

Wiki et Web Sémantique

Michel Buffa^{1,2}, Guillaume Ereteo², Fabien Gandon²

¹ Mainline, Laboratoire I3S, Université de Nice, France
buffa@unice.fr

² Acacia, Inria, Sophia-Antipolis, France
Fabien.Gandon@sophia.inria.fr

Résumé : Le concept de wiki a dix ans cette année mais a connu récemment un grand succès grâce à Wikipedia. Cependant, dans le monde des intranets, plusieurs études ont prouvé que l'utilisation des wikis est sujette à discussion et l'acceptation de tels outils de collaboration ouverts et peu structurés n'est pas évidente. Différentes raisons peuvent expliquer ce manque d'acceptation : des raisons sociales (la culture de l'entreprise n'est pas compatible avec un tel système ouvert), mais également d'usage (le wiki n'est pas assez structuré pour y naviguer et pour trouver l'information appropriée, l'apprentissage du langage de rédaction du wiki est un frein aux contributions, etc.). Nous présentons ici SweetWiki¹, un moteur de wiki réconciliant deux tendances du futur Web : un Web augmenté par des métadonnées et un Web d'applications sociales où chaque utilisateur est un acteur et un fournisseur actif. Les utilisateurs du wiki peuvent librement entrer des tags sur ses pages et leurs constituants et un mécanisme d'auto-complétion suggère des tags compatibles existants. En annotant sémantiquement les ressources du wiki et en réifiant le modèle même du wiki, SweetWiki fournit des fonctionnalités de structuration, de raisonnement et de requêtes sur le wiki. Tagguer est facilité et motivé par l'affichage en temps réel du nombre de pages liées aux tags suggérés.

Mots-clés : wiki, web sémantique, application collaborative.

1 Introduction

Les wikis ont été conçus au milieu des années 90 en exploitant les technologies Web de l'époque c.-à-d. HTML, HTTP, CGIs et URL. Pour compenser le manque d'outils simples d'édition et de stockage, les Wikis ont développé des variantes de langages de marqueurs ou « WikiML » mais il n'existe aucun WikiML standard même si plusieurs efforts ont été entrepris. Une autre caractéristique des wikis réside dans l'utilisation des WikiMots² pour indiquer des liens hypertexte et dans la mise à disposition de mécanismes simples pour la gestion de version ou l'édition concurrente. L'idée de SweetWiki est de revisiter la logique de conception des Wikis, en tenant compte des nouveaux standards disponibles pour le Web onze ans après, afin d'adresser certaines des imperfections identifiées par les retours d'expérience.

¹ <http://www-sop.inria.fr/acacia/soft/sweetwiki.html>

² WikiMot : un mot composé d'au moins deux mots capitalisés. qui devient un lien hypertexte.

Après l'évaluation de plusieurs moteurs wiki nous avons décidé d'écrire un nouveau moteur parce que notre vision du wiki du futur n'était pas compatible avec ce que nous avons trouvé dans des wikis existants. Nous voulions:

- *reposer sur les standards du Web* : standards pour le format des pages wiki (XHTML), pour les macros incluses dans les pages (JSPX/XML), etc. ;
- *inclure un environnement pour structurer et interroger le wiki dans les pages elles-mêmes* : le wiki est articulé autour d'un moteur web sémantique qui nous permet d'utiliser des langages de description du web sémantique (RDFa³, RDF, RDFS, OWL) et le langage de requêtes SPARQL⁴ pour respectivement décrire et interroger le modèle du Wiki, les ontologies de domaine et les métadonnées indexant les pages et leur contenu ;
- *permettre de tagguer collectivement*, et d'inclure des ontologies externes ;
- *abandonner les dialectes de WikiML* employés et modifiés par la plupart des wikis ;
- *améliorer l'accès à l'information* : navigation par facettes et des outils de recherche ;
- *permettre l'édition de métadonnées* : en fusionnant dans une seule interface l'édition du contenu d'une page et de ses métadonnées ;
- *rendre les métadonnées publiques*: en permettant aux métadonnées d'être extraites et exploitées par d'autres applications.

Dans la section suivante nous proposons un état de l'art sur les wikis sémantiques. Puis dans la section 3 nous expliquons le positionnement original de SweetWiki, la manière dont il supporte de multiples ontologies, y compris de son propre modèle. Dans la section 4 nous expliquons comment il est possible de créer dans SweetWiki des « pages d'applications » contenant des requêtes. Dans la section 5 nous montrons comment une ontologie particulière, une « folksonomie⁵ », composées de mot-clés (des « tags ») que les utilisateurs utilisent pour annoter les pages, images, vidéos, peut être utilisée pour améliorer la navigation, la recherche, les fonctionnalités de notification et de veille. Enfin, nous concluons sur une discussion.

2 Wikis et Wikis sémantiques

Beaucoup de wikis sémantiques sont actuellement développés et nous pouvons distinguer les approches considérant « l'utilisation des wikis pour des ontologies » des approches considérant « l'utilisation des ontologies pour des wikis » ; très peu de moteurs fusionnent actuellement les deux approches. La plupart des projets en cours sur des wikis sémantiques entrent dans la première catégorie c.-à-d. qu'ils considèrent chacune des pages du wiki comme des concepts et les liens typés (dans le contenu de page) comme des propriétés. Dans ce modèle, appelé « Wikipologie » (Decker et al., 2005), le Wiki devient l'interface d'un système de maintenance d'une ontologie.

