ...
}
```

Figure 169. Extrait du fichier JSON décrivant une configuration des étages de l'ampli simulé. Le fichier complet contient une centaine de paramètres.

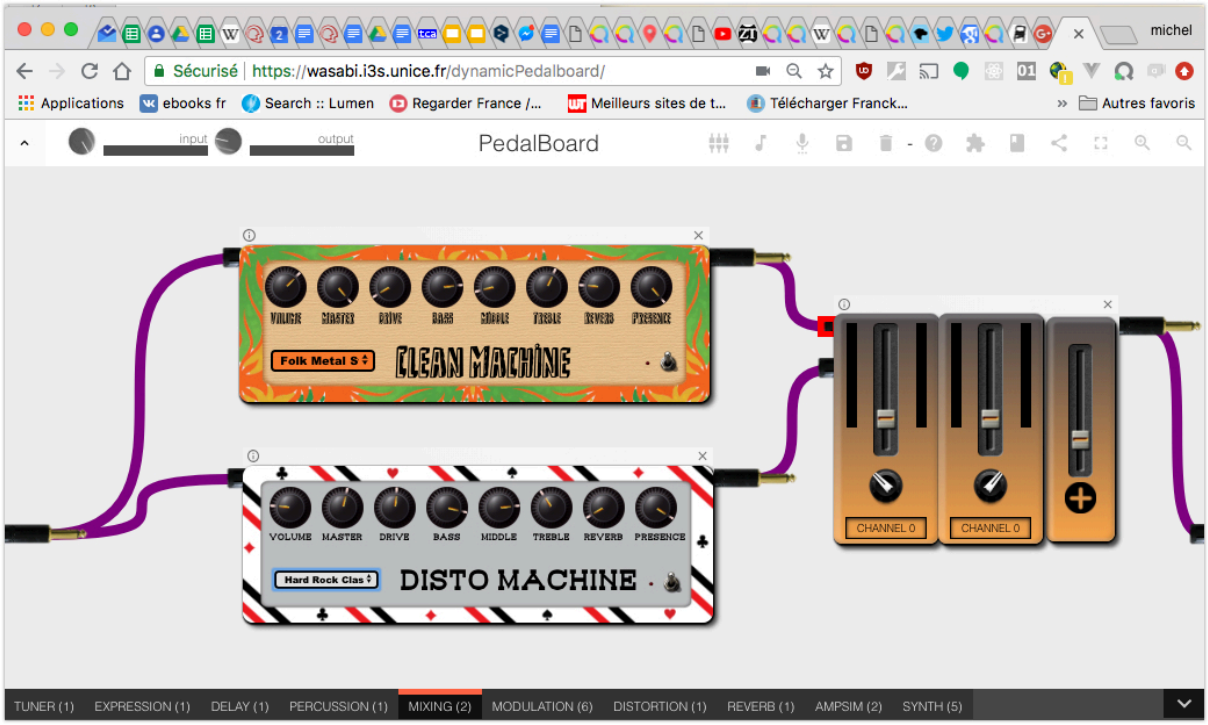


Figure 170. Ici deux amplificateurs créés à partir de fichiers de configuration. Le « Clean machine » possède un préamplificateur multi-bandes et produit des sons très clairs. Le « Disto Machine » est une réplique du JCM800.

10.6 Tests utilisateurs (*blind tests* et interviews)

Nous avons procédé à deux séries d'évaluation au cours du développement de cet amplificateur :

1. Une série avec une première version de l'amplificateur (version simplifiée de l'amplificateur de puissance), dont les résultats détaillés ont été publiés dans [37] et qui ont montré pour la première fois qu'il était possible d'avoir une simulation temps réel d'un amplificateur de guitare en WebAudio qui pouvait se comparer avec les meilleures simulations natives. La simulation WebAudio est arrivée globalement seconde derrière Guitar Rig, une référence dans le monde de la MAO, un logiciel commercial, et a été préférée à plusieurs autres simulations natives.
2. Une seconde série avec la dernière version de la simulation, celle décrite dans ce chapitre et détaillée dans cet article [199], qui a montré que les apports récents ont globalement amélioré la qualité de la simulation, notamment en ce qui concerne la dynamique générale de l'amplificateur et la palette de sons qu'il est possible d'obtenir. Cet amplificateur a notamment été testé par des professeurs de guitare (du conservatoire de musique de Gap, de l'école de musique de Biot) et par des guitaristes amateurs et professionnels de studio.

Pour la première session de tests, nous avons demandé à nos testeurs guitaristes d'horizons différents de jouer sur plusieurs simulations (Guitar Rig de Native Instruments - une référence utilisée par de nombreux studios d'enregistrement et musiciens, le plugin

GarageBand JCM800, Guitarix JCM800 - une simulation d'ampli open source - essayé sur un appareil Linux dédié : la pédale Mod Duo, et notre ampli WebAudio), ainsi que sur l'amplificateur hardware Yamaha THR10, qui utilise également la modélisation numérique.

Nous avons jumelé les simulations d'amplificateurs natives avec une simulation de baffles adaptées, et avec une réverbération si nécessaire, pour avoir des configurations similaires (ampli, réverbération, simulation de baffles) qui pourraient être comparées.

Nous avons utilisé une guitare Blade Austin (équipée de deux micros simple bobinage et d'un micro double bobinage actif), un Mac Book pro 2016 Mac OS Sierra, la carte son Presonus 44VSL et une paire d'enceintes de studio Tapco S5 (voir Figure 171). Nous avons utilisé Google Chrome et Firefox Nightly[159]. C'est la configuration matérielle qui propose la latence la plus faible. Nous avons laissé nos testeurs changer les réglages des micros sur la guitare, et nous avons modifié à la demande les réglages des simulations d'amplis ("ajoutez des aigus, ajoutez de la réverbération, donnez-moi plus de crunch..."). L'expérience de nos testeurs est > 10 ans de jeu de guitare, 3 d'entre eux ont joué dans des groupes pendant des années, ont donné des concerts, sont habitués aux amplis de guitare, 1 d'entre eux joue beaucoup avec des simulateurs d'amplis à la maison.

[159] Chrome version 59.0.3053.3 dev (64-bit) et Firefox Nightly version 55.0a1 (2017-04-01) (64-bit) pour Mac OS.

Figure 171. Configuration typique pendant l'évaluation qualitative

Nous leur avons demandé de classer les différentes simulations en utilisant cette échelle E (mauvais), D (moyen), C (bon), B (très bon) et A (excellent).

Vous pouvez trouver en ligne les vidéos de ces sessions[160]. Un rapport complet figure dans [37].

La simulation WebAudio a été ressentie « très proche de Guitar Rig », une référence dans le monde des plugins audio, et c'est très encourageant. Le plugin LV2 Guitarix JCM800 fonctionnant sur la pédale Mod Duo n'a sonné correctement pour aucun de nos testeurs.

Notre amplificateur matériel de référence, le Yamaha THR10, a obtenu un A dans chaque catégorie, avec tous nos testeurs. Ses haut-parleurs ne sont pas "neutres" comme les haut-parleurs de studio que nous avons utilisés avec les simulations, et ils ajoutent certainement beaucoup à la couleur du son. On peut conclure que les simulations que nous avons utilisées lors des évaluations ne sont pas encore comparables à du matériel dédié (latence, qualité du son).

Pour la seconde session de tests, nous avons surtout comparé la nouvelle version de notre simulation, qui implémente l'étage d'amplification de puissance, avec la précédente version qui n'implémentait pas cet étage. Nous avons notamment demandé à nos testeurs d'activer de de désactiver le Power Amp, pour voir les différences de ressenti. Nous leur

[160] https://wasabi.i3s.unice.fr/AmpSim3/userEvaluation.html

avons également demandé si les réglages de Master Volume et Présence répondaient mieux et si la palette des sons était plus intéressante.

Les vidéos de ces sessions de tests sont en ligne[161].

10.7 Perspectives

Cette simulation fait partie du projet ANR WASABI et fait partie d'un ensemble d'applications WebAudio incluant un pédalier virtuel d'effets audio pour guitaristes (présenté dans le chapitre suivant), une table de mixage, des outils d'analyse sonore, et ainsi de suite, liés et préréglés pour les sons des instruments utilisés dans les chansons de la base de données.

Les simulations d'amplificateurs que nous avons conçues peuvent être encore améliorées avec un modèle de tonestack plus fidèle, un meilleur modèle pour la partie amplificateur de puissance (nous n'avons pas encore modélisé les effets non linéaires de l'impédance réelle d'un haut-parleur), par ailleurs d'autres types de simulations des triodes pourront être testés et comparés (à base d'équation différentielles). Des simulations d'autres modèles d'amplificateurs sont prévues et seront réalisées au cours du projet.

10.8 Conclusion

Nous n'avons pas pu trouver de travaux antérieurs qui reproduisaient chaque étape d'un amplificateur de guitare à lampes dans un navigateur Web. Nous avons montré qu'il est possible de créer une simulation temps réel d'un véritable amplificateur (un modèle très connu dans la musique populaire) en se basant sur l'API émergeante WebAudio. La simulation proposée est à la fois jouable "guitare en mains", et compétitive en termes de son et de confort de jeu avec des plugins audio natifs commerciaux de référence.

Les modèles proposés jouent avec les limitations propres à la plateforme Web mais se sont avérés particulièrement performants, en attestent les mesures quantitatives et les tests qualitatifs effectués.

Cependant, le problème principal qui demeure avec WebAudio aujourd'hui, illustré par nos expériences, est toujours le manque de support des pilotes audio à faible latence par les navigateurs standard sous Windows, l'absence de moyen de régler la taille de la mémoire tampon audio ou de régler la fréquence d'échantillonnage pour ajuster la latence, et enfin, le manque de support pour les entrées/sorties audio multiples.

Le framework Jack Audio est certainement la voie à suivre car il peut agir comme un " proxy/barrière " entre les fournisseurs de navigateurs et les politiques de licence des éditeurs de pilotes audio (ASIO par exemple), mais il semble que son support par la communauté des logiciels libres soit en perte de vitesse depuis quelques années.

Andrew Mac Pherson, de la société Soundtrap.io (une station audionumérique sur le Web, sorte de GarageBand en line) a proposé des solutions pour WebKit en utilisant un autre

[161] http://wasabi.i3s.unice.fr/AmpDesigner/focusGroup.html

driver Windows présent dans les dernières versions de Windows 10, ses contributions n'ont toujours pas été acceptée pour intégrer les versions officielles des navigateurs[162].

WebAudio est encore une API très jeune, et l'intégration d'application de MAO dans les navigateurs récente. La communauté des chercheurs et développeurs, en s'attaquant à des problèmes difficiles comme celui présenté dans ce chapitre, permet d'identifier les points durs (notamment dans les implémentations et les contraintes inhérentes à la nature des navigateur web classiques), et motive le groupe de travail du W3C pour ajuster les APIs au mieux, les implémenteurs disposent de cas d'utilisations qui repoussent les limites ou identifient des problèmes sous-jacents importants (performances, sécurité, etc.). Au final, toute la communauté bénéficiera du travail de que nous avons mené en associant objectifs de recherche et amélioration des standards du Web.

[162] https://bugs.chromium.org/p/chromium/issues/detail?id=316908

11. Proposition pour un Standard ouvert pour Plugins WebAudio et applications (2018-)

Contexte

Le standard WebAudio comprend un ensemble de nœuds appelés AudioNodes, qui correspondent à certaines unités de traitement numérique du signal, par exemple, on trouve un nœud pour régler des gains (volumes), des nœuds pour faire des filtres biquad (supprimer ou diminuer certaines plages de fréquences), des nœuds pour splitter un signal mono en deux signaux stéréo etc. On assemble ces nœuds dans ce qu'on appelle un "graphe audio": le signal entre par un nœud d'entrée (il peut être un fichier audio ou vidéo, ou le micro, ou l'entrée d'une carte son, ou encore, certains nœuds peuvent générer des ondes sinusoïdales, triangulaires, etc.), traverse des nœuds de traitement, puis atteint un nœud de sortie, en général correspondant à la sortie de la carte son. En construisant des graphes WebAudio ont peut fabriquer des instruments (synthétiseurs ou lecteurs d'échantillons comme un piano), ou des effets (le simulateur d'amplificateur de la section précédente).

On trouve aujourd'hui de nombreuses applications différentes créées avec WebAudio qui fonctionnent indépendamment, mais il n'existe aucun moyen standard pour les rendre interopérables. Par exemple, il n'est pas possible de prendre une boîte à rythmes développée par X, la charger dans une application développée par Y et appliquer un traitement audio développé par Z. Dans le monde audio natif, ces unités interchangeables sont appelées des *plug-ins audio* et les applications qui peuvent les utiliser sont connues sous le nom *d'hôtes* qui sont typiquement des stations de travail audionumériques comme GarageBand ou Logic d'Apple, Cubase, Ableton Live, Pro Tools, etc.

Contributions

J'ai proposé, avec d'autres chercheurs provenant de différents équipes/pays, un standard ouvert pour des "Plug-in Web Audio" que nous avons appelé WAP (Web Audio Plugin) ainsi que l'infrastructure entourant une telle norme (SDK, protocoles, tutoriaux, preuve de concepts, exemples d'applications). Notre proposition est originale en ce qu'elle vise à regrouper plusieurs approches permettant de coder ces plug-ins en JavaScript, mais aussi en C/C++ (via le standard émergent WebAssembly) ou en utilisant des DSL (Domain Specific Languages).

L'un des groupes impliqués a créé FAUST, un langage spécifique au domaine pour la programmation d'algorithmes DSP (Digital Sound Processing), et nous avons travaillé avec ces chercheurs pour supporter WebAudio et compiler directement du code FAUST vers le standard WAP proposé [203][197]. Un autre groupe a créé les WebAudio Modules (WAMs) - une API pour développer des plug-ins basés sur le Web en utilisant C++ et

WebAssembly [38] et qui s'est rendu compatible avec le standard WAP proposé. Enfin, avec mon équipe, nous avons créé plusieurs applications WebAudio, en particulier un hôte virtuel capable de télécharger et d'assembler dynamiquement des plug-ins instruments ou effets comme par exemple des synthétiseurs, des pédales d'effet audio pour guitaristes et les simulateurs d'amplificateurs à lampes présentés dans la section précédente [41]. Nous travaillons actuellement à la finalisation d'une première version stable du standard WAP, et sommes en contact avec le groupe de travail WebAudio pour discuter ce cette contribution.

Nous avons proposé et animé le « WebAudio Plugin Workshop » qui s'est tenu lors de la dernière WebAudio Conference qui a eu lieu à Berlin en Octobre 2018.

Collaboration, encadrement

Je travaille depuis un an sur ce sujet avec Stéphane Letz, co-auteur du langage FAUST et chercheur au laboratoire GRAME (Lyon), avec Oliver Larkin, de l'Université de Huddersfield en Angleterre, créateur renommé de logiciels audio natifs, et avec Jari Kleimola, de l'Université de Aalto en Finlande, aujourd'hui CTO de la société Amped Studio, qui vise à commercialiser des applications de MAO de qualité professionnelle, capables de fonctionner dans un navigateur Web. J'encadre deux ingénieurs dans l'équipe WIMMICS qui développent avec moi les logiciels WebAudio implémentant le standard que nous proposons. Je travaille aussi sur la thématique WebAudio avec Jérôme Lebrun, chercheur au CNRS, depuis 2017. Je fais partie du groupe de travail du W3C sur WebAudio et contribue à ce standard. La proposition du standard WAP pour des plug-ins WebAudio a mis en évidence certains besoins qui ont fait l'objet de propositions pour la version 2 de l'API WebAudio. Certaines entreprises commerciales ont montré de l'intérêt pour le standard WAP et ont déjà intégré son support dans des versions béta de leurs logiciels (soundtrap.io, Amped Studio). Une école de musique en ligne leader français du domaine collabore également avec notre groupe et va intégrer les outils présentés dans une version expérimentale de son offre en ligne (https://imusic-school.com).

