

## **SWooki: Un Wiki Sémantique sur réseau Pair-à-Pair**

Charbel Rahhal, Hala Skaf-Molli, Pascal Molli

► **To cite this version:**

Charbel Rahhal, Hala Skaf-Molli, Pascal Molli. SWooki: Un Wiki Sémantique sur réseau Pair-à-Pair. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, Lavoisier, 2009, 14 (1), pp.117-140. <<http://isi.revuesonline.com/article.jsp?articleId=12955>>. <10.3166/isi.14.1.117-140>. <inria-00432216>

**HAL Id: inria-00432216**

**<https://hal.inria.fr/inria-00432216>**

Submitted on 14 Nov 2009

**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.

---

# **SWooki: Un Wiki Sémantique sur réseau Pair-à-Pair**

**Charbel Rahhal — Hala Skaf-Molli — Pascal Molli**

*LORIA – INRIA Nancy-Grand Est, Nancy Université, France  
{Charbel.Rahal, skaf, molli}@loria.fr*

---

*RÉSUMÉ. Les systèmes wiki évoluent vers des wikis sémantiques ou vers des wikis pair-à-pair. Les wikis sémantiques intègrent des technologies du web sémantique afin d'améliorer leur structuration, la recherche et la navigation entre les pages. Les wikis pair-à-pair offrent un support pour l'édition massive, pour l'édition déconnectée et une collaboration ad-hoc. Un des défis majeurs dans la conception d'un wiki sémantique pair-à-pair combinant les deux est la fusion des pages wikis contenant des annotations sémantiques. Les algorithmes de fusion utilisés dans les wikis pair-à-pair ont été conçus pour des structures linéaires qui ne considèrent pas les annotations sémantiques. Dans ce papier, nous présentons SWooki le premier wiki sémantique pair-à-pair. Il combine les avantages des wikis sémantiques et des wikis pair-à-pair. Nous détaillons l'approche SWooki et son algorithme de fusion, nous insistons sur ses caractéristiques et ses fonctionnalités. Nous concluons par une présentation de son architecture et de son implémentation.*

*ABSTRACT. Wiki systems have evolved in two different ways : semantic wikis and peer to peer wikis. Semantic wikis integrate the semantic web technologies in order to improve their structure, the search and the navigation between pages. Peer-to-peer wikis offer a support for massive collaboration, off-line editing mode and an ad-hoc collaboration. The main challenge in designing a peer-to-peer semantic wiki combining both approaches is the merge of wiki pages embedding semantic annotations. Merging algorithms used in peer-to-peer wiki systems have been designed for linear text. They do not handle semantic data. In this paper, we present SWooki the first peer-to-peer semantic wiki. SWooki combines the advantages of both semantic wikis and peer-to-peer wikis. We detail SWooki approach and its merging algorithm, we insist on its characteristics and its functionalities. We conclude by presenting its architecture and its implementation.*

*MOTS-CLÉS : wiki sémantique, wiki pair-à-pair, algorithme de réplication optimiste, collaboration massive.*

*KEYWORDS: semantic wiki, P2P wiki, optimistic replication algorithm, massive collaboration.*

---

## 1. Introduction

Les dix dernières années ont vu un développement sans précédent des outils collaboratifs. Le plus célèbre d'entre eux est certainement le WikiWikiWeb. Sa simplicité d'utilisation a permis de rendre le Web vraiment éditable. Des millions d'utilisateurs ont pu alors produire des pages wikis traitant des domaines les plus variés. Le succès le plus significatif est sans aucun doute la Wikipedia. Cette encyclopédie libre a montré le potentiel de la collaboration de masse. Des millions de personnes ont oeuvré à présenter des connaissances, à les maintenir, à les améliorer, à les classer, à les traduire et à s'auto-organiser. Une expérience d'une telle ampleur est sans précédent.

De nombreux travaux académiques et industriels proposent des améliorations pour les systèmes wikis. Nous nous intéressons à deux d'entre elles : les wiki sémantiques et les wikis sur réseaux pair-à-pair.

– Les wiki sémantiques sont une nouvelle génération d'outils d'édition collaborative. Ils permettent d'éditer des pages wikis contenant des annotations sémantiques. Les utilisateurs peuvent alors collaborer non seulement pour écrire des pages wikis mais aussi dans l'annotation de ces pages. L'aspect sémantique permet de rendre le contenu d'un wiki 'compréhensible' par une machine. Généralement, ceci est réalisé par l'annotation des liens entre les pages. Ainsi, les liens deviennent typés. Par exemple, un lien hypertexte reliant Paris à France peut-être annoté par *est capitale de*. Les wiki sémantiques essaient de préserver les avantages des wiki classiques, c'est-à-dire la simplicité dans la création et dans la modification des pages. La "Semantic Wikipedia"(Völkel *et al.*, 2006b) est un exemple d'édition collaborative autour des wiki sémantiques. Les wiki sémantiques fournissent une meilleure structuration des wiki en offrant un moyen de naviguer et de rechercher les pages wiki en se basant sur les annotations qu'elles contiennent. Ces annotations expriment des métadonnées sur les pages wiki et les relations entre ces pages, elles sont écrites dans une syntaxe formalisée de sorte qu'elles soient traitées automatiquement par les machines et qu'elles soient exploitables par des requêtes sémantiques.