Un des premiers wikis à entrer dans cette catégorie est Platypus (Campanini et al., 2004) qui impose d'éditer séparément les métadonnées pour chaque page du wiki dans une deuxième page dédiée. Il permet l'édition de base d'une ontologie mais sans

³ <http://www.w3.org/2001/sw/BestPractices/HTML/2006-01-24-rdfa-primer>

⁴ <http://www.w3.org/TR/rdf-sparql-query/>

⁵ La folksonomie représente « le vocabulaire de la communauté des utilisateurs ».

contrôler la cohérence avec les annotations. Il ne permet pas de raisonnements et ne permet que des requêtes simples. Les métadonnées sont employées pour l'amélioration de la navigation mais les utilisateurs doivent osciller entre l'édition de la page et l'édition des annotations sémantiques puisque ces activités reposent sur deux éditeurs distincts. SHAWN (Aumueller, 2005) offrent des options semblables.

Rise (Decker et al., 2005) entre également dans la première catégorie : l'ontologie utilisée par la communauté est éditée par l'intermédiaire du Wiki lui-même et un ensemble de conventions de nommage est employé pour construire automatiquement l'ontologie à partir du contenu du Wiki. Un langage propriétaire est utilisé pour décrire les métadonnées et l'exportation en RDF est possible. L'information sémantique est utilisée pour la navigation et le contrôle de la cohérence. L'ontologie est reconstruite chaque nuit pour tenir compte des évolutions.

Rhizome (Souzis, 2005) utilise une version modifiée de WikiML (ZML) qui repose sur des conventions de formatage pour rendre les propriétés sémantiques explicites directement dans le contenu de page. Les pages sont sauvées en RDF et un autre éditeur peut être employé pour modifier le RDF directement. Ce dispositif met la structure du wiki à risque et Rhizome fournit donc des mécanismes d'autorisation précis et des outils de validation. Il n'est pas clairement dit comment les métadonnées améliorent le comportement du wiki ; il n'y a pas de recherche avancée et pas d'aide à la navigation. RDF-Wiki⁶ est semblable à Rhizome en ce qu'il permet l'export des annotations RDF pour le traitement externe.

Semantic MediaWiki (Krötsch et al., 2005) est basé sur MediaWiki. Ici, contrairement à Rise (Decker et al., 2005), typer des liens peut également être employé pour indiquer des attributs de la page. Par exemple, le texte `San Diego est une [[est un::ville]] située dans le sud-ouest de [[est localisé dans::La Californie]]` établit les faits « San Diego est une ville » et « San Diego est situé en Californie ». Tandis que le texte `ses coordonnées sont [[coordonnées:=32°42'54"N,117°09'45"W]]` définit un attribut « coordonnées ». Ces données sont employées pour la navigation par facettes. Semantic MediaWiki traduit ces métadonnées en RDF et le moteur de raisonnement KAON2 peut être utilisé. D'autres extensions sémantiques de MediaWiki sont encore en développement (Muljadi et Takeda, 2005).

Makna (Dello et al., 2005) est basé sur JSPWiki et fournit des extensions sémantiques basées sur des liens typés. Il repose sur JENA pour permettre des requêtes complexes. Son éditeur textuel propose des formulaires avec auto-complétion pour questionner le moteur JENA et rechercher des concepts/propriétés/reliations ; ceci est utile en particulier lorsque l'ontologie est grande.

WikSar (Aumueller et Auer, 2005), (Aumueller, 2005) permet à des utilisateurs d'écrire des annotations sémantiques dans l'éditeur textuel du wiki en utilisant des WikiMots. Par exemple : si une page appelée « PrinceHamlet » contient une ligne « `PersonnageDe : WilliamShakespeare` », ceci sera traduit par un triplet en RDF. En combinant tous ces triplets, une ontologie peut être extraite à partir du contenu du Wiki. L'éditeur est uniquement textuel et propose ni aide ni contrôle de cohérence.

⁶ <http://infomesh.net/2001/05/sw/#rdfwiki>

Lorsque les pages sont sauveées, les métadonnées sont utilisées pour proposer une navigation par facettes. WikSar permet des requêtes complexes, et celles-ci peuvent être incluses dans les pages du wiki ou dans leurs modèles. Un outil de navigation graphique supplémentaire peut être utilisé pour explorer le wiki par son réseau des métadonnées.

Les liens typés sont rapides mais l'utilisateur doit se rappeler chaque concept, relation, et propriété qu'il veut utiliser et ceci n'est pas évident à l'usage. Avec AceWiki⁷ on peut ajouter et modifier des phrases écrites en utilisant le langage ACE⁸ (un anglais contrôlé), à l'aide d'un éditeur interactif. Cet éditeur prend en compte l'ontologie et fournit des conseils à l'utilisateur en proposant des liens valides lors de la frappe. L'éditeur peut aussi être utilisé pour étendre l'ontologie en créant de nouveaux concepts, rôles et individus.