Publications principales ayant résulté de ces travaux

Michel Buffa, Jerôme Lebrun, Jari Kleimola, Oliver Larkin, Stephane Letz. and Guillaume Pellerin. *"WAP: Ideas for a Web Audio Plug-in Standard*". 4th Web Audio Conference (WAC 2018), Berlin, Germany. (2018). 〈hal-01893660〉

Michel Buffa, Jerôme Lebrun. *WebAudio Virtual Tube Guitar Amps and Pedal Board Design.* WebAudio Conference 2018, Berlin. (2018). 〈hal-01893781〉

Michel Buffa, Jérôme Lebrun, Jari Kleimola, Oliver larkin, and Stéphane Letz. "Towards an open Web Audio plugin standard". In Companion Proceedings (Developer's track) of the The Web Conference 2018 (WWW '18), Mar 2018, Lyon, France. 〈hal-01721483〉

Michel Buffa, Jerôme Lebrun, Jari Kleimola, Oliver Larkin, Stephane Letz. *Programming virtual musical instruments and audio effects in the Web browser.* 2018, "The Art, Science, and Engineering of Programming", April 2018, Nice, France. 〈hal-01735478〉

Résumé de cette section

Plusieurs formats de plug-ins audio natifs sont aujourd'hui populaires, notamment le format VST (Virtual Studio Technology[163], créé en 1997) de Steinberg (créateurs de Cubase), le format Audio Units d'Apple (Logic Audio, GarageBand), le format AAX d'Avid (créateurs de ProTools) et le format LV2 issu de la communauté audio Linux. Bien que les APIs proposées par ces formats soient différentes, elles existent toutes pour obtenir plus ou moins la même chose : « *représenter un instrument ou un effet audio et permettre son chargement par une application hôte* ». Dans l'API WebAudio, beaucoup plus récente (2011), il n'existe pas de format standard pour des plug-ins audio de haut niveau. Avec l'émergence des logiciels audio basés sur le Web tels que les stations de travail audionumériques (*DAW ou Digital Audio Workstations*) développées par des sociétés comme SoundTrap (propriété de Spotify), BandLab ou encore AmpedStudio, il est souhaitable d'avoir une norme afin de rendre les instruments et les effets audio Web interopérables sous forme de plugins compatibles avec ces DAWs et plus généralement, avec des logiciels hôtes compatibles.

Il existe de nombreuses façons de développer des logiciels avec l'API WebAudio (en JavaScript pur, ou bien en compilant du C/C++ vers WebAssembly ou encore en utilisant des langages spécialisés pour la programmation d'algorithmes DSP qui compilent eux aussi vers WebAssembly). Une telle norme de plugins devrait être suffisamment flexible pour supporter ces différentes approches, y compris l'utilisation d'une variété de langages de programmation.

Les nouvelles fonctionnalités rendues possibles par la nature même de la plate-forme Web (par ex : les plugins peuvent être distants ou locaux et identifiés par des URIs) devraient être disponibles pour les plug-ins écrits de différentes manières. À cette fin, plusieurs groupes de chercheurs et développeurs se sont réunis pour rendre leurs logiciels audio compatibles, et le présent document présente le travail accompli jusqu'à présent pour la création d'un standard de plugins WebAudio appelé WAP (**W**eb**A**udio **P**lugins).

Cela inclut le développement d'une spécification d'API, un SDK, des outils de validation de plug-ins en ligne et une série d'exemples de plugins écrits en JavaScript mais aussi avec d'autres langages. Ces exemples servent de preuves de concept pour les développeurs et illustrent aussi la puissance de la plateforme Web : on peut découvrir des plug-ins à partir de repositories distants, les télécharger dynamiquement dans une webapp hôte et les instancier, les connecter ensemble etc. Le projet comporte des exemples de plugins et de logiciel hôtes très simples, mais aussi un logiciel plus ambitieux pour valider le standard WAP : un "pédalier" de guitare virtuelle qui découvre les plug-ins de plusieurs dépôts à distance, et permet au musicien de chaîner par exemple des plug-ins pédales d'effets audio virtuelles, synthétiseurs, simulateurs d'amplificateurs de guitare, boîtes à rythmes etc. et de les contrôler via MIDI en temps réel.

[163] https://fr.wikipedia.org/wiki/Virtual_Studio_Technology

Le lecteur pourra avoir une idée « multimédia » de ce travail en regardant des vidéos en ligne qui présentent les résultats de ce travail[164].

11.1 Introduction

L'API WebAudio comprend un ensemble d'unités élémentaires appelés *AudioNodes* pour implémenter des algorithmes DSP audio basés sur le concept d'un graphe de traitement. Le signal entre en provenance d'une source sonore (média ou instrument / micro branché sur l'entrée d'une carte son) par un nœud du graphe audio, ou est généré par un ou plusieurs nœuds, et traverse le graphe. Chaque nœud traversé affecte le signal d'entrée (par exemple, un nœud peut être un filtre, un réglage de gain/volume, etc.). L'API WebAudio propose un ensemble d'AudioNodes de haut niveau qui facilitent l'implémentation d'effets classiques et permettent de créer des instruments virtuels simplement.

Par rapport à l'offre multimédia que propose HTML5 avec les éléments `<audio>` et `<video>` ainsi que leurs APIs JavaScript associées, les *AudioNodes* standards permettent de développer une large gamme d'applications Web nécessitant des moteurs audio qui vont au-delà de la simple lecture d'un média. Par ailleurs, l'API WebAudio cohabite très bien avec les APIs audio et vidéo d'HTML5. Il est tout à fait possible, à l'aide d'un AudioNode particulier, fourni par l'API, d'utiliser un élément `<audio>` ou `<video>` comme source sonore, et ensuite traiter le son de ce média au travers d'un graphe WebAudio.

L'ajout récent d'un nœud particulier, *l'AudioWorkletNode* offre par ailleurs un moyen efficace d'implémenter un traitement personnalisé de très bas niveau, augmentant considérablement les possibilités de cette technologie. Par exemple, on peut compiler le code DSP d'un plug-in natif existant, écrit en C++, vers du WebAssembly, et le fichier WebAssembly généré pourra être exécuté dans un *l'AudioWorkletNode.* Ceci donne la possibilité de porter de très nombreux travaux existants, écrits dans des langages divers, vers le monde WebAudio.

Aujourd'hui, en 2018, il existe de nombreuses applications différentes créées avec l'API Web Audio qui s'exécutent indépendamment les unes des autres, mais il n'existe aucun moyen standard de les rendre interopérables, c'est-à-dire qu'on ne peut pas prendre une boîte à rythmes développée par X, la charger dans une application développée par Y et appliquer un traitement audio développé par Z. Dans le monde audio natif, ces unités interchangeables sont les "plug-in audio" et les applications qui peuvent les utiliser sont connues comme "hôtes", ce sont des stations de travail audionumériques en général (DAWs –Digital Audio Workstations).

[164] Application « pedalboard » avec illustration de plugins pour guitare : https://www.youtube.com/watch?v=pe8zg8O-BFs. Création d'un plugin avec le langage FAUST : https://www.youtube.com/watch?v=24jUS4zei8Q&, Professeur de guitare du conservatoire de Gap lors d'une évaluation des logiciels du projet WASABI : https://www.youtube.com/watch?v=lIfm9ZMtG-I

Les auteurs du travail présenté ici proviennent de différents groupes de recherche qui ont tous développé leurs propres solutions pour implémenter des plug-ins audio dans le navigateur et qui ont fait converger leurs visions.

Ce chapitre présente ces idées pour créer une norme unifiée appelée "Web Audio Plug-in" (WAP) et décrit également l'infrastructure logicielle envisagée entourant une telle norme (dépôts de plugins, publication de plugins, etc.). D'autres initiatives de chercheurs existent, comme par exemple le Web Audio API eXtension framework (WAAX) [202] et la proposition pour le standard de plugins JSAP [201].

Notre proposition diffère de ces travaux en ce qu'elle vise à regrouper plusieurs approches déjà utilisées par nos groupes, permettant de développer des plug-ins Web Audio :

1. En JavaScript en utilisant des *AudioNodes* de haut niveau,
2. En JavaScript via les *AudioWorkletNodes*, pour coder des algorithmes DSP de bas niveau,
3. En C++ (via Emscripten/WebAssembly), ou…
4. En utilisant des Domain Specific Languages (DSL).

Nous souhaitons, avec notre proposition, pouvoir prendre en charge toutes ces approches à l'aide d'un standard de plug-in WebAudio unificateur. Ce standard doit être flexible et tenir compte des possibilités futures, telles que l'utilisation dans des applications Web progressives (PWA[165]) ou dans des environnements natifs, par exemple en embarquant un navigateur Web dans un plugin natif, et en exécutant une webapp hôte pour des WebAudio plug-ins dans ce browser. On peut ainsi exécuter un plug-in « universel » écrit à l'aide des technologies Web, dans une DAW native comme Cubase, Logic Audio, etc…

L'un des groupes impliqués dans la création du standard WAP a créé FAUST, un DSL pour le codage d'algorithmes DSP audio, et qui prend en charge la compilation vers Web Audio [197][203]. Un autre groupe a créé le standard WebAudioModules (WAM) - une API pour le développement de plug-ins basés sur le Web utilisant C++ et WebAssembly [38], enfin, avec mon équipe, j'ai créé une variété d'applications audio Web en JavaScript, y compris un hôte de plug-in comprenant des plugins de traitement audio sous la forme de pédales d'effet pour guitaristes, de simulateur d'amplificateur de guitare à lampes ou d'instruments virtuels (synthétiseurs, boîte à rythmes etc.) [199].

Au moment de la rédaction de ce manuscrit, il existe très peu de produits commerciaux basés sur l'API WebAudio, comparativement à la vaste gamme de logiciels disponibles sur le marché audio natifs, établi de plus de 30 ans. Il s'agit, par exemple, de trois DAWS[166], d'une application de test pour déficients auditifs et de deux outils en ligne visant les écoles de musique traditionnelles[167], qui proposent d'éditer et jouer des

[165] Les Applications Web Progressives permettent aux utilisateurs de vivre des expériences de type natif tout en bénéficiant des avantages du Web. En installant une PWA, les utilisateurs « montrent un engagement » qui permet de contourner certaines contraintes des webapps (par exemple, une PWA peut accéder au stockage sur disque dur, etc.)

[166] SoundTrap.com, BandLab.com et AmpedStudio.com

[167] noteflight.com, ultimate-guitar.com

partitions de musique. Cela est susceptible de changer depuis l'introduction de *l'AudioWorkletNode*, qui facilitera sans doute le portage d'applications audio professionnelles existantes (voir l'initiative de la société Propellerheads qui à titre d'essai, propose son synthétiseur logiciel commercial Europa à la fois sous forme de plugins natifs mais aussi en version de démonstration utilisable directement sur le Web, illustré dans la Figure 172).

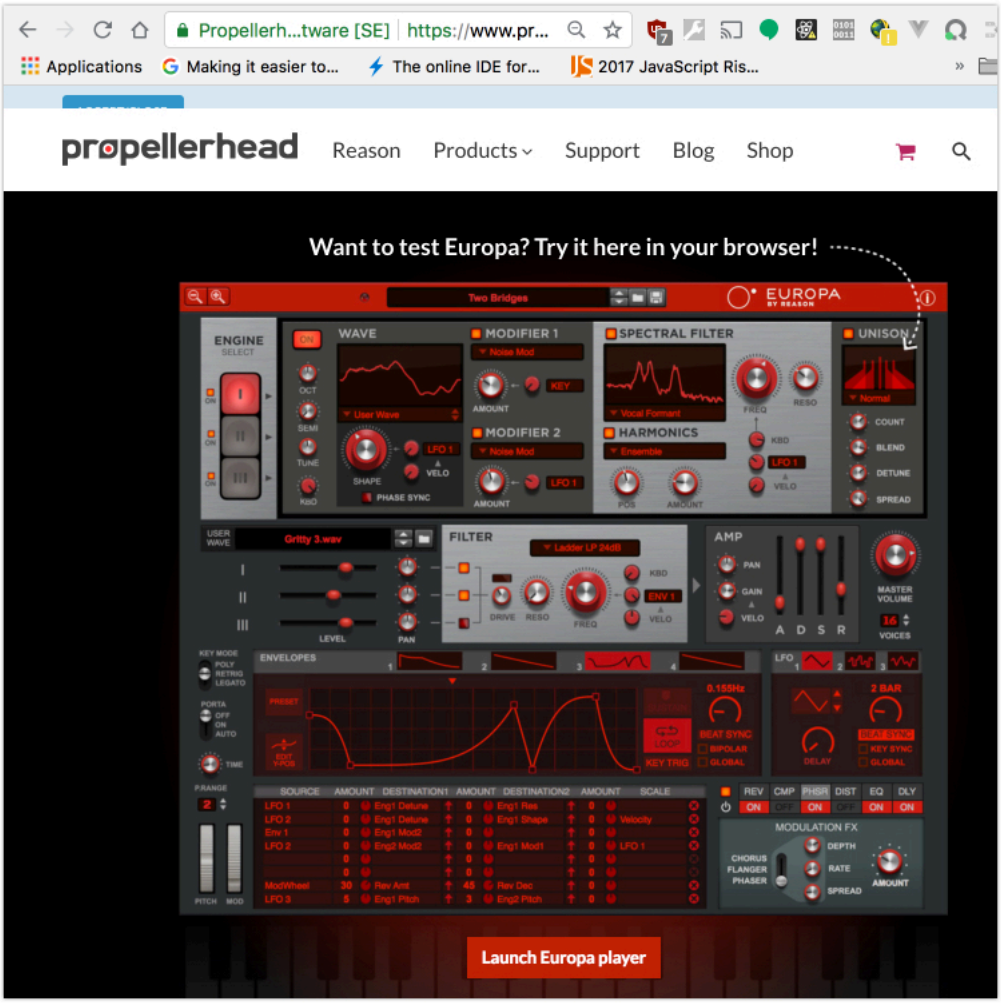


Figure 172. La société Propellerheads propose son synthétiseur logiciel commercial Europa à la fois sous forme de plugins natifs mais aussi en version de démonstration utilisable directement sur le Web.

11.2 Contexte

Dans [40] nous avons proposé un état de l'art des formats de plug-ins audio natifs existants, et précisé ce qui rendait la plate-forme Web différente. Par ailleurs, les groupes de chercheurs avec qui j'ai collaboré ont également présenté dans [38] et dans [39] des aperçus plus détaillés des caractéristiques définissant les différentes API de plug-ins natifs.

Pour la création du format WAP nous sommes également revenus plus loin dans le temps en examinant la proposition d'interface de plug-in audio généralisée intitulée GMPI[168] (pour *« Generalized Music Plugin Interface »),* qui date de 2003. Malgré son âge, cette proposition, réalisée par la « MIDI Manufacturers Association » (MMA) fournit un aperçu complet des qualités souhaitables dans une API de plug-ins audio natifs.

Le format de plug-ins LV2 par exemple, a été comparée au document GMPI dans une table qui est publié en ligne[169] et qui montre quels sont les caractéristiques recommandées par GMPI qui sont implémentées par le standard LV2. Nous avons décidé de faire une comparaison similaire[170] tout en faisant la spécification du format WAP. Cela nous a permis à la fois d'identifier les fonctionnalités les plus importantes qu'une API de plug-in devrait fournir, mais aussi d'éliminer les spécifications non pertinentes, et aussi réfléchir aux différences offertes par la plate-forme web.

Dans les sections suivantes, nous présentons les principales caractéristiques de notre proposition ainsi que son état actuel.

11.2.1 Le format WAP comparé aux recommandations de la proposition GMPI

La proposition GMPI énumère 114 exigences regroupées en 23 catégories. Nous avons dans un premier temps divisé ces catégories en trois groupes en fonction de leur pertinence par rapport aux exigences fondamentales d'un plug-in audio.

Nous avons décidé que la première ébauche de spécification du plug-in Web Audio devrait porter sur un premier groupe de catégories (les plus prioritaires) :