– Les wiki sur réseaux pair-à-pair tentent de concilier les avantages de l'édition collaborative de masse et les qualités intrinsèques des réseaux pair-à-pair. En effet, les systèmes wikis sont actuellement des systèmes centralisés qui passent mal à l'échelle et qui ne tolèrent pas les pannes. En plus, ils souffrent des problèmes du contrôle centralisé et de la censure. Un wiki réside sur un seul serveur d'une entreprise qui contrôle son fonctionnement. Ces problèmes sont bien résolus dans le monde des réseaux pair-à-pair(Androutsellis-Theotokis *et al.*, 2004). Il existe actuellement des propositions de recherche pour construire un wiki sur réseau pair-à-pair(Weiss *et al.*, 2007; Morris, 2007). Ces propositions sont de nature différentes : certaines proposent de répliquer entièrement les données sur chaque noeud du réseau. Il faut ensuite gérer la cohérence des différentes pages éditées de manière concurrente(Weiss *et al.*, 2007). D'autres proposent une réplication partielle des différentes pages wiki sur les pairs du réseau pair-à-pair.

Dans cet article, nous présentons SWooki le premier wiki sémantique pair-à-pair. SWooki combine les avantages des wikis sémantiques et des wikis pair-à-pair. Un des défis majeurs dans la réalisation d'un tel wiki est la fusion des pages wikis contenant des annotations sémantiques. Les algorithmes de fusion utilisés dans les wikis pair-à-pair ont été conçus pour des structures linéaires qui ne traitent pas les annotations sémantiques. L'article est organisé comme suit : la section 2 détaille des cas d'utilisation intéressants pour des wikis pair-à-pair. La section 3 présente l'état de l'art. Dans cette section, nous détaillons les wikis pair-à-pair disponibles, nous présentons des approches pour la synchronisation des métadonnées dans le web sémantique pair-à-pair et nous présentons les types d'approches des wikis sémantiques disponibles. Les algorithmes de fusion disponibles pour les wiki pair-à-pair et leur façon de traiter les éditions concurrentes sont détaillés au moyen d'un scénario dans la section 4. La section 5 présente l'approche générale de SWooki et son algorithme de fusion. Nous présentons l'architecture et l'implémentation de SWooki dans la section 6. La section 7 présente une analyse de performance de SWooki. La dernière section conclut le papier.

## 2. Cas d'utilisation pour des wikis pair-à-pair

Dans le cadre du projet Xwiki Concerto<sup>1</sup>, nous avons identifié trois cas d'utilisation intéressants pour les systèmes wiki pair-à-pair : **Une collaboration massive, une édition déconnectée et une édition collaborative Ad-hoc.**

- **Une collaboration massive.** Dans ce cas, un système de wiki pair-à-pair est déployé similairement au réseau Usenet (Spencer *et al.*, 1988). Par exemple, des milliers de serveurs wikis peuvent être déployés au sein des organisations ou des universités. Tout utilisateur peut se connecter à n'importe quel serveur wiki. Ce déploiement permet :

- Le traitement d'un grand nombre d'utilisateurs en divisant la charge sur l'ensemble du réseau.

- La tolérance aux pannes : La panne d'un serveur wiki n'arrête pas tout le fonctionnement du système.

- Le partage des coûts de l'infrastructure. Les wikis sont mis en place et maintenus par des différentes organisations. Par conséquent, il n'est pas nécessaire de recueillir des fondations pour maintenir cette infrastructure. Par exemple, la fondation Wikipedia doit collecter 150000\$ chaque trois mois seulement pour la maintenance de son infrastructure.

- La résistance à la censure : Une organisation contrôle seulement un serveur wiki et non pas la totalité de l'ensemble des données.