La deuxième famille des approches se concentre sur « l'utilisation des ontologies pour les wikis ». IkeWiki (Schaffert et al., 2005) combine WikiML et édition WYSIWYG du contenu et des métadonnées des pages. L'éditeur fournit certains dispositifs dynamiques comme l'auto-complétion sur les métadonnées. Il suppose qu'une ontologie existante a été chargée. Il fournit aussi une aide à l'édition de l'ontologie. Il emploie JENA comme un moteur de raisonnement et les métadonnées sont utilisées pour la navigation et le rendu des pages. Des annotations peuvent être visualisées dans un cartouche à côté de la page et IkeWiki propose une interface utilisateur très agréable à l'usage.

SweetWiki, notre prototype, entre également dans cette deuxième catégorie. Il ne suit pas le modèle Wikitology mais nous avons fait le nécessaire pour permettre une telle évolution dans le futur. Nous permettons de tagguer des pages (social tagging), les tags générant une folksonomie; nous permettons aussi l'utilisation d'ontologies externes. SweetWiki est proche de WikSar puisque ces deux moteurs wiki partagent beaucoup de dispositifs comme la construction d'une ontologie par l'usage, l'inclusion de requêtes dans les pages, l'édition des métadonnées et du contenu de la page dans la même interface. SweetWiki ajoute un moteur de raisonnement et un éditeur WYSIWYG extensible pour le contenu et les métadonnées, (comme IkeWiki ou Makna). Nous avons écarté l'utilisation de WikiML car nous avons précédemment rencontré beaucoup de problèmes pendant la traduction entre le WikiML et XHTML. L'éditeur de SweetWiki utilise AJAX et l'annotation des pages est motivée auprès des utilisateurs par le fait que : (a) ils peuvent voir un affichage instantané des liens / facettes que l'annotation ajoutera à leur page ; (b) un mécanisme d'auto-complétion suggère des concepts existants de l'ontologie, les catégories relatives et le nombre de pages utilisant cette annotation comme une incitation à réutiliser les tags existants. En outre, SweetWiki vient avec des outils de gestion et d'édition des ontologies pour les tâches de maintenance. Cependant, SweetWiki n'est pas dédié à la gestion collaborative d'une ontologie comme peut l'être par exemple OntoWiki (Hepp et al., 2005).

⁷ <http://gopubmed.biotec.tu-dresden.de/AceWiki/>

⁸ Attempt to Controlled English (ACE) www.ifi.unizh.ch/attempto/

3 Le principe de SweetWiki

Les Wikis sont des sites Web où des pages sont organisées autour de WikiMots et parfois d'autres concepts tels que les WikiWebs. Pour aller au delà de cette structure informelle d'hyperliens, nous proposons d'utiliser l'annotation sémantique et des fonctionnalités de restructuration. Pour rendre explicite, manipuler et exploiter une telle structure nous avons conçu SweetWiki autour d'une « ontologie de wiki » qui décrit les concepts des wiki eux-mêmes. SweetWiki est également capable de profiter des ontologies externes et des folksonomies développées par ses utilisateurs à travers l'annotation qu'ils font des pages (Gruber 2005, Passant 2007, Newman 2007).

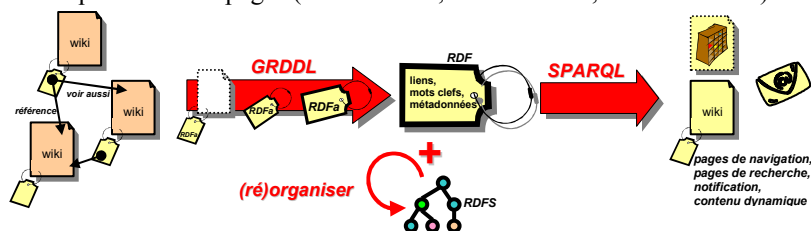


Fig. 1 – Schéma du principe de SweetWiki

SweetWiki se base sur ce que nous avons appelé « le modèle objet d'un Wiki », en d'autres termes : une ontologie de la structure d'un wiki. Dans beaucoup de moteurs de wiki, y compris plusieurs « wikis sémantiques », les concepts du wiki sont implicites à l'implantation. Poursuivant l'objectif des ontologies informatiques (Mizoguchi et al, 1997) nous les avons rendus explicites. On peut se demander en quoi ceci relève de l'ingénierie des connaissances. Avant tout, le domaine de modélisation n'est pas forcément le domaine d'application. En rendant le modèle de SweetWiki explicite et déclaratif, nous ne sommes pas en train de « reprogrammer » le logiciel. Le langage de programmation utilisé pour la programmation de SweetWiki n'est pas le langage de description ; par contre ce dernier permet de raisonner sur le logiciel.