- **Modèle de fonctionnement plug-in / hôte** : nous devons définir comment les plug-ins sont chargés, instanciés et connectés ensemble dans le logiciel hôte. Les scénari d'hébergement nécessitent des mécanismes de découverte des plug-ins et de description de l'interface hôte.
- **Gestion des événements et support de la norme MIDI** : l'envoi et la réception de notes et d'événements de contrôle entre les plug-ins et l'hôte doivent être supportés, en particulier via le protocole MIDI qui est le protocole de communication le plus courant en MAO pour relier logiciels contrôleurs matériels. Le standard ne doit pas se restreindre à la norme MIDI mais devra pouvoir être compatible avec d'autres protocoles comme OSC.
- **Paramètres, Persistance** : les plug-ins devront exposer leur jeu de paramètres et fournir des getter/setters.
- **Gestion de l'état d'un plug-in** : plus généralement, un moyen de rendre l'état d'un plug-in persistant pour que le chargement et la sauvegarde de presets/banques de presets puissent être implémentés.

[168] https://tinyurl.com/k2wy5ge, cette proposition n'a rien donné de concret et le groupe qui l'a créé, le groupe de travail GMPI (Generalized Music Plugin Interface) de la MIDI Manufacturers Association (MMA) est aujourd'hui inactif.
[169] « LV2 achievement of GMPI requirements »: http://lv2plug.in/gmpi.html
[170] WebAudio plug-in vs LV2 vs GMPI: https://tinyurl.com/yd5fedrc

- **Interfaces utilisateur / GUIs** : bien que certains plug-ins puissent fonctionner en mode *headless*, sans GUI, nous devons permettre aux développeurs de réaliser des interfaces graphiques génériques ou personnalisées.

Le deuxième groupe de catégories (moins prioritaires mais pertinentes) sera ciblé dans une version ultérieure du standard WAP. Ces catégories sont : Services fournis par l'hôte, gestion avancée du temps, de la latence, système de protection contre la copie, localisation, etc.

Enfin, dans un troisième groupe on retrouve certaines catégories qui ne sont pas pertinentes pour les environnements Web ou qui sont déjà définies dans les API de niveau inférieur telles que WebAudio et WebRTC. Il s'agit notamment des catégories liées au *threading* en temps réel, à la fréquence d'échantillonnage, aux E/S audio, etc.

De plus, il faut tenir compte du fait que la proposition GMPI est ancienne (2003), et que plusieurs API de plug-ins natifs modernes ont été développés depuis [39]. L'environnement du navigateur Web signifie également qu'il y a des considérations supplémentaires, qui n'ont pas été prises en compte lors de la proposition GMPI, qui portait exclusivement sur des plug-ins natifs.

11.2.2 Les WAP ont des URIs !

Une norme de plug-in WebAudio doit être « web aware » et utiliser les URIs comme identificateurs pour les plug-ins et les référentiels, qui sont des citoyens/ressources Web de première classe.

Les applications Web hôtes doivent pouvoir découvrir des plug-ins distants en interrogeant des dépôts de plug-ins à travers leur API REST.

Les plug-ins doivent être utilisables sans avoir besoin d'une installation manuelle, et le mélange de différentes bibliothèques et frameworks JavaScript ne doit pas causer de conflit de noms ou de problèmes de dépendance.

Une idée forte de notre proposition est qu'un plugin WebAudio est à son logiciel hôte ce « qu'une image est à une page HTML » : on doit pouvoir simplement l'obtenir via son URI.

11.2.3 Convaincre les développeurs de rejoindre le standard WAP

Un standard de Plug-in Web Audio doit pouvoir prendre en charge plusieurs approches en termes de langage de programmation et d'environnement de programmation, y compris les langages JavaScript, C+++ (via WebAssembly) et les langages spécifiques au domaine (DSL). Nous l'avons déjà dit, la majorité des développements en MAO aujourd'hui sont en C++ ou utilisent des DSL comme FAUST, PureData, SuperCollider, CSound, etc., et il est important de convaincre les développeurs habitués à travailler avec ces environnements à franchir le pas et venir sur le Web. Pour cela, il est important de leur proposer des outils et des méthodes pour faciliter le portage de leurs travaux existants.

Par exemple, les auteurs de [38] ont développé l'API WAM (WebAudio Modules) qui facilite le portage de plug-ins natifs vers des WAPs, et cela a été démontré par le portage

de plusieurs « gros » plug-ins (des émulations de synthétiseurs commerciaux existant sous forme de plugins natifs écrits en C++, voir Figure 173. Le format WAM est spécialisé pour décrire des plugins cross-compilés en WebAssembly et s'exécutant dans un unique nœud *AudioWorkletNode*, est compatible avec le format WAP qui est plus « général » et vise à rassembler tous les cas possibles. Lorsque nous avons créé les premières versions de la proposition WAP, nous avons travaillé avec les auteurs du format WAM avec l'idée que WAM était juste un cas particulier.

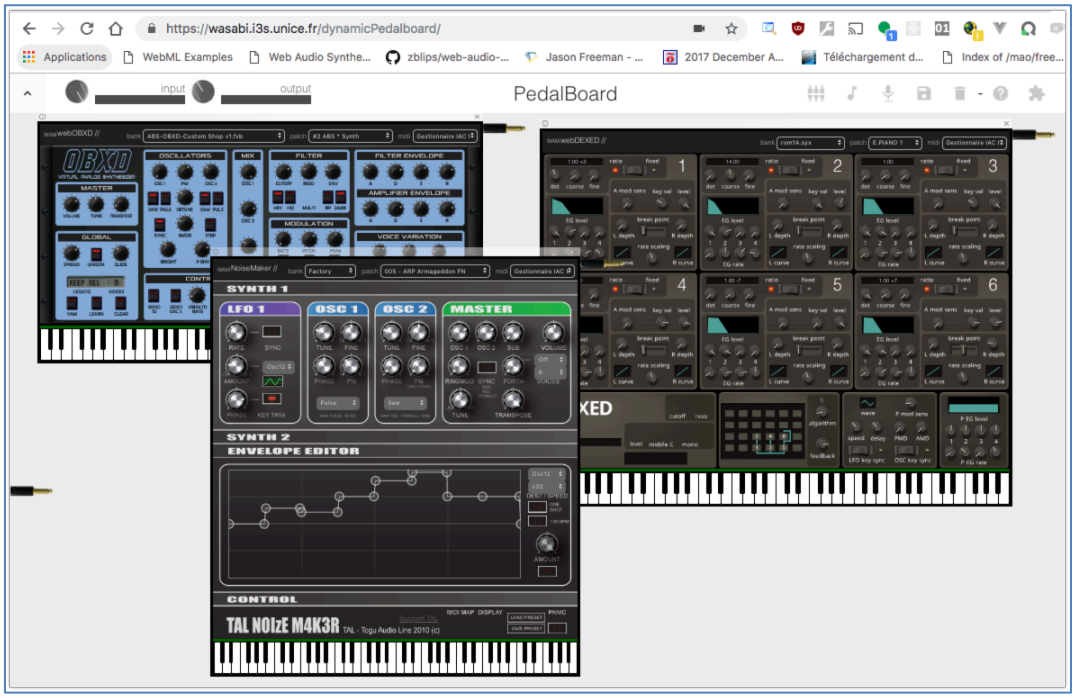


Figure 173. Synthétiseurs écrits originalement en C++ et compilés en WebAssembly puis mis au format WAM (WebAudio Modules), qui est un cas particulier de WAP (WebAudio Plugin). Ici dans l'application hôte que nous avons développée dans le cadre du projet WASABI.

La prise en charge du format WAM a également récemment été ajoutée au framework iPlug 2 (un framework de plug-in audio C++ capable de générer des plugins dans les différents formats existants : VST, AAX, LV2, etc.) [39], et pourrait également être ajoutée au framework JUCE[171], très populaire, pour permettre aux plug-ins natifs construits avec ces frameworks de fonctionner sur le Web.

Par ailleurs, avec les créateurs du langage FAUST nous avons développé un script pour compiler les fichiers « .dsp » de FAUST directement au format WAP. Nous espérons pouvoir prendre en charge un plus grand nombre de langages DSL à l'avenir, comme illustré par la Figure 174.

[171] Voir le site Juce.com, site officiel de ce standard.

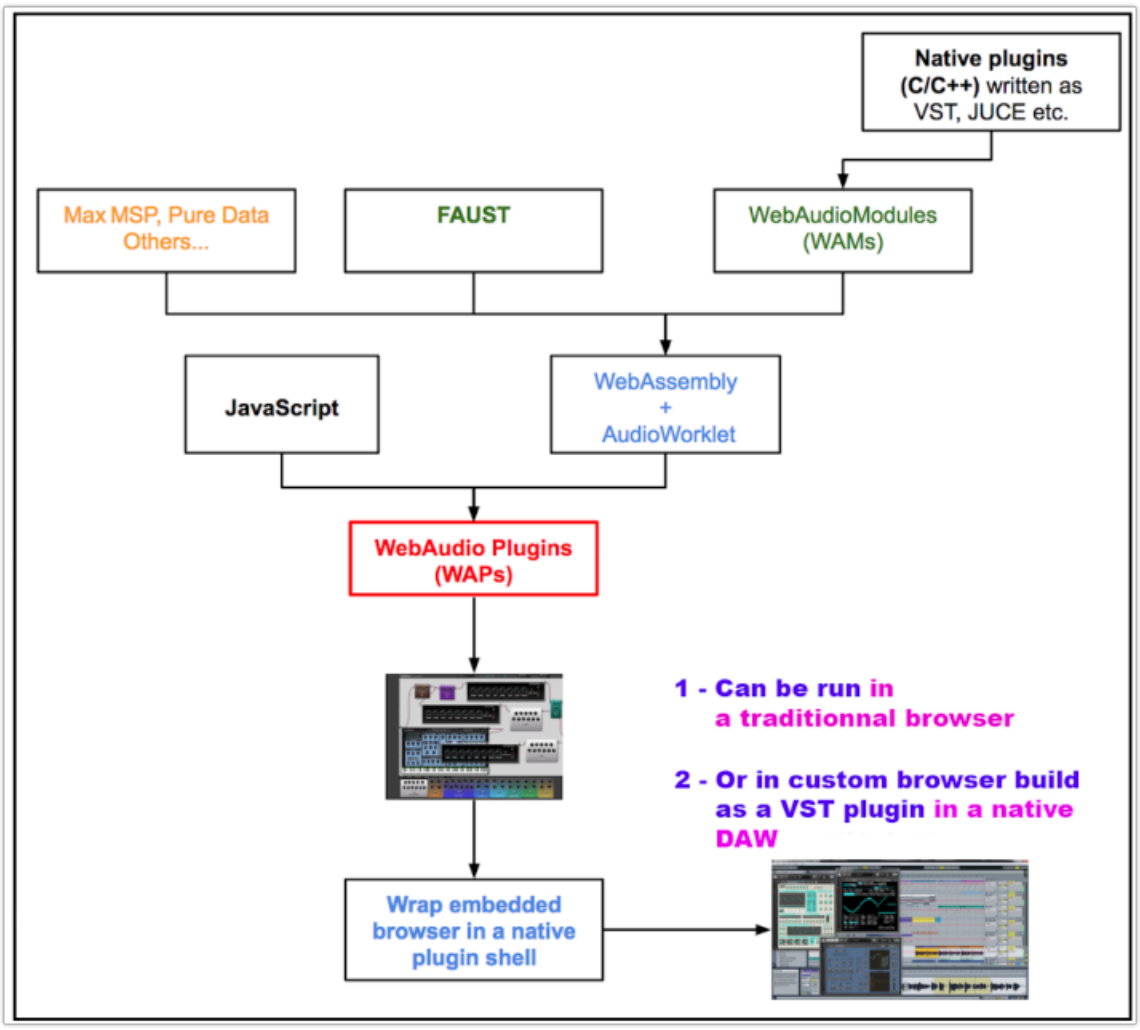


Figure 174. L'écosystème WAP.

Figure 175. Pédales d'effet audio écrites initialement en FAUST, ici sous forme de plug-ins WAP.

11.2.4 Prise en charge de plusieurs environnements d'exécution (Web et applications natives)

Bien que le Web soit la cible principale des WAPs, il existe des cas d'utilisation où l'intégration avec des applications natives peut être nécessaire. Par exemple, même si les

stations de travail audionumériques sur le Web (DAWS) offrent aujourd'hui des fonctions simples très attrayantes, les utilisateurs expérimentés de DAWS natives comme ProTools ou Logic Audio, qui demandent un long temps d'apprentissage et offrent de nombreuses possibilités supplémentaires, peuvent ne pas être facilement persuadés de passer immédiatement à un logiciel en ligne.

Dans ce cas, *la prise en charge des technologies Web dans les applications natives* pourrait contribuer à rendre les plug-ins WAPs exécutables dans des environnement connus des musiciens, producteurs, etc. La possibilité de tester des plug-ins natifs à l'intérieur d'un DAW familier avant d'effectuer un achat, mais sans installation, pourrait également être un cas d'utilisation intéressant. Les moteurs de jeu vidéo natifs et les frameworks de réalité virtuelle pourraient également bénéficier de l'API WebAudio qui prend en charge l'audio 3D. Les passerelles entre le Web et les domaines natifs sont explorées dans [204]. Une des solutions les plus prometteuses consiste à prendre un navigateur web sans GUI comme Chromium Embedded, de l'encapsuler dans du code C++ pour en faire un plug-in natif, et de charger ce plugin dans des DAWs natives comme on ferait pour un plug-in classique. On aurait donc un plugin/browser capable de s'exécuter dans des applications existantes natives, mais aussi capable de charger un logiciel hôte pour des WAPs, puisque ces derniers fonctionnent dans des pages Web. A partir de là on peut demander à cet hôte de charger et d'assembler des plug-ins WAPs. La Figure 176 montre une telle configuration : la station de travail audionumérique FL Studio, logiciel Desktop natif, peut utiliser des plugins WAP écrits en technologies Web. Sur la gauche de la figure on peut voir un plugin natif intitulé WebAudioVST qui embarque un navigateur Web qui fait tourner deux effets audio au format WAP (des pédales d'effets audio virtuelles). Sur la droite de la figure on voit qu'on a connecté ce plugin WebAudioVST avec un plugin natif Voxengo SPAN. Cela fonctionne sans perte de latence. Aujourd'hui un développeur doit compiler et packager son code C++ pour chacun des formats natifs existant. Il doit adapter les interfaces graphiques pour chaque plateforme visée (Mac OS, Windows, etc.). Ici, dans cet exemple, c'est intégralement le même code des pédales rouges et bleues qui va être exécuté quel que soit l'environnement d'exécution (page Web, DAW native, peu importe le système d'exploitation).

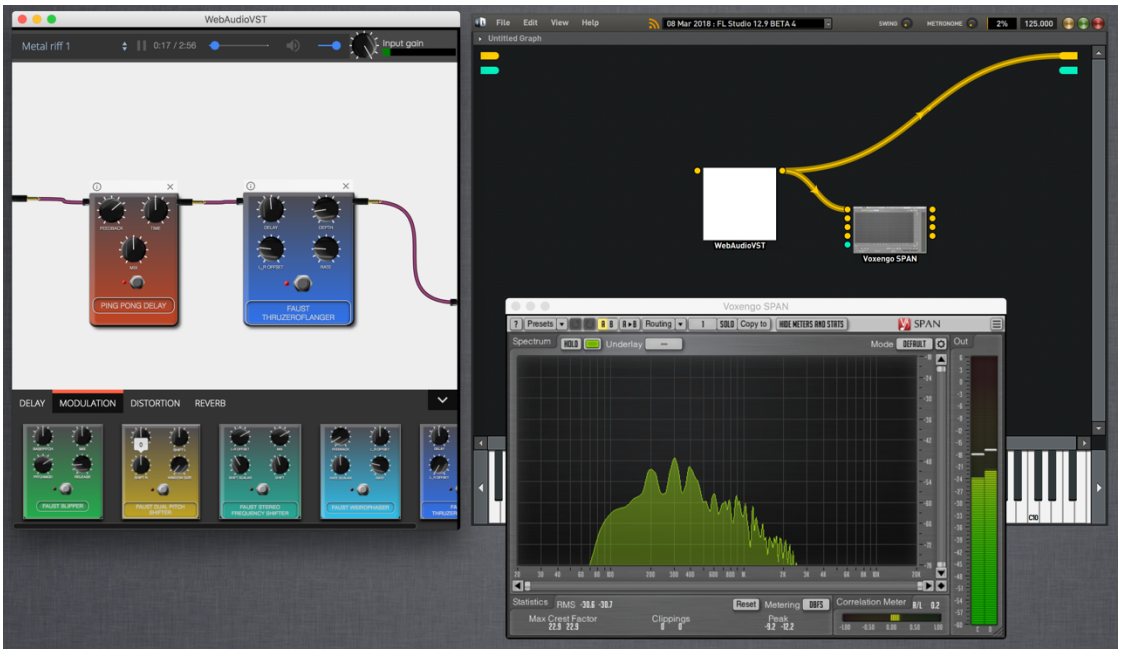


Figure 176. La station de travail audionumérique FL Studio, logiciel Desktop natif, peut utiliser des plugins WAP écrits en technologies Web. Cela fonctionne sans perte de latence notable (3ms).

La Figure 174 montre que le standard WAP a été pensé comme le standard pivot pour toutes sortes de sources (codes DSP écrits en JavaScript, DSL comme FAUST, C++, etc.) et d'environnements d'exécution (navigateur standard, ou navigateur intégré dans un plug-in natif, ou encore Progressive Web Apps). Les WAPs (en rouge sur la figure) sont au centre du monde Web et du monde natif.

11.3 Etat actuel de notre proposition

Notre proposition se compose d'une spécification (en état *draft*), d'outils en ligne et d'une série d'exemples et de tutoriaux. Notre objectif est, pour la version 1, de maintenir l'API WAP aussi simple que possible, tout en gardant la possibilité de s'intégrer avec les API Web existantes ou à venir comme les Progressive Web Apps, par exemple.

11.3.1 Une spécification

L'objectif de notre projet de proposition d'API de plug-in WebAudio [40, 205] est de concevoir un ensemble minimal de mécanismes permettant l'interopérabilité avec les standards existants. Une vue d'ensemble de haut niveau de la proposition est donnée ci-dessous.

- **Un WAP étend *AudioNode* (ou *AudioWorkletNode*) et hérite ainsi de ses propriétés et méthodes familières.** Ceci assure l'interopérabilité avec les nœuds standard proposés par l'API WebAudio et avec les applications construites autour de graphes WebAudio. L'intégration avec standard WebMidi est également assurée par les propriétés de type MIDIPort des *AudioNodes*.
- **Les WAP sont des nœuds audio *composites* ou *custom***. Les nœuds *composites* encapsulent un sous-graphe audio construit à partir d'un nombre quelconque

d'AudioNodes. Les nœuds *custom* sont des *AudioWorkletNodes*. Ainsi, qu'un plugin soit composé d'un seul nœud ou de plusieurs, on pourra le connecter/déconnecter simplement à l'aide des méthodes standards WebAudio connect/disconnect. La notion de « composite node » est un patron de conception qui permet de donner à un ensemble d'éléments la même interface que chacun des éléments. Ici on aura un plug-in composé de multiples AudioNodes qui reprendra les principaux aspects de l'interface *AudioNode.*

- **Les WAPs sont conscients de l'interface graphique mais agnostiques quant à leur stratégie de mise en œuvre.** Cela signifie que les WAPs peuvent être sans GUI (*headless*), ou bien ils peuvent exposer leur GUI sous la forme d'un unique élément HTML (par exemple, un div, un canvas, un canvas SVG, ou élément personnalisé) qui peut être attaché au DOM. La conception des WAPs garantit que le code de l'interface graphique ne sera chargé que si nécessaire.
- **Les WAPs sont chargés de manière asynchrone via leur URI.** Le standard utilise les promesses de JavaScript 2015 pour la gestion des aspects asynchrones liés au chargement des plugins.
- **Des métadonnées WAP décrivent les aspects spécifiques à la mise en œuvre du plug-in.** Ces métadonnées sont disponibles sous la forme d'un fichier JSON distinct. Elles décrivent la configuration des E/S audio et midi, les espaces de noms, les paramètres, le type de plug-in, les URIs, etc. Les dépôts WAP peuvent collecter ces fichiers JSON dans des agrégats à des fins de découverte.
- **Un serveur/référentiel WAP REST est décrit par une URI, qui peut pointer vers une ressource en ligne ou un système de fichiers local.** Les métadonnées peuvent décrire des URI distinctes pour les WAP GUI et les WAP avec GUI. Nous prévoyons deux stratégies d'intégration : une webapp hôte peut simplement utiliser l'une des URIs dans une balise script/link. Mais on doit pouvoir aussi instancier un WAP sans connaître le nom des classes qui le composent, simplement à partir d'une analyse des métadonnées. L'URI doit être suffisant.

11.3.2 Outils en ligne, didacticiels et exemples

En plus de la documentation en ligne, nous proposons des exemples simples et des didacticiels pour le "côté hôte" et le "côté plug-in" de notre proposition, ainsi que des outils en ligne tels que des validateurs/testeurs. Certaines sont présentées dans les différentes figures qui suivent. Chaque légende contient une note de bas de page avec le lien vers l'application web exécutable.