- **L'édition déconnectée.** L'ajout des capacités d'édition déconnectée à des applications web constitue actuellement une issue majeure. Les wikis sont des applica-

1. <http://concerto.xwiki.com/xwiki/bin/view/Main/WebHome>

tions web et le besoin d'éditer un wiki déconnecté est réel. Les technologies actuelles ajoutent la capacité du travail déconnecté aux applications web en se basant sur les applications Ajax. Dans ce contexte des outils ont été développés tels que Google gears (Gears, 2008) et Firefox3 déconnecté. Cependant, ce mode déconnecté dans ces applications web ne fournit pas toutes les fonctionnalités disponibles dans le mode en ligne. Cela peut constituer un obstacle pour un système Wiki. Par exemple, le mode déconnecté du wiki permet la navigation mais il ne permet pas l'édition. Un wiki pair-à-pair permet naturellement le travail déconnecté au moyen d'un algorithme de fusion intégré. Avec un tel système, il est possible de voyager avec un système de wiki complet placé sur un ordinateur portable, effectuer des modifications d'une façon déconnectée et re-synchroniser avec le réseau pair-à-pair dès qu'une connexion internet est disponible.

– **Une édition collaborative Ad-hoc.** Ce scénario dérive du scénario précédent. Supposons que plusieurs utilisateurs de wiki déconnecté participent à une réunion. Malheureusement, il n'existe pas de connexion internet disponible dans la salle de réunion. Par conséquent, les utilisateurs décident de créer un réseau ad-hoc dans la salle. Un wiki pair-à-pair est capable de propager des modifications dans le réseau ad-hoc et permet aussi une édition collaborative entre utilisateurs déconnectés. Bien sûr, lorsque la réunion s'achève et ces utilisateurs retournent dans leurs organisations, leur système wiki va re-synchroniser avec l'ensemble du réseau pair-à-pair.

Nous avons adopté ces cas d'utilisation dans la réalisation de notre wiki sémantique pair-à-pair SWooki.

### 3. État de l'art

Dans cette section, nous détaillons les wikis pair-à-pair disponibles. Nous étudions leur conformité aux cas d'utilisation ci-dessus. Ensuite nous présentons des approches pour la synchronisation des métadonnées dans le web sémantique pair-à-pair et nous étudions la possibilité de les utiliser dans le contexte d'un wiki sémantique pair-à-pair. Enfin nous présentons les types d'approches des wikis sémantiques disponibles.

#### 3.1. Wikis pair-à-pair

Dans la littérature, il y a trois wikis pair-à-pair disponibles : DistriWiki (Morris, 2007), Code Co-op (Co-op, 2006) et Wooki (Weiss *et al.*, 2007). L'idée de base de DistriWiki (Morris, 2007) est de répliquer les pages wiki sur différents pairs afin de réduire les coûts de l'infrastructure et de la limitation de la bande passante et de supporter la tolérance aux pannes. DistriWiki adopte une réplication partielle des pages wiki, c'est-à-dire un pair contient seulement un sous-ensemble de pages wiki et non pas la totalité des pages. Par conséquent, DistriWiki ne supporte pas les deux cas d'utilisation suivant : (1) le mode d'édition déconnectée et (2) la collaboration ad-hoc. Un pair déconnecté permet seulement d'éditer les pages wiki qu'il possède et de naviguer

entre les pages existantes sur ce pair. DistriWiki est basé sur JXTA(Oaks *et al.*, 2002) pour fournir des méthodes de recherche, de mise en cache et de découverte des pages wiki sur les différents pairs. Les pairs sont informés des pages wiki dans le réseau à travers des annonces nommées "publicités" contenant des métadonnées sur ces pages. Des pertes de mise à jour peuvent avoir lieu dans DistriWiki et la tolérance aux pannes n'est pas efficace. Supposons qu'un pair crée une page wiki et publie son advertisement, tout de suite ce pair tombe en panne sans que la page soit demandée par un autre pair. Le résultat est la perte des éditions faites par ce pair. DistriWiki ne traite pas le problème de l'édition concurrente et la résolution des conflits de versions est manuelle. Cette résolution de conflits des versions n'est pas compatible avec l'édition massive où les modifications concurrentes dans un wiki pair-à-pair sont fréquentes. DistriWiki a été conçu pour des pages wiki traditionnelles n'incluant pas des annotations sémantiques. Pour des pages wiki contenant des annotations sémantiques, ces annotations seront distribuées dans les pages des différents pairs. Dans ce cas, il n'est pas possible d'effectuer des requêtes sémantiques où les annotations sont dispersées sur les différents pairs. Par exemple, la recherche de tous les pays membres de l'Union Européenne sur un pair nécessite d'avoir toutes les pages wiki de ces pays sur ce pair ainsi que leurs annotations.