Dans notre cas nous avons défini en OWL Lite tous les concepts, les propriétés et les relations que nous employons dans le wiki lui-même. Des primitives comme « document », « page », « tag », « lien en avant », « lien en arrière », « contributeur », « version », « fichier joint », « image jointe », etc. sont décrites dans cette ontologie. Les métadonnées générées pour chaque page avec ces primitives, sont incluses dans les pages wiki elles-mêmes. Il y a plusieurs avantages à employer un modèle déclaratif du wiki : nous pouvons raisonner sur ce modèle (ex: règles de production pour compléter les données, déclaration de relations inverses pour maintenir dualité entre lien en avant et lien en arrière, etc.), nous pouvons l'étendre et le réorganiser (ex: restructuration du wiki), nous pouvons construire au-dessus (ex: interopérabilité entre plusieurs wikis), nous pouvons concevoir des éléments d'interface pour aider la navigation (ex: liste des pages connexes), etc. En reposant sur RDF, le langage SPARQL peut être employé pour interroger toutes ces métadonnées localement ou à distance. L'ontologie de la structure des wiki est maintenue par les concepteurs du wiki. Considérons par exemple, la requête :

```
PREFIX wiki:http://www.essi.fr/sweetTwiki/wiki.rdfs#
select ?doc ?type ?page group ?doc distinct display xml
where { ?page wiki:includeDocument ?doc . ?doc rdf:type ?type }
```

Le résultat est l'ensemble des éléments inclus dans une page, ainsi que leur type. Dans l'ontologie du wiki, nous avons défini qu'une page pouvait inclure des documents attachés et qu'une image, un fichier ou une vidéo pouvaient être des documents attachés. La requête citée en exemple affiche donc tous les fichiers, images, vidéos inclus dans des pages. On voit ici l'intérêt d'un tel modèle déclaratif et compatible avec un langage de requête. Il va de soi qu'il serait souhaitable que tous les wikis sémantiques reposent sur des modèles communs. Des efforts tels que le framework SIOC (Semantically-interlinked Online Communities)⁹ pour l'aspect communautaire ou WikiOnt (Harth, Gassert 2005) pour la structure du wiki ou FOAF¹⁰ pour la description des personnes (et utilisé ici) sont intéressants à ce titre.

SweetWiki permet de taguer collectivement les pages et ressources du wiki. Les pages et les documents joints (images, fichiers attachés) peuvent être annotés dans l'éditeur. Les tags sont des mots-clefs entrés par l'utilisateur et collectés pour former une folksonomie. Afin d'améliorer la navigation tout en maintenant la simplicité native des wikis, nous avons mis en application ce mécanisme d'étiquette/tags/mot-clé avec une ontologie de domaine commune à tout le wiki. En rendant cette ontologie explicite (en RDFS) nous pouvons de nouveau raisonner dessus (ex: trouver des tags proches), nous pouvons faire des requêtes complexes (ex: grouper les pages en fonction de leurs tags ou des profils de leurs auteurs), nous pouvons modifier le wiki (ex: réorganiser l'ontologie, en fusionnant des concepts équivalents, en déclarant des liens hiérarchiques, en important des ontologies, etc.). L'ontologie de domaine est enrichie directement par les utilisateurs et peut être restructurée par des volontaires pour améliorer la navigation et les requêtes. SweetWiki vient avec un éditeur d'ontologies légères qui peut être employé par n'importe quel utilisateur afin d'organiser les tags.

D'autres ontologies peuvent être ajoutées pendant l'exécution et sont alors immédiatement accessibles aux utilisateurs pour taguer et faire des requêtes. Si le wiki est employé dans un domaine pour lequel des ontologies sont déjà disponibles, ces ontologies peuvent être chargées dans le serveur web sémantique de SweetWiki. C'est également une excellente manière d'amorcer la collection des tags et cela facilite l'interopérabilité.

L'exécution de SweetWiki repose sur le moteur de recherche sémantique CORESE pour les requêtes et les inférences (Corby et al., 2004) et sur SEWESE, une extension associée fournissant un serveur web, une API et des balises JSP pour construire des interfaces exploitant des ontologies et fournissant aussi des outils génériques (gestion de la sécurité, éditeur d'ontologie, cycle de vie, etc.). Le serveur se base sur une architecture standard d'application Web : des filtres contrôlent la session (ex: autorisation, profils d'utilisateur, etc.) et la mise en page (en-têtes, bas de page) ; les pages sont directement disponibles dans XHTML ou JSPX pour une lecture rapide ; un servlet gère les pages sauveées ; un ensemble de balises JSP fournissent des fonctionnalités (ex: soumettre une requête SPARQL) ; Des bibliothèques javascript implantent l'éditeur WYSIWYG.

⁹ SIOC: <http://sioc-project.org/>

¹⁰ <http://www.foaf-project.org/>

SweetWiki intègre un éditeur WYSIWYG de XHTML basé sur Kupu. Cet éditeur a été considérablement étendu afin de permettre:

- l'édition des métadonnées en parallèle avec l'édition du contenu pour décrire les pages et les objets qu'elles contiennent ;
- la validation et l'inclusion de requêtes dans une page éditée ;
- la création simple de liens vers d'autres pages;

4 Un wiki comme plateforme d'application

Des logiciels wikis tels que TWiki, JotSpot, Confluence ou encore XWiki sont appelés "*application wikis*" dans la mesure où le langage qu'ils proposent pour formater les pages, inclut de puissantes macros. Ces dernières permettent d'écrire, en éditant des pages du wiki, des « appliquettes ». Par exemple on peut définir un formulaire pour saisir des adresses et numéros de téléphone, afficher la liste des données saisies dans un tableau, faire des recherches, etc. La plupart du temps ces applications sont semblables à celle que l'on peut développer avec Excel, leur intérêt principal étant qu'elles sont en ligne, qu'elles ne nécessitent pas de base de données et qu'un utilisateur peut copier/coller une application écrite par un autre et en faire facilement une version personnalisée dans une autre page, etc.