```
var ctx = new AudioContext();
var player = document.getElementById("soundSample");
var mediaSource = ctx.createMediaElementSource(player);
var intermediateGain = ctx.createGain();

var pluginURL = "https://wasabi.i3s.unice.fr/WebAudioPluginBank/WASABI/PingPongDelay3";
var plugin = new WasabiPingPongDelay(ctx, pluginURL);

plugin.load().then((node)=>{
  console.log("node",node);

  mediaSource.connect(node);
  node.connect(ctx.destination);
});
```

Figure 177. Chargement d'un plugin sans GUI. Ici on le charge et on le connecte entre un lecteur audio (ligne 3) et la destination (haut-parleur ou sortie de la carte son).

Hôte chargeant un plugin sans GUI : La Figure 177 montre des extraits d'une implémentation hôte minimale qui charge un plug-in sans GUI et le connecte au graphe WebAudio. En coulisses, dans le code du SDK fourni, un fichier de métadonnées JSON est chargé à partir de l'URI du plug-in. Une balise `<script src="..."></script>` HTML est ajoutée si nécessaire. Ensuite, le plug-in est initialisé. Comme il peut charger des ressources telles que des fichiers d'images ou un module WebAssembly (WASM) de manière asynchrone, la méthode de chargement renvoie une promesse JavaScript. Dans cet exemple, le nom de la classe du plug-in est codé en dur, mais il aurait pu être construit dynamiquement à partir du contenu du fichier JSON des métadonnées du plug-in (d'autres exemples montrent comment le faire, comme le testeur de plug-in en ligne de la Figure 178). Du point de vue de l'hôte, le plug-in peut être de n'importe quel type : un graphe Web Audio dans un nœud *CompositeNode* ou un seul nœud *CustomNode* (AudioWorklet), écrit en JavaScript ou dans WebAssembly, etc. L'implémentation du patron de conception *CompositeNode* est également fournie par le SDK.

Figure 178. Testeur de plug-in en ligne : on entre l'URI du plug-in pour tester et valider le plug-in avant de le publier dans un dépôt.

Hôte charge un plug-in avec sa GUI : La Figure 179 montre le même exemple que celui de la Figure 177, mais cette fois, nous chargeons aussi le code de l'interface graphique (HTML, CSS, JS) de manière asynchrone. La méthode `loadGUI` retourne un seul élément HTML qui contient l'interface graphique du plug-in. Là encore, la méthode est asynchrone et renvoie une promesse car un plug-in peut avoir à charger des ressources (images pour des boutons, fichiers d'échantillons audio, etc.).

Figure 179. Le même plug-in, avec une interface graphique.

Les implémentations des méthodes `load` et `loadGUI` sont héritées par défaut lorsque vous étendez la classe `WebAudioPluginFactory` du SDK, mais peuvent être

remplacées par le développeur. Dans nos exemples, nous utilisons des WebComponents pour regrouper les fichiers de l'interface graphique dans un seul fichier HTML, en bénéficiant de l'encapsulation proposées par les WebComponents et en évitant ainsi tout conflit de nom. En coulisses, la méthode `loadGUI` crée par défaut un lien `<link rel="import" href="main.html">` si nécessaire. Si le développeur préfère utiliser un canvas etc. pour l'interface graphique, il lui suffit de remplacer la méthode `loadGUI` par une implémentation de son choix.

Des exemples plus détaillés sont disponibles sur les pages de documentation de la proposition WAP. Certains montrent en particulier comment faire une vraie découverte dynamique, sans coder en dur les noms des classes des plugins dans le code de l'hôte.

Validateur de plug-ins en ligne : cet outil en ligne utilise ce comportement dynamique. Il est fourni aux développeurs de plug-ins pour tester leur travail (voir Figure 178). Copiez et collez l'URI d'un plug-in et le code du plug-in sera téléchargé, le plug-in testé, et si un ensemble minimal de tests est passé, le plug-in sera exécutable sur la page et son interface graphique affichée, etc. Il est alors possible de publier le plug-in sur un dépôt de plug-in en ligne. Notez que tous les tests ne sont pas obligatoires pour rendre le plug-in utilisable. Par exemple, si un plug-in n'implémente pas le chargement/sauvegarde de l'état de ses paramètres, il est toujours utilisable.

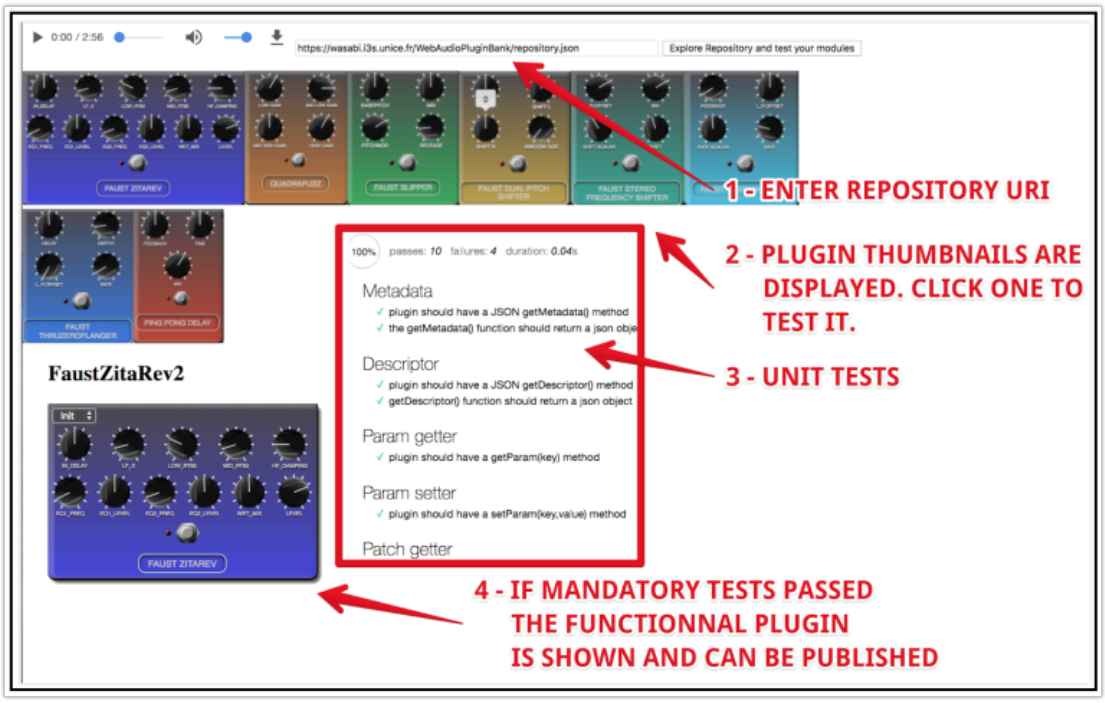


Figure 180. Testeur de dépôt de plug-ins en ligne

Validateur en ligne de dépôt de plug-in : La Figure 180 montre un testeur de dépôt de plug-ins distant : entrez l'URI du dépôt pour interroger son interface REST. Par défaut il renverra une description JSON avec la liste des plug-ins (avec leurs URIs associés) suivie du fichier de métadonnées de chaque plug-in. La vignette (*thumbnail*) de chaque plug-in est affichée sur la page et peut être cliquée pour tester le plug-in correspondant. Si les tests obligatoires ont réussi, vous pourrez essayer le plug-in en ligne et obtenir un rapport

complet des tests unitaires. Cet outil permet de tester si un dépôt de plug-ins implémente correctement l'API REST de découverte proposée par le standard WAP.

```
Promise.all([
        host.load("./../WASABI/QuadraFuzz/QuadraFuzz.html"),
        host.load("./../Faust/ZitaRev/FaustZitaRev.html"),
        host.load("./../Oliver-Larkin/ThruZeroFlanger/FaustThruZeroFlanger.html"),
        host.load("./../Oliver-Larkin/WeirdPhaser/FaustWeirdPhaser.html"),
        host.load("./../Oliver-Larkin/StereoFrequencyShifter/FaustStereoFreqShifter.html"),
        host.load("./../Oliver-Larkin/Blipper/FaustBlipper.html"),
        host.load("./../Oliver-Larkin/DualPitchShifter/FaustDualPitchShifter.html")
    ])
    .then(function (nodes) {
        nodes[0].connect(nodes[1]);
        nodes[1].connect(nodes[2]);
        nodes[2].connect(nodes[3]);
        nodes[3].connect(nodes[4]);
        nodes[4].connect(nodes[5]);
        nodes[5].connect(nodes[6]);
        nodes[6].connect(ctx.destination);
    });
```

Figure 181. Chargement et enchaînement de plusieurs plug-ins

Un utilitaire "loader" de plug-ins : dans certains cas, un développeur peut vouloir s'assurer que plusieurs plug-ins ont été chargés et sont prêts à fonctionner avant de les chaîner. Le SDK WAP fournit un objet utilitaire nommé "plug-in loader" qui peut être utilisé avec la méthode `Promise.all` de ES6, comme le montre la Figure 181.

Etapes minimales pour écrire un plug-in

Maintenant que nous avons étudié comment un hôte peut découvrir et utiliser un plug-in, il est plus facile d'illustrer les différentes étapes nécessaires à la création d'un plug-in. Le SDK fournit plusieurs classes qui peuvent être héritées et des classes utilitaires, ainsi que des exemples, un tutorial etc.[172]

Lors de l'écriture d'un plug-in, différentes parties doivent être prises en compte :

Partie 1 - Le descripteur de métadonnées : pour cette première version, nous avons voulu avoir un descripteur aussi simple que possible, pour faciliter l'adhésion des développeurs. Il contient un ensemble de paires clé/valeur dont certaines sont obligatoires et d'autres optionnelles.

[172] https://github.com/micbuffa/WebAudioPlugins

Obligatoire : nom du plug-in, fournisseur (« vendor » en anglais), version, catégorie, thumbnail (Figure 182). Le nom de la classe principale du plug-in et son URI relatif peuvent être déduits à partir du nom du fournisseur et du nom du plugin. Optionnel, le type utilisé pour décrire l'état actuel du plugin (la valeur courante de tous ses paramètres). Par défaut ce type est JSON, mais pour des instruments virtuels comme des synthétiseurs, qui peuvent avoir plusieurs centaines de paramètres, on utilise en général des blobs binaires.

Pour les prochaines versions une description RDF plus complète est envisagée, à l'instar de ce que propose le standard de plugins LV2.

```
{
  "documentation": "A classic Stereo Delay",
  "name": "PingPongDelay",
  "thumbnail": "WasabiPingPongDelay.png",
  "vendor": "Wasabi",
  "category":"Delay",
  "version": "1.0"
}
```

Figure 182. Descripteur typique d'un plugin WebAudio.

Partie 2 – La classe principale du plug-in, qui implémentera l'API du plug-in Web Audio. Elle doit étendre soit la classe `WebAudioCustomNode` (i.e. si le plug-in est un AudioWorklet) soit la classe `WebAudioPluginCompositeNode` (s'il est constitué d'un ensemble de nœuds Web Audio), fournie par le SDK. Nous avons opté pour une approche *« convention over configuration »* : seules des étapes minimales sont nécessaires pour avoir un plug-in exécutable car de nombreuses propriétés et méthodes, valeurs par défaut sont héritées. Par exemple, le fichier principal par défaut sera nommé `main.js` sauf si indiqué dans le fichier descripteur des métadonnées, les méthodes `getParam/setParam` seront héritées, etc. Les Figure 183 et Figure 184 montrent des extraits de code source des classes du SDK fournies pour la création d'un plug-in `CompositeNode`, la Figure 185 montre un squelette de ce à quoi ressemblerait la classe principale d'un plug-in, et enfin, la Figure 186 montre comment ce plug-in peut être utilisé comme un nœud Web Audio standard et inclus dans un graphe de traitement classique. Un exemple complet est disponible en ligne[173] (brancher le son !).

[173] https://jsbin.com/wadoqal/edit

```
//----- 1 - CompositeAudioNode ----
// has connect/disconnect methods
// A custom composite node can be derived from this prototype.
class CompositeAudioNode {
  get _isCompositeAudioNode () {
    return true;
  }

  constructor (context, options) {
    this.context = context;
    this._input = this.context.createGain();
    this._output = this.context.createGain();
  }

  connect () {
    this._output.connect.apply(this._output, arguments);
  }

  disconnect () {
    this._output.disconnect.apply(this._output, arguments);
  }
}
```

Figure 183. La classe `CompositeAudioNode` du SDK.

```
// ----------------------
// CREATE THE PLUGIN CLASS
// ----------------------
class WebAudioPluginCompositeNode extends CompositeAudioNode {
  constructor (context, options) {
    super(context, options);
    // ...
  }
  // P2 from WAP specification...
  set descriptor(descriptor) {
    this._descriptor = descriptor;
  }
  get descriptor() {
    return this._descriptor;
  }
  // ... other default properties and methods
  // that are indicated in the WAP spec
}
```

Figure 184. Classe du SDK qui implémente en partie l'API WAP avec des valeurs par défaut. Destinée à être sous-classée.