Code Co-op(Co-op, 2006) est un produit commercial. Ses auteurs affirment que Code Co-op offre les fonctionnalités d'un wiki pair-à-pair. Toutefois, il n'existe aucune publication scientifique décrivant les algorithmes de Co-op. Code Co-op se base sur un système de contrôle de versions distribué (DVCS) (Allen *et al.*, 1995) dans sa construction d'un wiki pair-à-pair. Les DVCS possèdent les caractéristiques pour répliquer des données textuelles. Ils ont été conçus pour le développement des logiciels distribués, il est évidemment possible d'utiliser Code Co-op en dehors de ce contexte. Toutefois, les systèmes DVCS n'ont jamais prétendu qu'ils assurent la convergence des répliques ou la préservation des intentions de l'utilisateur(Sun *et al.*, 1998). Dans ces systèmes, un site peut intégrer à tout moment une modification distante effectuée sur un autre site. Cette caractéristique est sûrement nécessaire pour construire un wiki pair-à-pair. Par contre, cette intégration est faite sous le contrôle d'un être humain. En cas de conflits générés par des modifications concurrentes, le système génère des blocs de conflits informant l'utilisateur local à propos de ces conflits. Cette stratégie est convenable dans le cas du génie logiciel. Dans un système wiki pair-à-pair le contexte change légèrement. Dans un tel système, un serveur wiki intègre les modifications distantes de ses voisins d'une façon continue. Si le mécanisme d'intégration génère des modifications au moment de l'intégration des modifications distantes, le système pair-à-pair peut se boucler infiniment. C'est pour cela, de nombreux DVCS bloquent l'espace du travail local en cas de conflit. Cela signifie que tous les noeuds du réseau pair-à-pair contenant des conflits ne peuvent pas traiter des demandes. Dans ce cas, le premier cas d'utilisation de l'édition massive n'est pas réalisable. L'approche de Code Co-op réplique actuellement des systèmes de fichiers et des fichiers de texte. Par conséquent, la fusion des annotations sémantiques sera faite sous forme de texte.

Wooki<sup>2</sup> (Weiss *et al.*, 2007) est un système wiki pair-à-pair composé d'un ensemble de serveurs wikis (pairs) interconnectés. Dans ce réseau, chaque pair joue le même rôle. Wooki est un réseau pair-à-pair non structuré, un pair Wooki peut joindre ou quitter le réseau à tout moment. Les pages wiki sont répliquées sur tous les pairs du réseau. Chaque pair héberge une copie des pages et peut offrir le service du wiki d'une façon autonome. Un mécanisme de réplication optimiste nommé WOOT (Oster *et al.*, 2006) assure la convergence des copies de pages wiki répliquées sur tous les sites. WOOT a été conçu pour la réplication des données linéaires. Il a été optimisé en WOOTO<sup>3</sup> afin de réduire sa complexité de  $O(N^3)$  en  $O(N^2)$  où  $N$  est le nombre de lignes par page, y compris les lignes supprimées. Une page wiki est considérée comme une séquence de lignes. WOOT respecte les principes de la communauté TCAO<sup>4</sup>, en assurant la convergence et la préservation des intentions des utilisateurs (Sun *et al.*, 1998). Dans un wiki traditionnel, une édition concurrente est détectée juste après la sauvegarde de la page par un utilisateur mais dans Wooki elle est détectée à posteriori c'est-à-dire lors de l'intégration des modifications concurrentes distantes provenant d'autres sites dans la page. Par conséquent, dans Wooki certaines pages sont produites automatiquement par un algorithme de fusion et leur contenu doit être relu par un humain. Wooki intègre un système de conscience de la concurrence (Alshattawi *et al.*, 2008) pour informer les utilisateurs de l'existence de ces pages wiki et de leur état. Wooki supporte tous les cas d'utilisation d'un système wiki pair-à-pair. L'algorithme de réplication de Wooki est conçu pour fusionner une structure de texte linéaire et non pas des données sémantiques.

### 3.2. Synchronisation des ressources

Les recherches dans le web sémantique pair-à-pair pour la synchronisation des ressources ou des métadonnées des ressources distribuées sur le réseau pair-à-pair ne sont pas orientées pour synchroniser ces ressources dans un contexte d'une édition collaborative. Par conséquent, elles ne sont pas adaptées pour la synchronisation des ressources dans un wiki sémantique pair-à-pair.