Aucun des logiciels wikis cités n'utilise d'éditeur WYSIWYG lorsqu'il s'agit d'utiliser des macros dans les pages, ce qui rend l'édition relativement inconfortable. En effet, une étude (Buffa, 2006) a montré que seules quelques personnes développent ce type d'applications, néanmoins la somme de ces petites appliquettes finit par transformer le wiki en un puissant outil, mieux adapté aux besoins de ses utilisateurs que de gros logiciels génériques.

Dans SweetWiki il est possible d'inclure des requêtes dans des pages, sans quitter l'éditeur WYSIWYG. Ces requêtes peuvent « interroger » n'importe quelle « partie » du wiki en se basant sur le modèle du Wiki. On peut facilement afficher toutes les pages qui contiennent une vidéo insérée par l'utilisateur John Smith au cours de la semaine passée, ou encore afficher tous les types de vidéos supportés. Les données sont interrogeables aussi bien que le modèle lui-même. Un utilisateur de SweetWiki peut ainsi créer des « pages d'application » en incluant des requêtes complexes dans des pages. L'éditeur WYSIWYG propose même de tester les requêtes sans quitter le mode d'édition.

Regardons comme exemple d'application la gestion des profils utilisateurs. Lorsqu'un utilisateur s'enregistre dans le wiki pour la première fois, une page personnelle est automatiquement créée. Les métadonnées de cette page sont légèrement différentes de celles des autres pages : elles décrivent le profil de l'utilisateur. Dans la version actuelle, un utilisateur peut indiquer (en éditant cette page) une liste de tags qui correspond aux sujets qui l'intéressent. Par exemple, un utilisateur qui indique qu'il est intéressé par les « wikis » se verra proposer une liste de liens vers des pages ou des images. Cette liste est affichée dans sa page personnelle mais peut lui être envoyée par mail ou être exportée via un flux RSS. Elle est le résultat d'une requête qui est insérée automatiquement dans la page. Il est intéressant de remarquer que ces requêtes tiennent compte du polymorphisme et des relations

d'héritage qui peuvent exister entre les classes des métadonnées manipulées (ici les tags) ; dans l'exemple du wiki organisant des cours Java (Figure 3) si des pages ou des images sont tagguées avec "abstract class" elles seront également proposées comme résultat d'une recherche sur "class" et comme "interface" est une notion sœur des recherches sur ce tag seront suggérées.

5 Du web sémantique pour implémenter une folksonomie

SweetWiki supporte le "tagging social". Les utilisateurs peuvent tagguer des pages, des images, des vidéos, etc.

Comme (Bird, 2005) et (Olsen, 2005) nous proposons une approche mixte afin "d'organiser les tags" : nous relient les tags entre eux au sein d'une folksonomie (vocabulaire émergent d'une communauté) décrite à l'aide des langages du web sémantique (Gruber 2005) et (Passant 2007).. Les tags sont ainsi organisés dans une hiérarchie et reliés les uns aux autres à l'aide de relations telles `subClassOf`, `seeAlso`, etc. Nous pensons que le tagging social minimise l'effort et maximise la participation des utilisateurs. Des sites comme `del.icio.us`, `flicker`, `youtube` remportent un grand succès et utilisent massivement les tags ; SweetWiki propose également de tagger pages, images... cependant nous proposons une amélioration : nous avons implémenté un système qui permet à n'importe quel utilisateur d'organiser ces tags. Ce sont les utilisateurs qui sont responsables de la création et de l'organisation de cette « ontologie-utilisateur ». Nous avons montré que cette organisation permet d'améliorer la recherche et la navigation (Powers, 2005), (Hammond et al., 2005). Même si un très faible pourcentage d'utilisateurs utilise cette possibilité de réorganisation, l'ensemble des utilisateurs bénéficie de chaque amélioration. Des expériences ont également été menées chez EDF (Passant 2007) et ont confirmé l'intérêt d'une telle approche (même si dans cette expérience, seuls des administrateurs pouvaient mettre à jour la folksonomie). L'utilisation des technologies et des langages standards du web sémantique pour organiser et maintenir la folksonomie permet une grande souplesse dans son exploitation et sa mise à disposition pour d'autres applications.

Comme l'illustre la figure 2, lors de l'édition d'un document, un utilisateur peut librement saisir des mots-clés dans un champ textuel. Pendant la saisie, un mécanisme d'auto-complétion propose des tags existants en envoyant des requêtes SPARQL au moteur de recherche sémantique (nous utilisons ici la technologie AJAX). Ces requêtes permettent d'identifier les concepts existants dans la folksonomie dont les labels sont compatibles avec les caractères saisis. Pour chaque suggestion, la catégorie mère (si elle existe) est suggérée (ceci permet éventuellement de lever des ambiguïtés d'homonymie) et le nombre de ressources déjà tagguées est indiqué, incitant l'utilisateur à rejoindre un tag déjà présent dans la folksonomie.