```
class WasabiStereoDelay extends WebAudioPluginCompositeNode {

  constructor(ctx, options) {
    super(ctx, options)

    this.addParam({name:'feedback', defaultValue: 0.5, minValue: 0, maxValue: 1 });
    this.addParam({name: 'time',defaultValue: 0.5, minValue: 0, maxValue: 1 });
    this.addParam({name: 'mix',defaultValue: 0.5, minValue: 0, maxValue: 1 });

    /*    ################     API PROPERTIES    ###############   */
    this.params = {
      "feedback": this._descriptor.feedback.defaultValue,
      "mix": this._descriptor.mix.defaultValue,
      "time": this._descriptor.time.defaultValue,
      "status": "disable"
    }

    this.buildAudioGraph();
  }
  /*    ################     API METHODS    ###############   */
  get numberOfInputs(){
    return this.inputs.length;
  }

  get numberOfOutputs(){
    return this.outputs.length;
  }
}
```

Figure 185. Un plugin WAP composite doit étendre la classe `WebAudioPluginCompositeNode`.

```
var context = new AudioContext();
var myDelayPluginCompNode = context.createWasabiStereoDelayCompositeNode();
var oscNode = context.createOscillator();

oscNode.connect(myDelayPluginCompNode).connect(context.destination);

oscNode.start();
oscNode.stop(1.0);
```

Figure 186. Finalement, un WAP composite peut être utilisé comme un nœud WebAudio standard.

Partie 3 - La classe "factory" du plug-in, responsable du chargement asynchrone du plugin et de son instanciation. C'est cette classe qui implémentera (ou héritera) les méthodes `load` et `loadGUI`.

La Figure 187 montre un exemple d'une telle classe.

```
var WAPlugin = WAPlugin || {};

WAPlugin.WasabiStereoDelay = class WasabiStereoDelayFactory extends WebAudioPluginFactory {
    constructor(context, baseURI) {
        super(baseURI)
    }