RDFSsync (Morbidoni *et al.*, 2007) est un algorithme de synchronisation des modèles RDF. RDF (RDF, 2004) est un modèle de métadonnées qui permet de décrire les ressources Web, à l'aide des triplets < sujet, prédicat, objet > où le sujet représente la ressource décrite, le prédicat représente une propriété appliquée à cette ressource et l'objet est une donnée (un attribut) ou une autre ressource. RDFSsync est basé sur la décomposition des modèles RDF en un ensemble de graphes minimaux MSG (Minimum Self-Contained Graph). Un modèle RDF est représenté par une liste de hachage de ses MSGs. La synchronisation de deux modèles RDF est faite par une évaluation et par une comparaison distante de leur liste ordonnée. RDFSsync synchronise un modèle

2. <http://wooki.sourceforge.net/>

3. Dans ce papier, WOOTO et WOOT sont utilisés d'une manière indifférenciée.

4. TCAO est l'acronyme de Travail Collaboratif Assisté par Ordinateur, qui correspond à l'anglais CSCW (Computer Supported Cooperative Work).



cible avec un modèle source. Il transforme le modèle cible en un modèle fusionnant les deux modèles cible et source. Tout d'abord RDFSyc calcule la différence entre les listes de hachage des modèles RDF source et cible. Ensuite, il envoie les MSGs existant seulement dans la source à la cible. RDFSyc ne peut pas être utilisé pour synchroniser des modèles RDF dans une architecture pair-à-pair où les modèles sont produits par collaboration puisqu'il est incrémental. Dans un wiki sémantique pair-à-pair répliquant les pages wiki et leurs annotations sémantiques sur tous les pairs, la synchronisation de ces annotations doit assurer leur convergence indépendamment de leur suppression et/ou de leur insertion. Par contre, la synchronisation dans RDFSyc permet toujours l'ajout des annotations, tandis que la suppression n'est possible que si les annotations sont supprimées de tous les modèles synchronisés. Par exemple, prenons un modèle RDF initial M contenant deux triplets A et B qui est le même sur deux pairs du réseau. Un utilisateur U1 supprime sur le premier pair le triplet A de M alors son modèle devient M0. Aucune modification n'est produite sur le deuxième pair. Supposons maintenant que M0 doit être synchronisé avec M qui est le modèle du deuxième pair qui n'a pas été modifié afin de converger. Le résultat de cette synchronisation est de nouveau M et la suppression de A n'a pas eu lieu sur les deux pairs. Une suppression d'un triplet du modèle RDF ne peut pas être effectuée tant que ce triplet n'est pas supprimé au même moment sur tous les pairs.

RDFGrowth(Tummarello *et al.*, 2004) est un algorithme pour des applications pair-à-pair échangeant des métadonnées. Les pairs participent dans des "groupes d'intérêt" pour accroître leurs connaissances sur un ou plusieurs "sujets". Dans ce réseau pair-à-pair, chaque pair a une base de données RDF locale et utilise l'algorithme RDFGrowth pour accroître ses connaissances internes par la découverte et par l'importation des ressources d'autres pairs. RDFGrowth synchronise un ensemble de URIs(URI, 2001) exprimant l'intérêt du groupe et les informations les concernant. Par exemple, un intérêt de groupe (IG) est de "trouver les albums d'un chanteur X". En utilisant RDFGrowth, toutes les informations concernant cette requête seront collectées et stockées sur les différents pairs intéressés par IG. RDFGrowth assure la convergence des bases de données RDF des pairs, mais il est seulement incrémental, c'est-à-dire un triplet RDF ou une information supprimé sur un pair ne sera pas supprimé sur les autres pairs. Par conséquent, RDFGrowth ne peut pas être utilisé pour synchroniser des métadonnées dans un wiki sémantique pair-à-pair où les triplets RDF qui représentent les annotations dans les pages wiki sont mises à jour fréquemment.

SemVersion(Völkel *et al.*, 2006a) fournit des versions sémantiques et structurales pour les modèles RDF et des langages d'ontologie basés sur RDF tel que RDFS. Il est inspiré par le système CVS (Control Version Systems) et il est basé sur une architecture centralisée. SemVersion n'est pas conçu pour une architecture pair-à-pair. SemVersion gère seulement des graphes RDF et n'est pas adapté pour synchroniser du texte c'est-à-dire le contenu des pages wiki. Il peut être utilisé dans un wiki sémantique pair-à-pair pour assurer la convergence des graphes RDF sur tous les sites mais dans ce cas il sera nécessaire d'utiliser un autre algorithme pour assurer la convergence du texte des pages wiki. En outre, des mécanismes de propagation et d'intégration des différences des versions sur tous les sites seront nécessaires.