Les nouveaux mots-clés sont collectés comme labels (termes) de nouvelles classes attachées par défaut à la catégorie « nouveaux concepts ». Ces classes viennent enrichir la folksonomie « au kilomètre » et plus tard, éventuellement, des utilisateurs peuvent les repositionner dans l'ontologie, les éditer, ajouter des labels dans d'autres langues, créer des relations de synonymie, fusionner des classes, etc. Il suffit qu'une personne fasse une modification pour que toute la communauté en bénéficie. Dans

tous les cas le feedback provenant de ces tags est important pour découvrir le vocabulaire de la communauté et son organisation incrémentale, par les utilisateurs, garantit une bonne adéquation avec leurs besoins de classification.

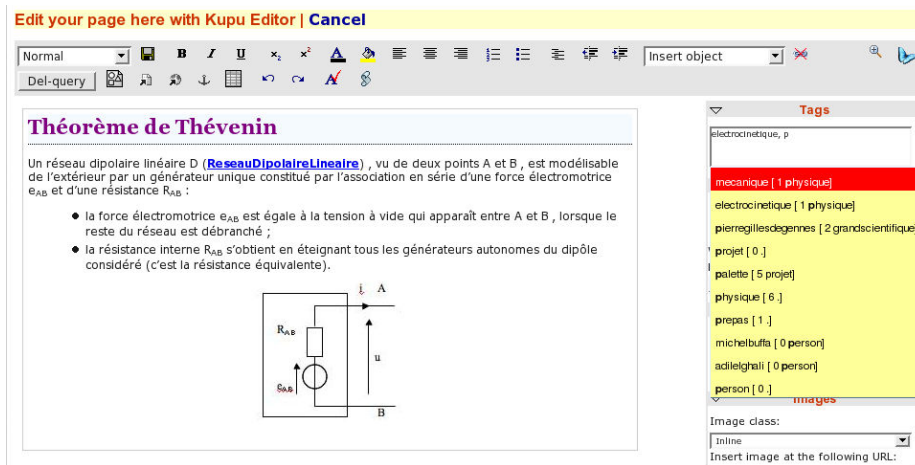


Fig. 2 – Editeur WYSIWYG et tags pour la page – Communauté ePre du projet IST Palette

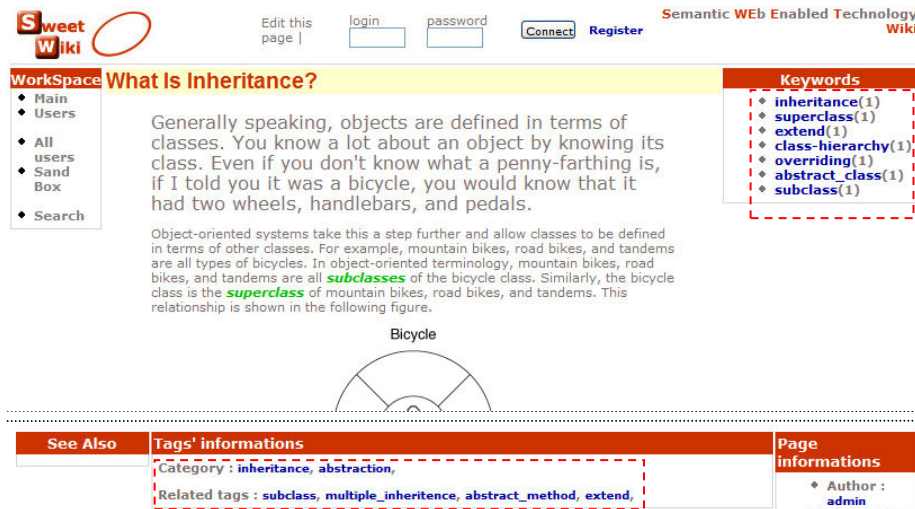


Fig. 3 – Visualisation de la page, de ses tags et suggestions

Lors de la visualisation d'une page (Figure 3), le moteur CORESE est utilisé pour générer des widgets de navigation par facette: la sémantique des tags de chaque page est utilisée pour suggérer des pages « voisines », des catégories « voisines », pour prévenir les utilisateurs intéressés par le sujet dont traite la page, etc. A chaque fois il s'agit de requêtes SPARQL insérées dans les pages du wiki. Ces inférences ne s'appliquent évidemment pas aux classes catégorisées sous « nouveaux concepts ».

6 Discussion

Avec SweetWiki, nous avons proposé une approche permettant aux utilisateurs d'éditer des pages du wiki et de les tagger tout en utilisant en coulisse les représentations du Web sémantique. En outre, des membres de la communauté, des "volontaires du wiki", peuvent consulter le modèle ainsi construit, examiner les tags proposés par les utilisateurs et les réorganiser. Lors de chacune des interventions de ce type, les annotations des utilisateurs demeurent inchangées mais la navigation par facette et la recherche toutes deux basées sur des requêtes sémantiques se retrouvent améliorées en proposant de nouveaux liens.