    // load and loadGUI methods are inherited but
    // can be overriden here...
}
```

Figure 187. Classe Factory associée au plug-in des figures précédentes. Elle hérite des méthodes `load` et `loadGUI`.

11.3.3 Pedalboard : un hôte plus complexe pour plugins WAP

En plus des "exemples de hosts minimalistes" inclus dans le SDK, une application web plus ambitieuse intitulée "pedalboard virtuel" a été développée [199].

Cette application en ligne[174] donne la possibilité de chaîner interactivement des plug-ins afin de créer des sons et configurations plus complexes (voir Figure 188). Elle gère la découverte de plug-ins à partir de plusieurs référentiels (locaux ou distants), les réglages de gain des cartes son en entrée et en sortie, les cycles de vie et les interconnexions des plug-ins ainsi que la sauvegarde et la restauration des états. Par ailleurs, l'application donne la possibilité de sauvegarder des configurations et de les organiser en banques de presets (Figure 189).

L'utilisateur peut créer des instances de plug-ins en glissant et déposant leurs imagettes/thumbnails dans la zone principale. Ils peuvent ensuite les positionner, les relier entre eux, etc. Enfin, à l'aide de l'interface graphique (boutons, curseurs, commutateurs), l'utilisateur peut ajuster chaque plug-in individuellement. En assemblant un simulateur d'amplificateur, une réverbération, un effet fuzz et un delay stéréo, par exemple, on peut créer un son psychédélique riche. Nous avons également contribué à la bibliothèque webaudiocontrols[175] en ajoutant le contrôle MIDI à tous les éléments de l'interface de widgets graphiques « photo réalistes » qu'elle propose (boutons, commutateurs, etc.). Ainsi, les plug-ins peuvent avoir leur interface graphique contrôlée à distance via n'importe quel contrôleur MIDI. Les Figure 188 et Figure 189 montrent des captures d'écran typiques de cette application hôte, combinant des plug-ins écrits en FAUST, en C++ (WAMs) et en JavaScript (simulateur d'amplificateur de guitare, pédales d'effets).

[174] https://wasabi.i3s.unice.fr/dynamicPedalboard/
[175] https://github.com/g200kg/webaudio-controls, une librairie de widgets photo réalistes pour développer des GUI WebAudio.

Figure 188. Plug-ins WAP assemblés à l'intérieur de l'application Web Pedalboard.

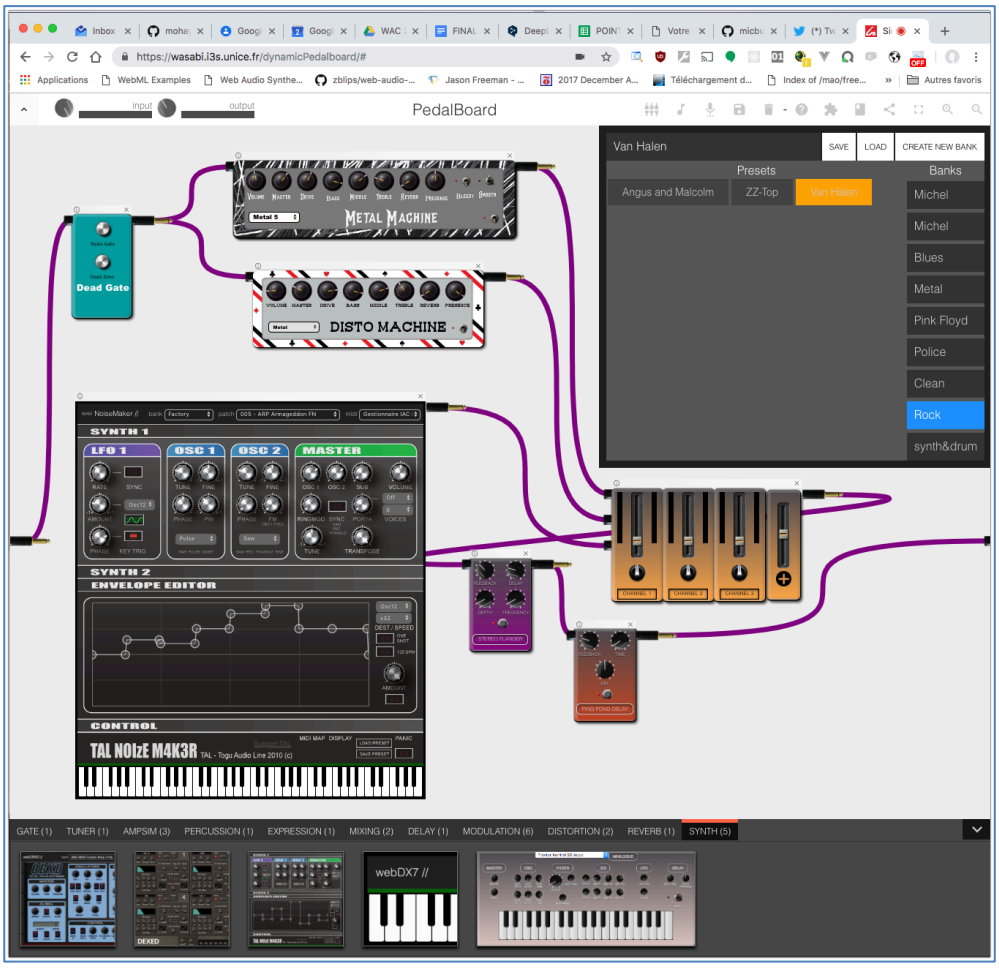


Figure 189. Gestion de banques de presets. L'état de chaque plug-in, ses paramètres, position, connexions sont sérialisés. Le développeur de plug-ins n'a rien à faire : les méthodes `getState()` et `setState()` sont héritées des classes du SDK.

11.4 Perspectives

Pour la première version de l'API WAP, nous avons isolé de la spécification GMPI un ensemble minimal de fonctionnalités à implémenter.

Les travaux futurs à court terme portent sur l'amélioration des fonctionnalités MIDI et sur la manière dont nous pouvons synchroniser le tempo entre plug-ins d'une manière similaire aux plug-ins natifs (ex : une boîte à rythmes et un synthétiseur qui joue des arpèges doivent être synchronisés sur le même tempo et sur les mêmes mesures).

Les bus d'entrée et de sortie multiples pour factoriser des traitements sont également à prendre en compte, ainsi que la gestion de l'automation des paramètres des plug-ins (évolution dans le temps de leurs valeurs, par interpolation).

Nous devrions également étudier la possibilité d'avoir la flexibilité nécessaire pour soutenir les capacités de "production musicale intelligente" offertes par JSAP [201] en intégrant de la sémantique dans la description des WAPs et de leurs interactions.

Nous prévoyons aussi d'évaluer plus en détail les problèmes de performance et d'occupation mémoire du graphe audio, nous avons déjà fait des retours au groupe de travail WebAudio du W3C dans ce sens-là, pour la version 2 de l'API WebAudio (possibilité de regrouper des nœuds par plug-ins, de faire de l'introspection sur le graphe audio, de disposer de fonctions d'APIs pour les outils de debug intégrés aux navigateurs etc.)

11.5 Conclusion : des plugins audio universels, fonctionnant même dans des logiciels natifs !

Nous avons présenté une proposition pour une norme ouverte de plug-in audio Web (WAP), qui consiste en une spécification et un SDK comprenant un tutoriel et des exemples écrits en JavaScript, en FAUST et en C++. Nous avons également réalisé des validateurs de plug-ins et des validateurs de référentiels en ligne. A partir d'aujourd'hui, FAUST, ainsi que l'API des WebAudioModules (WAM) sont compatibles avec les WAPs et nous espérons que d'autres langages/environnements seront bientôt supportés.

Nous avons développé une application hôte complète (le pedalboard) qui scanne les dépôts de plug-ins (locaux ou distants) et qui peut être utilisée pour assembler plusieurs plug-ins dans un graphe. Cette application a été évaluée avec des musiciens professionnels [199] et montre que le modèle hôte / plugin que nous proposons est adapté à des applications plus sophistiquées. Jusqu'à présent, nous avons intégré plus de 20 plug-ins, y compris des synthétiseurs et des effets qui ont été développés en utilisant différentes technologies mais qui sont tous conformes à notre API.

Le SDK WAP préliminaire est disponible en ligne[176] et de nombreuses vidéos de démonstration sont sur YouTube (chercher « webaudio plugins » et « webaudio guitar »).

[176] https://github.com/micbuffa/WebAudioPlugins

Le standard WAP permet de rapprocher le monde des applications natives, fort de milliers d'applications existantes, et le monde du Web : les standards WAMs et iPlug2 destinés à faciliter le portage de telles applications sont aujourd'hui compatibles avec WAP.

Les illustrations présentées dans ce chapitre montrent de nombreuses applications natives compilées en WebAssembly et tournant sous forme de WAP dans des applications web hôtes.

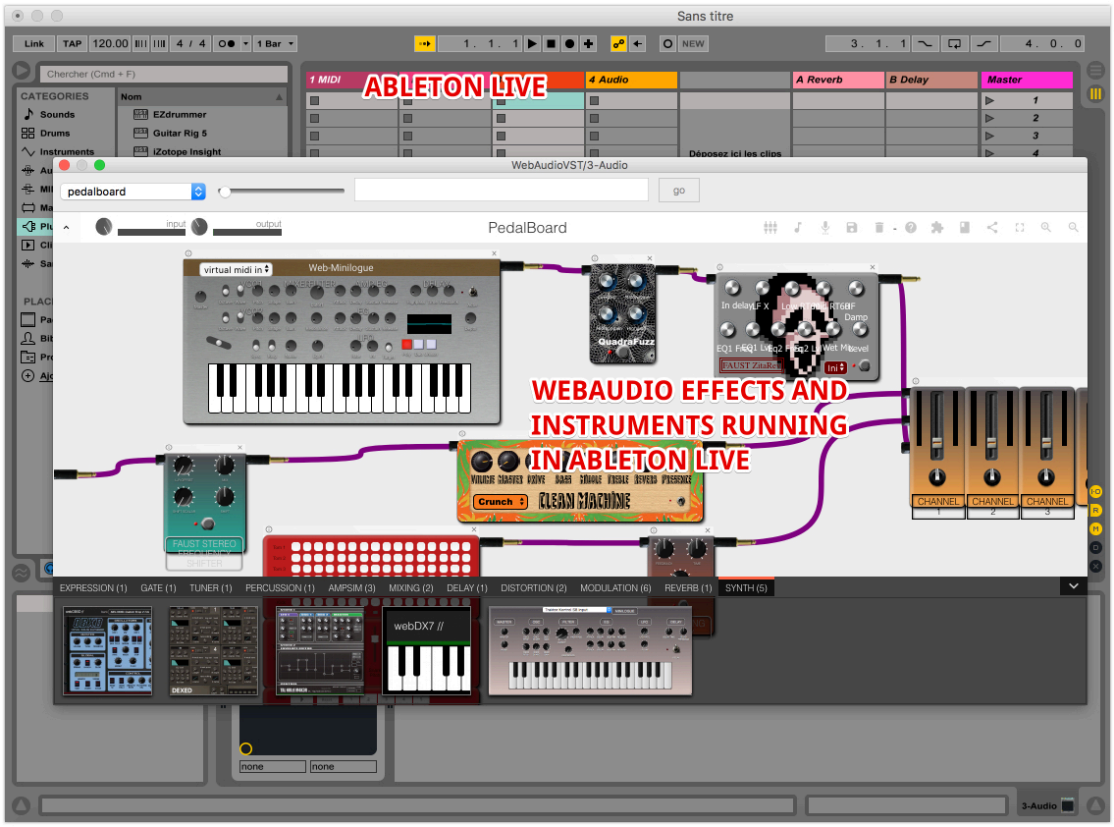


Figure 190. L'application pedalboard dans un plugin VST2 embarquant une version spécialement modifiée de CEF (Chromium Embedded Framework) pour rediriger les I/O sons et midi. Ce plugin intitulé WebAudioVST est chargé dans la station de travail audionumérique Ableton Live. On peut ainsi utiliser sans latence supplémentaire notable les plugins Web dans des environnements natifs.

Et vice-versa : en embarquant un navigateur Web embarqué dans « une coquille » de plugin natif, il est possible de charger des webapps hôtes (et donc des plugins WAP) dans des applications de MAO natives comme Ableton Live, Logic Audio, etc. (Figure 190)

L'implication directe est qu'un plugin WAP peut s'exécuter *dans tous les environnements disponibles Web et natif, sans modifier une seule ligne de code*, tant pour sa composante traitement du signal que pour son interface graphique, et avec des performances équivalentes.

12. Conclusion générale

Nous avons commencé avec un texte de vulgarisation scientifique sur l'histoire des outils collaboratifs et nous terminons ce manuscrit par des applications relevant de la Musique Assistée par Ordinateur, incluant de riches interfaces graphiques, des modèles de traitement du son complexes, une proposition de standard pour des composants logiciels audio réutilisables « universels » capables de fonctionner dans divers environnements, et pouvant être écrits à l'aide d'une variété de langages de programmation.

Entre temps, nous avons voyagé au pays du Web sémantique, modélisé des réseaux sociaux, fait naviguer un robot à distance dans des musées, simulé des amplificateurs de guitare à lampes…. Mais quel rapport entre tous ces travaux ?

Tout d'abord, la plateforme utilisée : le Web ! Et puis il y a les standards technologiques sous-jacents sans lesquels il n'aurait pas été possible de mettre en œuvre les algorithmes et modèles conceptuels élaborés par les travaux de recherche dans lesquels j'ai été impliqués.

Mais c'est également grâce à ces travaux, et à ceux de nombreux autres chercheurs, que ces standards ont évolué. Le cercle est vertueux !

Par exemple, le projet WASABI a repoussé les limites de l'audio sur le Web et c'est avec enthousiasme que nous avons participé à la création du standard WebAudio du W3C et que nous y apportons nos contributions.

Il en a été de même pour les standards du Web sémantique (les membres de l'équipe WIMMICS ont toujours été présents lors de la création et au cours de l'évolution de ces standards), ou pour le standard WebRTC qui a été utilisé pour le contrôle pair à pair du robot du projet AZKAR.

Nos projets se sont toujours inscrits à la fois dans des thématiques de recherche mais aussi dans des participations à des groupes de travail du W3C. Par nos retours, nos échanges, nos contributions à ces groupes de travail, nous avons aidé à établir les standards dont tout le monde bénéficie aujourd'hui : développeurs comme utilisateurs finaux.

Autre point commun à nos travaux : les graphes ! La modélisation des réseaux sociaux, des expositions de musées, des chansons, des effets audio, des interactions entre utilisateurs d'un même intranet, ce sont des graphes ! Et ce n'est pas une coïncidence, le Web est aussi un graphe ! Le modèle utilisé pour décrire les métadonnées RDF est un graphe également…

Oui, les points communs sont nombreux, mais il y en a un autre qui n'est pas explicite dans ce manuscrit : la passion qui ne m'a jamais quitté depuis que j'ai découvert en 1993 la première page web créée par la NASA montrant des images de la face cachée de Jupiter.

La passion de chercher, transmettre, partager la connaissance. Avec mes collaborateurs, avec mes étudiants, avec d'autres chercheurs, avec mes enfants, mes parents.

Et quel meilleur support que le Web pour s'affranchir des distances ? Je fus un des premiers à installer des wikis à l'Université de Nice en 1996 pour des projets étudiants collaboratifs internationaux, à réaliser des MOOCs en 2015, 2016 et 2017 qui ont rassemblé plus d'élèves que je n'en aurai jamais, issus de tous les pays du monde[177], avec à chaque fois le sentiment que ces contributions, gouttes d'eau dans l'Océan du Web, auront contribué, à une toute petite échelle, à étendre les usages du Web. WikiPedia existerait-il sans le travail préalable d'une communauté autour des wikis ?

En 2012 je n'avais que des connaissances rudimentaires en traitement du signal. Pendant plusieurs années j'ai lu des articles, des livres, discuté avec des personnes passionnées, en présentiel ou à distance, du fonctionnement des amplificateurs de guitare et des effets audio classiques. Et en 2017, quand le simulateur d'amplificateurs de guitare à lampes basé Web que j'ai créé avec Jérôme Lebrun a produit enfin des résultats que l'on pouvait comparer à l'état de l'art dans le domaine, j'ai pris conscience à la fois du merveilleux métier que j'exerçais -qui me laissait le loisir d'étudier un sujet aussi exotique pendant plusieurs années – et de l'incroyable potentiel du Web en tant que plateforme. Musicien amateur, usager de logiciels de MAO depuis la fin des années quatre-vingt-dix, je me suis longtemps demandé s'il serait un jour possible de faire fonctionner de telles applications dans un navigateur. Et là, entre 2012 et 2017 j'ai vu ce jour se rapprocher au fur et à mesure que la communauté WebAudio et son standard éponyme se développaient.

Je suis persuadé que les pratiques des musiciens nomades vont changer dans les années à venir. Le projet WASABI qui m'occupe aujourd'hui développe des applications WebAudio pour les écoles de musique et les écoles d'ingénieurs du son, pour les musiciens et les compositeurs, tout en s'inscrivant dans une logique scientifique. Il propose des outils uniques qui seront sans doute à repenser lors de WASABI 2 s'il voit le jour : la 5G à venir pouvant offrir une latence de 1ms si les serveurs et les terminaux sont situés à la périphérie du réseau. Cela ouvre des scénarios intéressants pour la création musicale et l'éducation future, en particulier pour la collaboration temps réel. Les techniques d'intelligence artificielle qui sont déjà appliquées dans le projet (pour la détection des genres de chansons, pour la mise en correspondance de configurations d'instruments avec des chansons, ou pour les instruments eux-mêmes lorsqu'ils jouent tout seuls) font l'objet d'une attention particulière dans le domaine de la musique et du e-learning, et il sera intéressant d'ici deux ans de voir quelles perspectives s'offrent à nous…Le monde de la recherche, lorsqu'on le couple aux évolutions techniques du Web, est passionnant !

Je suis un enseignant chercheur qui pense, qui transmet, qui code et surtout qui s'amuse !

J'ai envie de m'amuser encore longtemps ☺

Michel

[177] Nous avons vérifié : nous avons eu des étudiants en provenance d'absolument tous les pays référencés dans le monde, du plus petit au plus grand !

13. Bibliographie

1. **Baget, J-F., Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F., Giboin, A., Gutierrez, A., Leclère, M., Mugnier, M.-L., Thomopoulos, R., Griwes.** *Generic Model and Preliminary Specifications for a Graph-Based Knowledge Representation Toolkit.* Proc. ICCS'2008. (2008)
2. **Brandes, U., Pich, C.**: *Centrality estimation in large networks. Bifurcation and Chaos in Applied Sciences and Engineering* 17(7), 2303–2318. (2007)
3. **Breslin, J.G., Harth, A., Bojars, U., Decker, S.**: *Towards Semantically-Interlinked Online Communities.* Proceedings of the 2nd European Semantic Web Conference (ESWC '05), LNCS vol. 3532, pp. 500-514, Heraklion, Greece. (2005)
4. **Cavazza, F.**: http://www.fredcavazza.net/2009/04/10/social-media-landscape-redux/ (2009)
5. **Conein, B**. Communautés épistémiques et réseaux cognitifs: coopération et cognition distribuée. Revue D'Economie Politique 113, 141-159. (2004)
6. **Corby, O., Faron-Zucker, F.**: *Corese: A Corporate Semantic Web Engine.* Workshop on Real World RDF and Semantic Web Applications. (2002)
7. **Corby, O., Dieng-Kuntz, R., and Faron-Zucker, C.**: *Querying the Semantic Web with the Corese Search Engine.* ECA/PAIS2004. (2004)
8. **Corby, O** : *Web, Graphs and Semantics*, ICCS2008. (2008)
9. **Corby, O., Kefi-Khelif, L., Cherfi, H., Gandon, F., Khelif, K.** *Querying the Semantic Web of Data using SPARQL, RDF and XML*. INRIA Research Report n°6847. (2009)
10. **Erétéo, G., Buffa, M., Gandon, F., Grohan, P., Leitzelman, M., Sander, P.** *A State of the Art on Social Network Analysis and its Applications on a Semantic Web.* SDoW2008, Workshop at ISWC2008. (2008)
11. **Erétéo, G., Gandon, F., Corby, O., Buffa, M.** *Semantic Social Network Analysis.* Web Science 2009, Athens, Greece, 7-9 October (2009)
12. **Finin, T., Ding, L., Zou, L.** *Social networking on the semantic web.* Learning organization journal 5 (12): 418-435. (2005)
13. **Freeman, L.C.** Centrality in Social Networks: Conceptual Clarification. Social Networks 1, 215-239. (1979)
14. **Golbeck, J., Parsia, B., Hendler, J.** *Trust network on the semantic web.* Proceedings of cooperative information agents. (2003)
15. **Goldbeck, J., Rothstein, M.** *Linking social Networks on the web with FOAF.* Proceedings of the twenty-third conference on artificial intelligence, AAA08. (2008)
16. **Henri, F., Pudelko, B.** *Understanding and analyzing activity and learning in virtual communities.* Journal of Computer Assisted Learning 19, 474-487. (2003)
17. **McAfee, A.-P.** *Enterprise 2.0: The Dawn of Emergent Collaboration,* MIT Sloan Management Review, Management of Technology and Innovation, April 1 (2006)
18. **Mika, P**. *Ontologies are Us: a unified Model of Social Networks and Semantics.* In ISWC, volume 3729 of LNCS, p. 522-536: Springer. (2005)
19. **Paolillo, J.C., Wright, E.** *Social Network Analysis on the Semantic Web: Techniques and Challenges for Visualizing FOAF,* in Book Visualizing the semantic Web Xml-based Internet and Information. (2006)
20. **Newman, M. E. J.** *"The Structure and Function of Complex Networks"*. SIAM rev. 45, 167-256. (2003)
21. **San Martin, M., Gutierrez, C.** Representing, Querying and Transforming Social Networks with RDF / SPARQL. ESWC09. (2009)
22. **Scott, J.** *Social Network Analysis,* a handbook, second edition. Sage. (2000)

23. **Vidou, G., Dieng-Kuntz, R., El Ghali, A., Evangelou, C., Giboin, A., Tifous, A., Jacquemart, S**. *Towards an Ontology for Knowledge Management in Communities of Practice.* (2006)
24. **Shirky, C.** *A Group Is Its Own Worst Enemy*, A speech at ETech, available on the Web at : http://www.shirky.com/writings/group_enemy.html, (2003)
25. **Fagin, R., Kumar, R., McCurley, K., Novak, J., Sivakumar, D., Tomlin, J. and Williamson, D.** *Searching the Corporate Web*. In Proceedings of WWW2003, Budapest, Hungary, 366-375, (2003)
26. **Majchrzac, A., Wagner, C., Yates, D.** *"Corporate Wiki Users: Results of a Survey"*, Proc. of the ACM International Symposium on Wikis (Wikisym 2006), Odense, Denmark, 2006
27. **Völkel, M., Oren, E.** *"Towards a Wiki Interchange Format (WIF)"*, Proc. of the 1st Workshop on Semantic Wikis, Budva, Montenegro, 2006
28. **Limpens, F., Gandon, F. and Buffa, M.** *Helping Online Communities to Semantically Enrich Folksonomies*, Web Science Conference, April 2010, Raleigh, NC, USA.
29. **Limpens, F., Gandon, F. and Buffa, M.** *Collaborative Semantic Structuring of Folksonomies*, Conference IEEE Web Intelligence 2009, Milan September 15-18.
30. **Limpens, F., Buffa, M., Gandon, F.** *A complete life-cycle for the semantic enrichment of folksonomies*, book chapter in "Advances in Knowledge Discovery and Management Vol. 3 (AKDM-3)", Series "Studies in Computational Intelligence," F. Guillet, P. Pinaud, G. Venturini and D. Zighed (editors), Springer, 2013, page 133.
31. **Raghavan, U N, Albert, R. and Kumara, S.** Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 036106, (2007)
32. **Erétéo, G., Buffa, M., Gandon, F., Grohan, P., Leitzelman, M., Sander, P.** *A State of the Art on Social Network Analysis and its Applications on a Semantic Web*. SDoW2008, Workshop at ISWC2008, Karlsruhe, Germany (2008)