### 3.3. Wikis sémantiques

Afin de combiner un système wiki pair-à-pair avec un système wiki sémantique, il est important de savoir comment les wikis sémantiques représentent leurs données sémantiques et la façon dont ils combinent les parties textuelles avec les parties sémantiques. De nombreux wikis sémantiques sont actuellement développés et nous pouvons distinguer deux types d'approches, (1) des approches considérant l'utilisation des wikis pour des ontologies et (2) des approches considérant l'utilisation des ontologies pour des wikis ; très peu de moteurs fusionnent actuellement les deux approches. Dans cet article nous nous basons sur les wikis sémantiques faisant parti du premier type d'approches (1), et plus précisément sur Semantic MediaWiki. Dans cette dernière, les wikis sémantiques (Völkel *et al.*, 2006b; Tazzoli *et al.*, 2004; Souzis, 2005) sont une combinaison directe des systèmes de wiki existants et des paradigmes de représentation des connaissances du web sémantique. Ces wikis n'ont pas été conçus comme un éditeur d'ontologie spécialisé et ils n'imposent aucune restriction sur les annotations sémantiques. Une ontologie formelle émerge au cours de l'édition des pages wiki. Dans ce modèle, appelé Wikitology (Decker *et al.*, 2005) le wiki devient l'interface d'un système de maintenance d'une ontologie. Les annotations sémantiques peuvent être intégrées soit directement dans le texte de la page wiki sous forme de liens typés, soit séparément dans une interface spécifique aux annotations sémantiques. Par exemple, dans Semantic MediaWiki, les annotations sémantiques sont intégrées directement dans le texte via un langage de balisage spécial comme c'est indiqué dans la figure 1. Chaque article représente une ressource. Une annotation dans un article est un fait sur cet article. Elle représente ses métadonnées et spécifie ses relations avec d'autres pages. La classification des pages est faite manuellement par les utilisateurs en utilisant l'annotation Catégorie. Cette localisation dans le texte assure une meilleure lisibilité des annotations.

```
SWooki est un
wiki [[diaplecte::MediaWiki]]
sémantique [[langage::RDF]]
pair-à-pair [[typeRéseau::Usenet]]
[[Catégorie:Wiki]]
[[Catégorie:Sémantique]]
[[Catégorie:P2P]]
```

**Figure 1.** Page de SWooki dans Semantic MediaWiki

Les wikis sémantiques appartenant au second type d'approches (2) peuvent être considérés comme des outils de génie d'ontologie. Ces wikis exigent de charger des ontologies existantes avant de les utiliser. Un exemple de wikis suivant cette approche est IkeWiki (Schaffert, 2006) et SweetWiki (Buffa *et al.*, 2006). Dans cet article, nous proposons de combiner Wooki avec les technologies du web sémantique d'une façon similaire à celle utilisée dans Semantic MediaWiki (Völkel *et al.*, 2006b). Nous appelons cette combinaison SWooki, un wiki sémantique pair-à-pair fournissant les mêmes

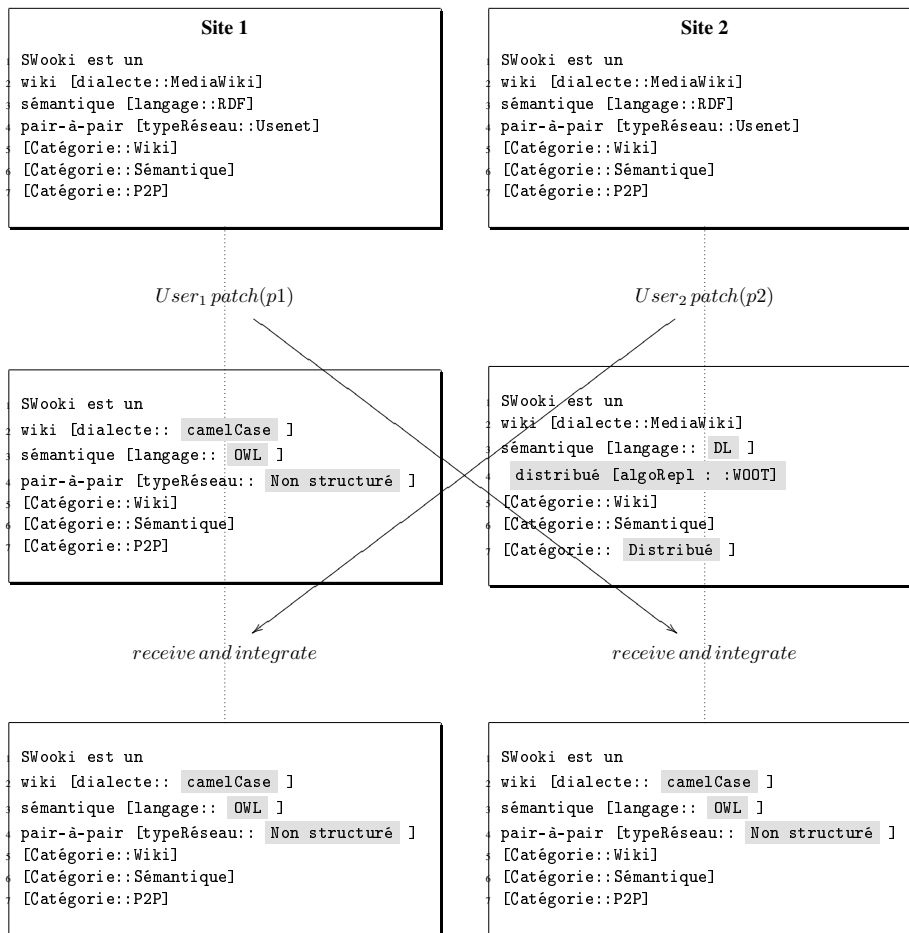
fonctionnalités de base de Semantic MediaWiki. En plus, SWooki assure une collaboration massive, l'édition déconnectée et l'édition Ad-hoc grâce à sa réplification totale des données et son algorithme de réplification.