SweetWiki est disponible en ligne et il est utilisé par plusieurs projets nécessitant une plateforme de partage d'information entre participants. Ces communautés l'utilisent pour coordonner leur travail et forment un ensemble très hétérogène quant au niveau technique de chacun (on retrouve des chercheurs, des enseignants du secondaire, des secrétaires, etc.) :

- Palette : Il s'agit d'un projet Européen ayant pour sujet l'étude d'outils et de méthodes pour assister l'émergence de communautés d'intérêt et de pratiques ; par exemple le site WikiPrepas¹¹ est utilisé par la communauté ePrep¹² regroupant des enseignants en CPGE ayant pour but de construire un dictionnaire de mathématiques, physique et chimie pour les classes préparatoires aux grandes écoles scientifiques.
- e-Wok¹³ : projet national concernant les techniques de capture du CO₂, il implique la collaboration entre des instituts de géologie et des instituts pétroliers;
- Tests informels : nous conduisons plusieurs expérimentations informelles avec des élèves d'écoles primaires et des instituteurs, des étudiants du supérieur, des enseignants, des secrétaires, qui l'utilisent pour des tâches quotidiennes.

Des utilisateurs occasionnels, de même que des experts peuvent utiliser le même outil pour écrire de petites « appliquettes » ou pour améliorer la folksonomie en utilisant l'éditeur d'ontologie embarqué. Le support pour des ontologies externes ainsi que la possibilité d'insérer du contenu dynamique directement dans les pages, par exemple des requêtes SPARQL, font de SweetWiki une « plateforme de développement d'application web sémantiques ». Même des concepts au cœur du wiki tels que les pages, liens, images, auteurs, etc. peuvent être étendus et sont accessibles par requêtes. Nous avons récemment ajouté de nouvelles fonctionnalités comme le support pour l'insertion de vidéos dans les pages ou l'ajout de contexte sur les tags (qui taggue, où, quand, comment ?) et il a suffi de modifier quelques fichiers XML de manière déclarative. Nous n'avons pas eu besoin de modifier le code du logiciel, aussitôt les nouveaux concepts sont devenus accessibles via des requêtes (quelles sont les vidéos tagguées avec « wiki »).

Une extension évaluée actuellement porte sur le profil des utilisateurs. Nous l'avons vu, les utilisateurs peuvent indiquer leurs centres d'intérêt dans leur page personnelle et tagger les pages librement. En traçant l'historique des tags et leur

¹¹ <http://argentera.inria.fr/wikiprepas>

¹² <http://www.eprep.org>

¹³ <http://argentera.inria.fr/>

contexte d'utilisation (qui a taggué une ressource, quelles ressources ont été tagguées, quand un tag a-t-il été ajouté ou supprimé, etc.) il est possible de former un réseau d'accointance entre les utilisateurs. Ainsi, nous avons implémentés de nouveaux outils de recherche et de navigation permettant :

- de trouver les personnes les plus actives sur un tag : cette facette de navigation permet de trouver les personnes qui ont le plus contribué à un sujet.
- d'établir des similarités de comportement entre les utilisateurs : cette fonctionnalité permet de trouver les auteurs qui ont des activités similaires et des centres d'intérêts communs.
- d'établir les relations construites entre les tags au cours de l'évolution du wiki : ce critère de navigation permet de retrouver rapidement les tags qui sont souvent associés et ainsi mieux se documenter sur le contexte relié à un sujet.

Un certain nombre d'autres évolutions sont actuellement envisagées:

- Traitement de la langue naturelle pour tagguer automatiquement: plusieurs wikis commencent à analyser le texte des pages pour suggérer des tags potentiels. La suggestion de métadonnées pourrait utiliser des traitements de la langue naturelle (semi-)automatiques ; même des approches légères (shallow parsing) peuvent améliorer l'utilisation.
- Gestion de versions plus complète: actuellement nous supportons la gestion de versions des pages, mais pas des ontologies sous-jacentes.
- Maintenance collaborative de la folksonomie: fournir des outils wiki pour aider les utilisateurs à gérer le cycle de vie des ontologies et proposer des pages de discussion relatives à ces modifications, garder trace de tout ce qui a été modifié sur les ontologies, etc.

Ce dernier point nous ramène à la vision à double face des wikis pour des ontologies et des ontologies pour les wikis. Cette division des approches actuelles est symptomatique de l'émergence des wikis sémantiques ; il s'agit en effet d'un concept récent. A plus ou moins long terme, il est probable que les deux approches vont fusionner ; plusieurs projets ont déjà commencé à travailler dans ce sens ; l'objectif étant de transformer cette vision à deux faces en un cercle vertueux où les utilisateurs maintiendront le wiki et les ontologies sous-jacentes en même temps, sans aucune distinction. Nous avons conçu SweetWiki comme une plateforme d'expérimentation pour tester cette vision.