33. **Java, A., Joshi, A., Finin, T.** *Detecting Communities via Simultaneous Clustering of Graphs and Folksonomies.* WebKDD 2008. (2008)
34. **Buffa, M., Gandon, F., Ereteo, G., Sander P. and Faron, C.** *SweetWiki: A semantic wiki*, Special Issue of the Journal of Web Semantics on Semantic Web and Web 2.0, Volume 6, Issue 1, February 2008 , Edited by Mark Greaves and Peter Mika, Elsevier, Pages 84-97 PDF: http://smtp.websemanticsjournal.org/index.php/ps/article/viewFile/138/136
35. **Buffa, M., Gandon, F.** *SweetWiki: Semantic Web Enabled Technologies in Wiki*, WikiSym, IEEE International Symposium on Wikis, August 21-23, 2006, Odense, Denmark. Accept rate 31%. PDF: http://www.wikisym.org/ws2006/proceedings/p135.pdf
36. **Meseguer-Brocal, G., Peeters, G., Pellerin, G., Buffa, M., Cabrio, E. et al**. *"WASABI: A Two Million Song Database Project with Audio and Cultural Metadata plus WebAudio enhanced Client Applications".* Web Audio Conference 2017 – Collaborative Audio #WAC2017, Aug 2017, London, United Kingdom. 2017. 〈hal-01589250〉
37. **Buffa, M., Lebrun, J.** *Real-Time Emulation of a Marshall JCM 800 Guitar Tube Amplifier, Audio FX Pedals, in a Virtual Pedal Board*. WWW2018 - TheWebConf 2018 : The Web Conference, 27th International World Wide Web Conference, Apr 2018, Lyon, France. 2018, 〈hal-01721463〉
38. **Kleimola, J. and Larkin, O**. 2015. *Web Audio modules.* In Proc. 12th Sound and Music Computing Conference (SMC 2015), Maynooth, Ireland.
39. **Larkin, O., Harker, A., and Kleimola J.** 2018. *iPlug 2: Desktop Audio Plug-in Framework Meets Web* Audio Modules Submitted to the 4th Web Audio Conference (WAC-2018), Berlin, Germany.
40. **Buffa, M., Lebrun, J., Kleimola, J. and Larkin, O, Letz, S**. "*Towards an open Web Audio plug-in standard".* WWW2018 - TheWebConf 2018 : The Web Conference, 27th International World Wide Web Conference, Mar 2018, Lyon, France. 〈hal-01721483〉
41. **Buffa, M., Lebrun, J.** *Real time tube guitar amplifier simulation using WebAudio.* Web Audio Conference 2017 – Collaborative Audio #WAC2017, Aug 2017, London, United Kingdom. 2017. 〈hal-01589229〉
42. **Pakarinen, J., & Yeh, D. T.** *A review of digital techniques for modeling vacuum-tube guitar amplifiers.* Computer Music Journal, 33(2), 85-100. (2009)

43. **Yeh, D. T., & Smith, J. O.** *Discretization of the'59 Fender Bassman tone stack.* In Int. Conf. on Digital Audio Effects (DAFx-06) (pp. 18-20). (2006, September)
44. **Holmes, B., & van Walstijn, M.** *Improving the robustness of the iterative solver in state-space modeling of guitar distortion circuitry.* 18th Int. Conference on Digital Audio Effects (DAFx-15), Trondheim, Norway, 2015.
45. **Cheng-Hao C**. *A Guitar Overdrive/Distortion Effect of Digital Signal Processing*. https://tinyurl.com/kc6s4er
46. **Macak, J., & Schimmel, J**. *Real-time guitar tube amplifier simulation using an approximation of differential equations.* In Proceedings of the 13th International Conference on Digital Audio Effects (DAFx'10). 2010.
47. **Yeh, D. T., Abel, J. S., Vladimirescu, A., & Smith, J. O.** *Numerical methods for simulation of guitar distortion circuits.* Computer Music Journal, 32(2), 23-42. (2008).
48. **Clark, J. J.** *Advanced programming techniques for modular synthesizers.* https://tinyurl.com/3xskmv (2003).
49. **Cohen, I., & Helie, T.** Simulation of a guitar amplifier stage for several triode models: examination of some relevant phenomena and choice of adapted numerical schemes. In Audio Engineering Society Convention 127. Audio Engineering Society. (2009, October)
50. **Berners-Lee, Tim, James Hendler, and Ora Lassila**. "*The semantic web*." Scientific american 284.5 (2001): 34-43.
51. **Rech, J., Bogner, C., & Haas, V.** *Using wikis to tackle reuse in software projects.* Software, IEEE, 24(6), 99-104. (2007).
52. **Buffa, M., Grattarola, J-C., Sander, P.** *Une nouvelle forme de coopération, nationale et internationale, entre laboratoires de recherche et filières d'enseignement.* Numéro 10 de la revue « Information Science for Decision Making » (ISDM), numéro 10 d'Octobre 2003. http://isdm.univ-tln.fr/articles/num_archives.htm#isdm10
53. **Dirat, L.** *Outils pour des mathématiques interactives et distribuées*, thèse de doctorat de troisième cycle, Université de Nice, 2001.
54. **Buffa, M.** *Du Web aux Wikis, histoire des outils collaboratifs.* Revue Interstices, Mai 2008. http://interstices.info/jcms/c_37151/du-web-aux-wikis-une-histoire-des-outils-collaboratifs
55. **Buffa, M.** *Intranet Wikis*, workshop "Intrawebs" of WWW 2006, Edinburgh, Scotland.PDF : http://wwwconference.org/proceedings/www2006/www-sop.inria.fr/acacia/WORKSHOPS/IntraWebs2006/Buffa_Intrawebs2006.pdf
56. **Dirat, L., Buffa, J-M., Fédou, J-M., Sander, P.** *JOME, un composant logiciel pour le télé-enseignement des mathématiques via le Web, compatible OpenMath et MathML*. International Conference TICE 2000, Troyes, France.
57. **Buffa, M., Dirat, L., J-M., Fédou, Sander, P.** *JOME : Java OpenMath Editor*, International Conference ICTMT 4, Plymouth, 9-13 August 1999.
58. **Buffa, M., Sander P., Grattarola J.-C.** *Distant cooperative software development for research and education: three years of experience.* In proceedings of CALIE'04, Grenoble, France, . (2004).
59. **Chat, C. and Nahaboo, C.** *Let's Build an Intranet at ILOG Like the Internet!*, workshop "Intrawebs" of WWW 2006, Edinburgh. (2006).
60. **Cunningham, W and Leuf, B**. *The Wiki Way: Quick collaboration on the web.* Addison-Wesley, Boston. (2001).
61. **Levi, P.** *L'intelligence collective*, Éditions La Découverte. Paris, 1997, p. 12. (1997).
62. **Stenmark, D**. *Knowledge sharing on a corporate intranet: Effects of re-instating web authoring capabilities*. Proceedings of ECIS 2005, Regensburg, Germany, 26-28 May 2005.
63. **Stenmark, D.** *Intranets as formative context: A study of under-utilised corporate webs*. I. Proceedings of AMCIS 2003, Tampa, FL, 1697-1703. (2003a).
64. **Stenmark, D.** Knowledge sharing on a corporate intranet: Effects of re-instating web authoring capabilities. Proceedings of ECIS 2005, Regensburg, Germany, 26-28 May 2005.

65. **Stenmark, D.** *Organisational creativity in context: Learning from a failing attempt to introduce IT support for creativity*. International Journal of Technology and Human Interaction, Vol 1, Issue 4, pp. 80-98. (2005).

66. **Hahn, J. and Subramani, M. R.** *A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice*. In Proceedings of ICIS 2000, 302-312. (2000)

67. **Shirky, C.** *A Group Is Its Own Worst Enemy*, A speech at ETech, (2003) available on the Web at : http://www.shirky.com/writings/group_enemy.html

68. **Chang, B.-W**. *In-place editing of Web pages: Sparrow community-shared documents.* In Proceedings of WWW7, Brisbane, Australia. (1998).

69. **Fagin, R., Kumar, R., McCurley, K., Novak, J., Sivakumar, D., Tomlin, J. and Williamson, D**. *Searching the Corporate Web*. In Proceedings of WWW2003, Budapest, Hungary, 366-375. (2003).

70. **Ciborra, C.** *A Critical Review of the Literature on the Management of Corporate Information Infrastructure.* In Ciborra el al. (eds.) From Control to Drift, Oxford University Press, 15-40. (2000).

71. **Curry, A. and Stancich, L.** *The Intranet–an intrinsic component of strategic information management?,* International Journal of Information Management, 20, 249-268. (2000).

72. **Damsgaard, J. and Scheepers, R.** *Managing the crises in Intranet implementation: a stage model*. Information Systems Journal, 10(2), 131-149. (2000).

73. **Finck N. (Digital Web Magazine), Hodder M., and Stone B. (Google).** "*Enhancing Internal Communications with Blogs, Wikis, and More*" (2005) http://www.nickfinck.com/presentations/bbs2005/01.html

74. **Merrill D.** *A view into Google's inner workings*, audio report about presentation at Vortex 2005: http://www.citizenvalley.org/blocnotes . (2005)

75. **Coyne K.P., Stover A., and Nielsen J.** *Designing Usable Intranets, available on the Web*. http://www.NNgroup.com/reports/intranet/guidelines (2002)

76. **Rupley, S.** *What's a Wiki?* PC Magazine, May 9 issue. Available on the web at: http://www.pcmag.com/article2/0%2C1759%2C1071705%2C00.asp (2005).

77. **Schremmer, C., Effelsberg, W., Grattarola, J.-C. and Berera, E**. "*A Novel Student Cooperation Across Country Borders*.", In: Proc. 4th International Conference on New Educational Environments (ICNEE) 2002, Lugano, CH, 08.05.2002.

78. **Hammond T., Hannay T., Lund B., and Scott J.** *Social Bookmarking Tools, a General Review*, D-Lib Magazine, April 2005, Volume 11 Number 4, http://www.dlib.org/dlib/april05/hammond/04hammond.html , (2005).

79. **Decker B., Ras E., Rech J., Klein B. and Hoecht C.** *Self-organized "Reuse of Software Engineering Knowledge Supported by Semantic Wikis*", Proc. of the Workshop on Semantic Web Enabled Software Engineering (SWESE), ISWC, Galway, Ireland, 2005

80. **Campanini, S.E., Castagna, P. and Tazzoli R**. *Platypus Wiki: a Semantic Wiki Wiki Web*. Proc. of the 1st Italian Semantic Web Workshop, Ancona, Italy, 2004.

81. **Aumueller, D.** *SHAWN: Structure Helps a Wiki Navigate*. Proc. of the BTW-Workshop WebDB Meets IR, Karlsruhe, Germany, 2005. http://dbs.uni-leipzig.de/~david/2005/aumueller05shawn.pdf

82. **Decker, B., Ras, E., Rech, J., Klein, B. and Hoecht, C**. *Self-organized "Reuse of Software Engineering Knowledge Supported by Semantic Wikis"*, Proc. of the Workshop on Semantic Web Enabled Software Engineering (SWESE), ISWC, Galway, Ireland, 2005

83. **Souzis A.**, *Building a Semantic Wiki*, IEEE Intelligent Systems, 20 (5), September/October, 2005

84. **Krötsch, M., Vrandečić, D., Völkel M**. *Wikipedia and the Semantic* Web - *The Missing Links*, WikiMania 2005.

85. **Muljadi, H. and Takeda, H**. *Semantic Wiki as an Integrated Content and Metadata Management System*, Proc. of the 4th International Semantic Web Conference, Galway, Ireland, 2005, , LNCS 3729, Springer Verlag.

86. **Dello, K., Tolksdorf, R. and Paslaru, E.** *Makna*. Free University of Berlin. http://www.apps.ag-nbi.de/makna/wiki/About, 2005.

87. **Aumueller, D. and Auer, S**. *Towards a Semantic Wiki Experience – Desktop Integration and Interactivity in WikSAR*, Proc. of the Workshop on Semantic Desktop, Galway, Ireland, 2005.

88. **Schaffert, S., Gruber, A., and Westenthaler, R.** *A Semantic Wiki for Collaborative Knowledge Formation.* Proc. of the Semantics 2005, Vienna, Austria.

89. **Schaffert, S., Bischof, D., Buerger, T., Gruber, A., and Hilzensauer, W.**, *Learning with Semantic Wikis*. Proc of 1st Workshop on Semantic Wikis, Budva, Montenegro, 2006.

90. **Hepp, M., Bachlechner, D. and Siorpaes, K.** *OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis*. Proc. of the International Symposium on Wikis (WikiSym 2005), San Diego, USA, 2005.

91. **Auer, S.,** *Powl – A Web Based Platform for Collaborative Semantic Web Development*. Proc. of the 1st Workshop on Scripting for the Semantic Web (SFSW'05), Hersonissos, Greece, 2005.

92. **Corby, O., Dieng-Kuntz, R, Faron-Zucker, C.** *Querying the Semantic Web with the CORESE Search Engine*. Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, Spain, 2004, IOS Press

93. **Mizoguchi, R, Ikeda, M, Sinitsa, K**. *Roles of Shared Ontology in AI-ED Research, Intelligence, Conceptualization, Standardization, and Reusability*. Proc. of the 8th World Conference on Artificial Intelligence in Education, 1997, IOS Press.

94. **Corby, O., Dieng-Kuntz, R., Faron-Zucker C., Gandon F**. *Searching the Semantic Web: Approximate Query Processing based on Ontologies*. IEEE Intelligent Systems Journal, 21(1), 2006.

95. **Durville P., Gandon F.** *SeWeSe : Semantic Web Server.* WWW2007 Developers track, http://www-sop.inria.fr/acacia/personnel/Fabien.Gandon/research/www_2007_dev_track/

96. **Connolly D**. *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)*. W3C Recommendation 11 September 2007, http://www.w3.org/TR/grddl/

97. **Bird, F.** *Some Ideas to Improve Tags Use in Social Software, Flat Hierarchy versus Categories in Social Software*. http://fredbird.org/lire/log/2005-05-17-tags-structuration-proposal, 2005. Available on web.archive.org (https://web.archive.org/web/20050830112007/http://fredbird.org/lire/log/2005-05-17-tags-structuration-proposal)

98. **Olsen, H.**, *Navigation blindness, how to deal with the fact that people tend to ignore navigation tools*. The Interaction Designer's Coffee Break, Issue 13, Q1 2005.

99. **Gruber, T.** *Ontology of Folksonomy: A Mash-up of Apples and Oranges*. First on-Line Conference on Metadata and Semantics Research (MTSR'05). http://mtsr.sigsemis.org/, (2005).

100. **Hammond, T, Hannay, T, Lund, B., and Scott, J**. *Social Bookmarking Tools, a General Review*. D-Lib Magazine, April 2005, 11(4),

101. **Powers, S.**, *Cheap Eats at the Semantic Web Café*, , http://weblog.burningbird.net/archives/2005/01/27/cheap-eats-at-the-semantic-web-cafe/. (2005).

102. **Newman, R**. *Tag Ontology Design*. http://www.holygoat.co.uk/blog/entry/2005-03-23-2 (2007).

103. **Passant, A**. *Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs: Theoretical Background and Corporate Use-Case*. Proc. of the International Conference ICWSM, Boulder, USA, (2007).

104. **El Ghali ,A., Tifous, A., Buffa, M., Giboin, A., Dieng-Kuntz, R**. *Using a Semantic Wiki in Communities of Practice*. Proc. of TEL-CoPs'07 Workshop, EC-TEL'07, Crete, Greece, 2007.

105. **Gandon, F. et Al.** *ISICIL : Information Semantic Integration through Communities of Intelligence Online*. In Proc. International Workshop on Web Intelligence and Virtual Enterprise (WIVE), 10th IFIP Working Conference on Virtual Enterprises (PRO-VE'09) Thessaloniki, Greece. (2009).

106. **Erétéo, G**. *Semantic Social Network Analysis*. Ph. D. thesis, Telecom ParisTech. (2011).

107. **Delaforge, N., F. Gandon, et N. Vukosavljevic**. *Webmark : pour une approche contextuelle de la référence en ligne*. In Atelier Philosophie et Ingénierie, Le formel face à l'histoire, la technologie et la matérialité, IC2011 , Chambéry, France. (2011).

108. **Villata, S., F. Gandon, A. Gyrard, et N. Delaforge**. *An Access Control Model for Linked Data*. In Proc. 7th International IFIP Workshop on Semantic Web & Web Semantics, SWWS , Hersonissos, Crete, Greece. (2011)

109. **Völkel, M., M. Krötzsch, D. Vrandecic, H. Haller, et R. Studer.** *Semantic wikipedia*. In WWW '06: Proceedings of the 15th international conference on World Wide Web , New York, NY, USA, pp. 585–594. ACM. (2006).

110. **Schaffert, S., Westenthaler, R., et Gruber, A.** *IkeWiki: A user-friendly semantic wiki*. In 3rd European Semantic Web Conference (ESWC06) . (2006).

111. **Auer, S., Dietzold, S. and Riechert, T**. OntoWiki - *A Tool for Social, Semantic Collaboration.* In The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings. (2006).

112. **Meilender, T., Jay, N. Lieber, J. et Palomares, F.** *Les moteurs de wikis sémantiques un état de l'art.* In Extraction et gestion des connaissances (EGC'2011), Actes, 25 au 29 janvier 2011, Brest, France, pp. 575–580. (2011).

113. **Orlandi, F**. *Using and extending the sioc ontology for a fine-grained wiki modeling*. Master's thesis, Università degli Studi di Modena e Reggio Emilia. (2008).

114. **Schaffert, S. et Al**. *KiWi - A Platform for Semantic Social Software*. In 4th Semantic Wiki Workshop (SemWiki 2009) at the 6th European Semantic Web Conference (ESWC 2009), Hersonissos, Greece, June 1st, 2009.

115. **Nielsen, J**. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (1993).

116. **Cooper, A., Reimann, R. and Cronin, D**. *About Face 3: The Essentials of Interaction Design*. Number 3. Wiley. (2007).

117. **Buffa, M., Husson, G., Delaforge, N.** *SweetDeki : "Le wiki sémantique couteau suisse du réseau social ISICIL"*. 12e Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2012), Jan 2012, Bordeaux, France. pp.387-398. ⟨hal-01790117⟩ **Honorable mention for the best application paper.** Also published in RNTI journal issue EGC 2012, vol. RNTI-E-23, pp.113-124

118. **Arapov, P., Buffa, M., Ben Othmane, A.** "*WikiNEXT: a wiki for exploiting the web of data"*. SAC 2014, 29th Symposium On Applied Computing, Mar 2014, Gyeongju, South Korea. pp.8, 2014, Proceeding SAC '14 Proceedings of the 29th Annual ACM Symposium on Applied Computing Pages 727-734. ⟨hal-01083338⟩

119. **Buffa, M., Delaforge, N., Ereteo, G., Gandon, F., Giboin, A. and Limpens, F.** *"ISICIL: Semantics and Social Networks for Business Intelligence"*, conference SOFSEM 2013, 39th International Conference on Current Trends in Theory and Practice of Computer Science. January 26–31, 2013 Špindlerův Mlýn, Czech Republic.

120. **Arapov, P., and Buffa, M.** *WikiNext, a JavaScript semantic wiki*. In developer track, WWW2012 Conference, Lyon. (2012, April).

121. **Oren, E., Breslin, J. G., & Decker, S**. *How semantics make better wikis*. In Proceedings of the 15th international conference on World Wide Web (pp. 1071-1072). ACM. (2006, May).

122. **Oren E.** SemperWiki: a semantic personal Wiki. In SemDesk. 2005

123. **Schaffert, S.** *IkeWiki: A semantic wiki for collaborative knowledge management*. In Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on (pp. 388-396). IEEE. (2006, June).

124. **Auer, S., Dietzold, S., & Riechert, T.** *OntoWiki–A tool for social, semantic collaboration*. In "The Semantic Web"-ISWC 2006 (pp. 736-749). Springer Berlin Heidelberg. (2006).

125. **Meilendera, T., Jayb, N., Lieberb, J., & Palomaresa, F.** *Semantic wiki engines: a state of the art.* Semantic-web-journal.net. IOS Press, 2010.

126. Rospocher, M., Ghidini, C., Pammer, V., Serafini, L., Lindstaedt, S. *MoKi: the Modelling wiKi*. Conference SemWiki 2009.

127. **Di Iorio, A., Musetti, A., Peroni, S., Vitali, F.** *Ontology-driven generation of wiki content and interfaces.* The New Review of Hypermedia and Multimedia 16(1&2): 9-31 (2010)