#### 4. Algorithmes de fusion dans un wiki pair-à-pair

La réplification des pages wiki imbriquant des annotations sémantiques dans un wiki sémantique pair-à-pair nécessite un algorithme de réplification optimiste pour synchroniser les modifications dans ces pages et de leurs annotations et pour assurer leur convergence sur tous les pairs. Dans cette section, nous nous concentrons sur l'intégration des modifications concurrentes dans les wikis sémantiques pair-à-pair. Nous détaillons deux façons différentes pour gérer l'édition concurrente des pages wiki combinant des données textuelles et sémantiques. Nous appliquons WOOT et la règle de Thomas (Johnson *et al.*, 1975) puisqu'ils sont les seuls algorithmes de fusion disponibles pour les wikis pair-à-pair.

##### 4.1. La stratégie de la règle de Thomas

La stratégie de la règle de Thomas a été introduite en 1975 par Johnson et Thomas (Johnson *et al.*, 1975). Dans cette approche, l'algorithme de réplification assure une convergence éventuelle (Saito *et al.*, 2005) des données répliquées. Cet algorithme est conçu pour des systèmes de réplification à grande échelle. Chaque site maintient un ensemble de pairs (identifiant ; valeur). Ceci peut être facilement réalisé pour des données sémantiques telles que les triplets RDF (sujet ; prédicat ; objet). Chaque modification est affectée par deux estampilles, une estampille de création indiquant le temps de création de la modification utilisée dans la détection des données identiques et une estampille dernière modification indiquant le temps du dernier changement fait sur la modification. En cas des mises à jour concurrentes sur les mêmes données, le nouveau changement ne s'applique que si son estampille dernière modification est plus récent que celui de la copie locale. Dans la figure 2, nous appliquons la règle de Thomas pour fusionner des modifications concurrentes. En partant d'un texte initial, deux utilisateurs connectés chacun sur un site différent éditent simultanément en collaboration une page wiki sur SWooki. Cette page est considérée comme une paire (identifiantPage ; valeur). Sur le site 1, l'utilisateur1 met à jour les valeurs des propriétés suivantes : dialecte à camelCase, langage à OWL et typeRéseau à Non structurée. Toutes ces modifications sont marquées dans la figure 2. Le nouveau contenu de la page patch p1 est généré par les modifications faites par l'utilisateur1. P1 est exécuté localement sur le site1, puis envoyé avec son estampille dernière modification au site2 afin d'être intégré. Parallèlement au site2, l'utilisateur2 remplace la ligne 4 par une nouvelle ligne "distribué [algoRepl : :WOOT]". Il met à jour la valeur de la propriété langage à DL et la valeur de la Catégorie P2P à Distribué. Un patch p2 généré par les modifications faites par l'utilisateur2 est exécuté localement sur le site2 et envoyé avec son estampille dernière modification au site1 afin d'être intégré. Les deux patches