Actuellement, parmi les tests que nous menons, différentes expériences sont suivies : dans certains cas, il nous est demandé de verrouiller une partie de la folksonomie une fois que la classification est satisfaisante (cas de WikiPrepas, les enseignants contributeurs ayant la crainte que la folksonomie ne « dérive »), dans un autre cas (celui d'une entreprise de robotique utilisant SweetWiki pour son intranet et pour son site public communautaire) nous avons décidé de ne pas signaler la présence d'un éditeur de folksonomie. Dans quelques mois nous regarderons le vocabulaire qui aura émergé et commencerons avec l'aide de quelques volontaires à l'organiser. Nous observerons alors la réaction des utilisateurs face à la réorganisation induite dans le wiki (ex: nouveaux liens suggérés). Dans tous les cas, les folksonomies qui sont construites sont des « ontologies légères » ou des représentations intermédiaires utilisables comme point de départ pour générer des ontologies plus formelles.

Références

- AUMUELLER D. (2005). SHAWN: Structure Helps a Wiki Navigate. *Proc. BTW-Workshop*, W. Mueller, R. Schenkel editors. <http://dbs.uni-leipzig.de/~david/2005/aumueller05shawn.pdf>
- AUMUELLER D. & AUER S. (2005) Towards a Semantic Wiki Experience – Desktop Integration and Interactivity in WikSAR., *Workshop Semantic Desktop*, Galway, Ireland, Nov. 6th, 2005.
- BIRD. F. (2005) Some ideas to improve tags use in social software, flat hierarchy versus categories in social software.
- BUFFA, M. (2006). Intranet Wikis. *Intraweb workshop, WWW Conference*, Edinburgh.
- BUFFA M., GANDON F., (2006) SweetWiki : Semantic Web Enabled Technologies in Wiki, ACM Conference Wikisym, Odense.
- CAMPANINI S.E., CASTAGNA P., TAZZOLI R. (2004) Platypus Wiki: a Semantic Wiki Wiki Web. Semantic Web Applications and Perspectives, *Semantic Web Workshop*, 2004.
- CHAT C. & NAHABOO, C. (2006). Let's Build an Intranet at ILOG Like the Internet! *Intraweb workshop, WWW Conference*, Edinburgh.
- CORBY O., DIENG-KUNTZ R, FARON-ZUCKER, C., (2004) Querying the Semantic Web with the CORESE search engine. In Proc. of the 16th ECAI, Valencia, IOS Press, p. 705-709
- DECKER B., RAS E., RECH J., KLEIN B. AND HOECHT C. (2005) Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. *Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE)*, ISWC Galway
- DELLO K., TOLKSDORF R., PASLARU E. (2005) Makna. Free University of Berlin. <http://www.apps.ag-nbi.de/makna/wiki/About>
- HAMMOND, T., HANNAY T., LUND, B., SCOTT, J. (2005). Social Bookmarking Tools, a General Review. *D-Lib Magazine*, April 2005, Volume 11 Number 4,
- HEPP M., BACHLECHNER D., SIORPAES K. (2005) OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis. *Proc International Symposium on Wikis*
- KRÖTSCH M., VRANDEČIĆ D., VÖLKE M. (2005) Wikipedia and the Semantic Web - The Missing Links, *WikiMania*.
- MIZOGUCHI R., IKEDA M., SINITSA K. (1997) Roles of Shared Ontology in AI-ED Research, Intelligence, Conceptualization, Standardization, and Reusability, AIED-97, pp.537-544,
- MAJCHRZAC A., WAGNER C., YATES D. (2006) Corporate Wiki Users: Results of a Survey, *ACM Conference Wikisym*, Odense Denmark.
- MULJADI H., TAKEDA H. (2005) Semantic Wiki as an Integrated Content and Metadata Management System. Proceedings of ISWC, Galway, Ireland.
- OLSEN. H. (2005). Navigation blindness, how to deal with the fact that people tend to ignore navigation tools. *The Interaction Designer's Coffee Break*. Issue 13, Q.
- POWERS, S. (2005) Cheap Eats at the Semantic Web Café. <http://weblog.burningbird.net/archives/2005/01/27/cheap-eats-at-the-semantic-web-cafe/>.
- Harth A., Gassert H. (2005), WikiOnt Vocabulary Specification, proc. of WikiMania 2005.
- SCHAFFERT S., GRUBER A., WESTENTHALER (2005) R.: A Semantic Wiki for Collaborative Knowledge Formation. In: *Semantics 2005*, Vienna, Austria. November 2005.
- SOUZIS. A. (2005) Building a Semantic Wiki. *EEE Intelligent Systems*, vol. 20, no. 5, pp. 87-91, September/October
- VÖLKE M., KRÖTSCH M., VRANDEČIĆ D., HALLER H., STUDER R. (2006) Semantic wikipedia. *WWW*. Edinburgh, p196
- GRUBER T. (2005) of Folksonomy: A Mash-up of Apples and Oranges. In: Invited keynote, the First on-Line conference on Metadata and Semantics Research (MTSR'05).
- PASSANT A. (2007). Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs: Theoretical background and corporate use-case. ICWSM 2007, Boulder
- NEWMAN R. (2007). Tag ontology design, <http://www.holygoat.co.uk/blog/entry/2005-03-23-2>