128. **Crespo, A. & Bier, E**. *WebWriter: A browser-based editor for constructing web applications*. Computer Networks and ISDN Systems, 1996, vol. 28, no 7, p. 1291-1306. (1996)

129. **Rech, J., Bogner, C., & Haas, V**. *Using wikis to tackle reuse in software projects*. Software, IEEE, 24(6), 99-104. (2007).

130. **Hernandez A. G. and Garcıa M. N. M.** *A JavaScript RDF store and application library for linked data client applications.* In Devtracks of the, WWW2012, conference. Lyon, France. 2012

131. **Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A. and Stumme G**. *Evaluating similarity measures for emergent semantics of social tagging*. In 18th International World Wide Web Conference, pages 641–641, April 2009.

132. **Cattuto, C. Benz, D. Hotho A., and Stumme. G.** *Semantic grounding of tag relatedness in social bookmarking systems*. In ISWC '08: Proceedings of the 7th International Conference on The Semantic Web, pages 615–631, Berlin, Heidelberg, 2008. Springer-Verlag.

133. **Mika, P.** *Ontologies are Us: a Unified Model of Social Networks and Semantics.* In ISWC, volume 3729 of LNCS, pages 522–536. Springer, 2005.

134. **Heymann, P. and Garcia-Molina, H.** Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Stanford InfoLab, 2006.

135. **Tanasescu, V. and Streibel O.** *Extreme tagging: Emergent semantics through the tagging of tags.* In Peter Haase, Andreas Hotho, Luke Chen, Ernie Ong, and Philippe Cudre Mauroux, editors, Proceedings of the International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC2007, Busan, South Korea, November 2007.

136. **Huynh-Kim Bang, B. Dan, E. and Grandbastien, M.** *Merging semantic and participative approaches for organising teachers' documents*. In Proceedings of ED-Media 08 ED-MEDIA 08 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, pages p. 4959–4966, Vienna France, 07 2008.

137. **Passant, A. and Laublet. P**. *Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data.* In Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, Apr 2008.

138. **Limpens, F. Monnin, A., Laniado, D. and Gandon, F.** *Nicetag ontology: tags as named graphs.* In International Workshop in Social Networks Interoperability, Asian Semantic Web Conference 2009, 2009.

139. **Ribière, M**. Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels. PhD thesis, Université Nice-Sophia Antipolis, 1999.

140. **Bouquet, P. Giunchiglia, F. Van Harmelen, F. Serafini, L. and Stuckenschmidt. H.** *Contextualizing ontologies.* Web Semantics: Science, Services and Agents on the World Wide Web, 1(4):325 – 343, 2004. International Semantic Web Conference 2003.

141. **Angeletou, S. Sabou, M. and Motta, E.** *Semantically Enriching Folksonomies with FLOR.* In CISWeb Workshop at European Semantic Web Conference ESWC., 2008.

142. **Specia, L. and Motta, E.** *Integrating folksonomies with the semantic web.* In Proc. of the European Semantic Web Conference (ESWC2007), volume 4519 of LNCS, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.

143. **Van Damme, C. Hepp, M. and Siorpaes, K.** *Folksontology: An integrated approach for turning folksonomies into ontologies.* In Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007), pages 57–70, 2007.

144. **Aussenac-Gilles, N., Biébow, B., & Szulman, S.** *Corpus analysis for conceptual modelling.* In Workshop on Ontologies and Texts at Knowledge Acquisition, Modeling and Management, 12th International Conference, EKAW 2000, 2000.

145. **Golebiowska, J.** Exploitation des ontologies pour la memoire d'un projet-vehicule - Methode et outil SAMOVAR. PhD thesis, Université de Nice-Sophia Antipolis, 2002.

146. **Braun, S., Schmidt, A. P., Walter, A., Nagypal, G., & Zacharias, V.** *Ontology maturing: a collaborative web 2.0 approach to ontology engineering*. In CKC, volume 273 of CEUR Workshop Proceedings. CEUR-WS.org, 2007.

147. **Carroll, J. Bizer, C. Hayes, P. and Stickler, P**. *Named graphs, provenance and trust*. In WWW '05: Proceedings of the 14th international conference on World Wide Web, pages 613–622, New York, NY, USA, 2005. ACM.

148. **Gandon, F. Bottolier, V. Corby, O. and Durville, P**. *Rdf/xml source declaration*, w3c member submission. http://www.w3.org/Submission/rdfsource/, 09 2007.

149. **Levenshtein. V. I.** *Binary codes capable of correcting deletions, insertions and reversals*. Soviet Physics Doklady., 10(8):707–710, February 1966.

150. **Erétéo, G. , Gandon, F., Buffa, M., Corby, O.** *Analysis of a Real Online Social Network Using Semantic Web Frameworks*. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 180–195. Springer, Heidelberg (2009). Accept rate 17%. Only paper from a french research group. PDF: http://www-sop.inria.fr/members/Fabien.Gandon/docs/ISWC2009_ereteo_et_al.pdf

151. **Leicht, E. A., & Newman, M. E.**: *Community structure in directed networks*, Phys. Rev. Lett. 100, 118703 (2008)

152. **Ne Newman, M. E.** Fast algorithm for detecting community in networks. Phys. Rev. E 69, 066133 (2004)

153. Baget, J.-F., Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F., Giboin, A.,Gutierrez, A., Leclère, M., Mugnier, M.-L., Thomopoulos, R.: Griwes: *Generic Modeland Preliminary Specifications for a Graph-Based Knowledge Representation Toolkit.* In:Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 297–310.Springer, Heidelberg (2008)

154. **Ereteo, G., Buffa, M., Gandon, F., and Corby, O.** *Analysis of a real online social network using semantic web frameworks.* In Proc. International Semantic Web Conference, ISWC'09, Washington, USA, 2009

155. **Burt, RS**: *Structural Holes*. Cambridge University Press, New York (1992)

156. **Buffa, M., Faron-Zucker, C**. *Ontology-Based Access Rights Management*. In: Guillet F., Ritschard G., Zighed D. (eds) Advances in Knowledge Discovery and Management. Studies in Computational Intelligence, vol 398. Springer, Berlin, Heidelberg. (2012).

157. **Buffa, M., Faron-Zucker, C., & Kolomoyskaya, A.** *Gestion sémantique des droits d'accès au contenu : l'ontologie AMO*. In EGC (Vol. 10, pp. 471-482). (2010)

158. **Kruk, S.** FOAF-Realm-control your friends' access to the resource. In: FOAF Workshop proceedings. 2004.

159. **Alam, A., G. Subbiah, B. Thuraisingham, and Khan, L.** *Reasoning with Semantics-aware Access Control Policies for Geospatial Web Services.* In 3rd ACM Workshop On Secure Web Services, SWS 2006, pp. 69–76. (2006).

160. **Hollenbach, J., J. Presbrey, et T. Berners-Lee.** *Using RDF Metadata to Enable Access Control on Social Semantic Web*. In International Semantic Web Conference, ISWC 2009, LNCS. Springer. (2009)

161. **Coyle, K.** Rights Management and Digital Library Requirements. Ariadne 40. (2004)

162. **Lagoze, C., S. Payette, E., Shin, and Wilper, C.** *Fedora: an Architecture for Complex Objects and their Relationships*. Int. J. on Digital Libraries 6(2), 124–138. (2006).

163. **Kruk, S. R.** *Extensible Access Control (EAC) Ontology Specification*. DERI, http://www.jeromedl.org/eac/1.0/spec/index.html/. (2008).

164. **Kruk, S. R., M. Cygan, and Gzella, A**. JeromeDL - Semantic and Social Technologies for Improving User Experience in Digital Libraries. In World Wide Web Conference, WWW 2008. ACM. (2008).

165. **Story, H., B. Harbulot, I. Jacobi, et M. Jones.** *FOAF+SSL: RESTful Authentication for the Social Web.* In ESWC Workshop Trust and Privacy on the Social and Semantic Web, SPOT 2009. (2009).

166. **Corby, O., Dieng-Kuntz, R., Faron-Zucker, C. and Gandon, F.** *Searching the Semantic Web: Approximate Query Processing Based on Ontologies*. IEEE Intelligent Systems 21(1), 20–27. (2006).

167. **Gandon, F**. *Pour tout le monde : Tim Berners-Lee, lauréat du prix Turing 2016 pour avoir inventé... le Web*. 1024 : Bulletin de la Société Informatique de France, Société Informatique de France, 2017, 〈http://www.societe-informatique-de-france.fr/bulletin/1024-numero-11/〉. 〈hal-01623368〉

168. **Kuhn, T.** *Acewiki: collaborative ontology management in controlled natural language.* Arxiv preprint arXiv:0807.(4623). Available online at: http://arxiv.org/abs/0807.4623. (2008).
169. **Lange, C., & Kohlhase, M.** *SWiM: A semantic wiki for mathematical knowledge management.* In Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications (pp. 47-68). IGI Global. (2008).
170. **Limpens, F.** Multi-points-of view semantic enrichment of folksonomies. PhD thesis, University of Nice. (2010)
171. **Bertin-Mahieux, T., Ellis, D. P. , Whitman, B. and Lamere, P**. *The million song dataset.* In ISMIR, volume 2, page 10, (2011).
172. **Allik, A., Mora-Mcginity, M., Fazekas, G., & Sandler, M. B.** *MusicWeb: Music Discovery with Open Linked Semantic Metadata.* In International Semantic Web Conference (Posters & Demos). (2016).
173. **Allik, A., Thalmann, F., & Sandler, M.** *MusicLynx: Exploring Music Through Artist Similarity Graphs.* In Companion of the The Web Conference 2018 on The Web Conference 201 8(pp. 167-170). International World Wide Web Conferences Steering Committee. (2018, April).
174. **Fillon, T., Simonnot, J., Mifune, M. F., Khoury, S., Pellerin, G., & Le Coz, M.** *Telemeta: An open-source web framework for ethnomusicological audio archives management and automatic analysis*. In Proceedings of the 1st International Workshop on Digital Libraries for Musicology (pp. 1-8). ACM. (2014, September).
175. **Raimond, Y., Abdallah, S. A., Sandler, M. B. and Giasson, F.** *The music ontology*. In ISMIR, pages 417-422. Citeseer, 2007.
176. **Allik, A., Fazekas, G., Barthet, M., & Swire, M**. *myMoodplay: an interactive mood-based music discovery app.* WebAudio Conference 2016, Atlanta. (2016).
177. **Thalmann, F. S., & Perez Carillo, A.** The semantic music player: A smart mobile player based on ontological structures and analytical feature metadata. WebAudio Conference 2016, Atlanta. (2016)
178. **Schreiber, H**. Improving genre annotations for the million song dataset. In ISMIR, pages 241-247, 2015.
179. **Mahedero, J. P. G., Martinez, A. , Cano, P. , Koppenberger, M. and Gouyon, F.** *Natural language processing of lyrics.* In Proceedings of the 13th ACM International Conference on Multimedia, Singapore, November 6-11, 2005, pages 475{478, 2005.
180. **Schulder, M., & Hovy, E.** *Metaphor detection through term relevance*. In Proceedings of the Second Workshop on Metaphor in NLP (pp. 18-26). (2014).
181. **Sterckx, L., Demeester, T., Deleu, J., Mertens, L., & Develder, C.** *Assessing quality of unsupervised topics in song lyrics.* In European Conference on Information Retrieval (pp. 547-552). Springer, Cham. (2014, April).
182. **Mihalcea, R., & Strapparava, C.** *Lyrics, music, and emotions*. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 590-599). Association for Computational Linguistics. (2012, July).
183. **Tashiro, T., Ueda, T., Hori, T., Hirate, Y., & Yamana, H.** *EPCI: extracting potentially copyright infringement texts from the web.* In Proceedings of the 16th international conference on World Wide Web (pp. 1151-1152). ACM. (2007, May).
184. **Leung, C. H., & Chan, Y. Y.** *A natural language processing approach to automatic plagiarism detection.* In Proceedings of the 8th ACM SIGITE conference on Information technology education (pp. 213-218). ACM. (2007, October).
185. **Fell, M., Nechaev, Y., Cabrio, E., & Gandon, F.** (2018*). Lyrics Segmentation: Textual Macrostructure Detection using Convolutions.* In Proceedings of the 27th International Conference on Computational Linguistics (pp. 2044-2054). (2018).
186. **Meseguer-Brocal, G., Cohen-Hadria A., and Peeters, G.** DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using teacher-student machine learning paradigm. ISMIR 2018, Paris. (2018).
187. **Buffa M., Demetrio M., Azria N.** *Guitar pedal board using WebAudio. Web Audio Conference*, Apr 2016, Atlanta, United States. (2016)

188. **Zakai, A**. 2011. *Emscripten: an LLVM-to-JavaScript compiler*. ACM Int. Conf. companion on Object oriented programming systems languages and applications companion (OOPSLA '11). ACM, New York, NY, USA, pp. 301-312. Website at http://kripken.github.io/emscripten-site/ (2011).
189. **Denton D**., "*Electronics for Guitarists*", Springer Science & Business Media. (typ. Page 247). (2013).
190. **Barbour, E**. The cool sound of tubes [vacuum tube musical applications]. IEEE Spectrum, 35(8), 24-35. (1998).
191. **Randall, A.**, *Proper Dummy Load on Output of Tube Amp*, https://tinyurl.com/myxo9k3 (1998).
192. **Adenot P.** *Web Audio API performance and debugging notes*, 2nd WebAudio Conference, Atlanta, 2016. http://padenot.github.io/web-audio-perf/#latency (2016).
193. **D. Difilippo and K. Greenebaum**. Audio Anecdotes I: Tools, Tips, and Techniques for Digital Audio: v. 1. Book. (2004).
194. **Barbour, E.** The cool sound of tubes [vacuum tube musical applications]. IEEE Spectrum, 35(8), 24-35. (1998).
195. **Kuehnel, R.** Circuit Analysis of a Legendary Tube Amplifier: The Fender Bassman 5F6-A. Pentode Press. (2005).
196. **Choi, H.** *AudioWorklet: the future of web audio*. Conference ICMC 2018, Daegu, Korea. (2018).
197. **Letz, S., Orlarey, Y., and Fober, D.** *Compiling Faust Audio DSP Code to WebAssembly.* In Proc. 3rd Web Audio Conference (WAC 2017), London, UK. (2017).
198. **Buffa, M., Lebrun, J., Kleimola, J., Larkin, O., Letz, S. and Pellerin, G.** *WAP: Ideas for a Web Audio Plug-in Standard.* WebAudio Conference 2018, Berlin, Germany. (2018).
199. **Buffa, M.**, **Lebrun, J.** *WebAudio Virtual Tube Guitar Amps and Pedal Board Design.* WebAudio Conference 2018, Berlin. (2018).
200. **Buffa M., Ereteo, G. Gandon F.** *A Wiki on the Semantic Web*, book chapter in "Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications", Editors: Jorg Rech, Bjorn Decker, Eric Ras, Information Science Reference, ISBN: 9781599048772, p115-137, 2008
201. **Jillings, N. and al**. *Intelligent audio plug-in framework for the Web Audio API*. In Proc. 3rd Web Audio Conference (WAC 2017), London, UK. (2017).
202. **Choi, H. and Berger, J**WAAX: *Web Audio API eXtension*. In Proc. Int. Conf. New Interfaces for Musical Expression (NIME'13), Daejeon, Korea. (2013).
203. **Letz, S., Orlarey, Y., and Fober, D**. *FAUST Domain Specific Audio DSP Language Compiled to WebAssembly*. In Companion Proc. The Web Conference 2018 (WWW '18). Lyon, France. (April 23--27, 2018). DOI=https://doi.org/10.1145/3184558.3185970
204. **Kleimola J. and Campbell O**. *Native Web Audio API plugins*. 4th Web Audio Conference (WAC 2018), Berlin, Germany. (2018).
205. **Buffa, M., Lebrun, J., Kleimola, J., Larkin, O, Letz, S. and Pellerin G**. *"WAP: Ideas for a Web Audio Plug-in Standard"*. 4th Web Audio Conference (WAC 2018), Berlin, Germany. (2018).
206. **Zide M., Gandon, F., Faron-Zucker C.** *"Overlapping Community Detection and Temporal Analysis on Q&A Sites"*. Web Intelligence and Agent Systems (2016).
207. **Zide M., Gandon, F., Faron-Zucker C.**, Song, G. *"Detecting topics and overlapping communities in question and answer sites"*. Social Network Analysis and Mining 5(1): 27:1-27:17 (2015).
208. **Buffa, M. and Lebrun, J.** *"Guitarists will be happy: guitar tube amp simulators and FX pedal in a virtual pedal board, and more!"*, WebAudio Conference 2018. Demo Paper. Berlin. (2018).
209. **K.O. Arras, W. Burgard, and al.** *Robots in exhibitions.* In Proceedings of IROS 2002 IEEE/RSJ International. (2002).
210. **C.H. Park, E.S. Ryu, and A.M. Howard**. *Telerobotic haptic exploration in art galleries and museums for individuals with visual impairments*. IEEE transactions on Haptics, 8(3), (2015).
211. **W. Burgard, A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun**. *Experiences with an interactive museum tour-guide robot*. Artificial intelligence, 114(1), 1999.

212. **Y. Wang, N. Stash, L. Aroyo, L. Hollink, and G. Schreiber**. *Using semantic relations for content-based recommender systems in cultural heritage.* In Int. Conf. on Ontology Patterns, volume 516 of CEUR, (2009).

213. **Benouaret, I.** *Un système de recommandation sensible au contexte pour la visite de musée*. In CORIA, (2015).

214. **Gicquel, P.Y. and Lenne. D.** *Proximités sémantiques et contextuelles pour l'apprentissage informel: Application à la visite de musées*. In EIAH&IA 2013, (2013).

215. **Oppermann, R. and Specht, M**. *A context-sensitive nomadic information system as an exhibition guide. handheld & ubiq*. In Computing 2nd Int. Symp, (2000).

216. **El Ghali, A., Tifous, A., Buffa, M., Giboin, A., Gandon, F., and Dieng-Kuntz, R.**. *"Web-Based Learning Solutions for Communities of Practice"*, book chapter "Using Sweetwiki in Communities of Practice". IGI Global, 2009.

217. **Davoust, A., Skaf-Molli, H., Molli, P., Esfandiari, B., & Aslan, K**. *Distributed wikis: a survey*. Concurrency and Computation: Practice and Experience, 27(11), 2751-2777. (2015).

218. **Rahhal, C., Skaf-Molli, H., & Molli, P.** *Swooki: A peer-to-peer semantic wiki* (Doctoral dissertation, INRIA). (2008).

219. **Wilmering, T., Fazekas, G., & Sandler, M.** (2011, September). *Towards ontological representations of digital audio effects.* In Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11). (2011).

220. **Fazekas, G., Raimond, Y., & Sandler, M**. (2008, October). *A framework for producing rich musical metadata in creative music production*. In Audio Engineering Society Convention 125. Audio Engineering Society. (2008).

221. **Michel, F.**. *Integrating heterogeneous data sources in the Web of data*. Other [cs.OH]. Université Côte d'Azur, 2017. English. 〈NNT : 2017AZUR4002〉. 〈tel-01508602v3〉

222. **Valimaki, Vesa, et al**. *"Fifty years of artificial reverberation*." IEEE Transactions on Audio, Speech, and Language Processing 20.5 (2012): 1421-1448. (2012).

223. **Vroomen, J., & Keetels, M.** *Perception of intersensory synchrony: a tutorial review*. Attention, Perception, & Psychophysics, *72*(4), 871-884. (2010).