**Figure 2.** Édition concurrente avec la règle de Thomas

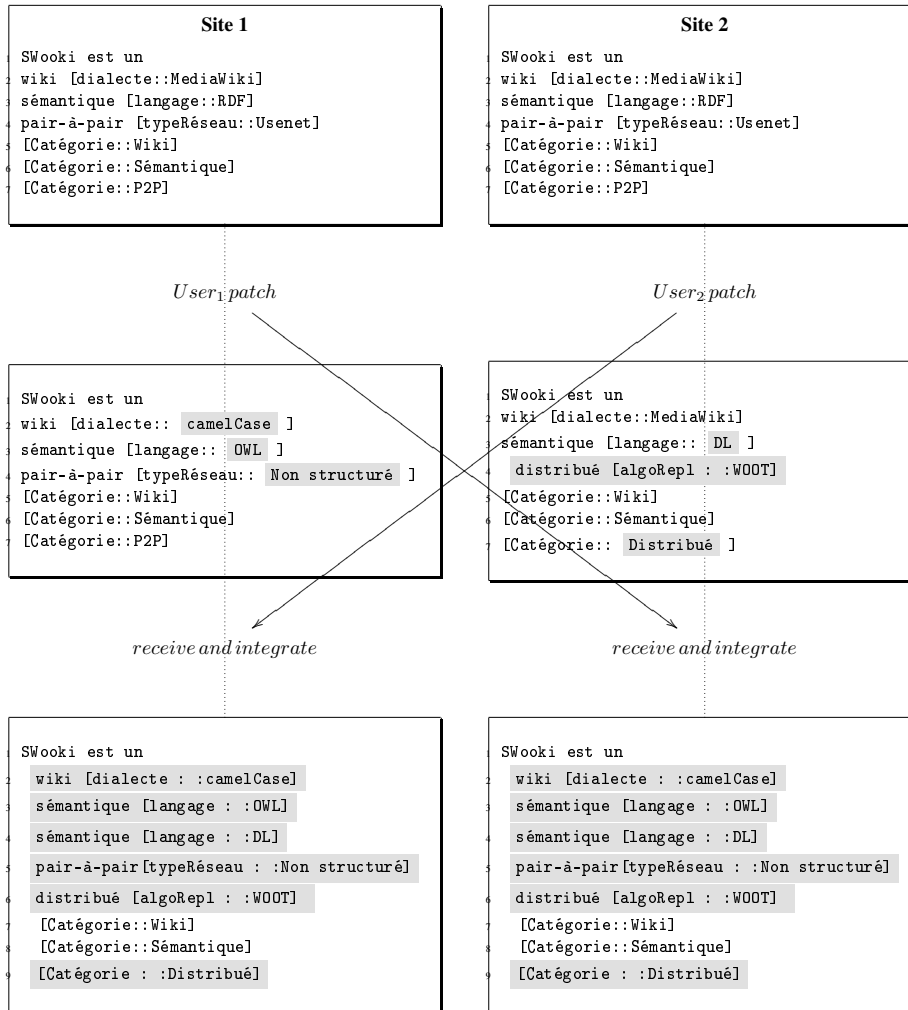
concurrents p1 et p2 (voir la figure 2) sont affectés par une estampille de création et une estampille dernière modification chacun. Dans notre scénario, nous supposons que p1 est plus récent que p2. Quand p2 est reçu sur le site1, il est tout simplement ignoré. Par contre, quand p1 est reçu sur le site 2, il est plus récent que la version courante de la page, il est donc appliqué sur le site 2. Cette stratégie assure une cohérence éventuelle et assure également que toute page visible sur un serveur wiki a été examinée par un humain. En fait, le problème principal de cette stratégie est que l'effet de p2 n'est pas visible sur l'état actuel de la page wiki i.e. les modifications faites par l'utilisateur2 sont perdues. Cela exige d'avoir un système de conscience de la concurrence pour informer les utilisateurs de l'existence d'une édition concurrente. Ce système n'existe pas actuellement pour la règle de Thomas. Les utilisateurs doivent être informés de la concurrence lorsqu'elle existe. Ils doivent fusionner manuellement les

modifications concurrentes en utilisant les différentes versions de la page wiki. Cette méthode n'est pas compatible ni avec le principe de la simplicité des wikis puisque les modifications concurrentes sont fréquentes dans le contexte d'un wiki sémantique pair-à-pair, ni avec le concept de préservation des intentions de l'utilisateur (Sun *et al.*, 1998) d'un outil collaboratif puisqu'il y aura des pertes de mise à jour des changements concurrents d'une page wiki sur des différents pairs. La fusion avec la règle de Thomas peut être acceptable dans un wiki sémantique pair-à-pair dans le cas (a) où il existe des restrictions à respecter sur les annotations dans les pages wiki fusionnées même en présence des éditions concurrentes et (b) où la perte des mises à jour est tolérée.

#### 4.2. La stratégie WOOT

WOOT gère une page wiki comme une séquence de lignes. WOOT considère la mise à jour d'une ligne comme une suppression de cette ligne suivie d'une insertion d'une nouvelle ligne avec le nouveau contenu. Afin d'assurer la convergence des répliques, WOOT ne supprime jamais les lignes. Les lignes supprimées sont toujours maintenues dans le modèle de la page ; elles sont tout simplement invisibles par l'utilisateur. Dans la figure 3, nous exécutons le même scénario de la figure 2, mais cette fois en utilisant l'algorithme WOOT(Oster *et al.*, 2006) pour effectuer la fusion. Sur les deux sites, les modifications effectuées par chaque utilisateur sont détectées par le système sous forme de patches<sup>5</sup>, p1 sur le site1 et p2 sur le site 2. Chaque patch est exécuté localement sur un site, envoyé et intégré sur l'autre site. La propriété de la convergence dans WOOT assure que les deux sites convergent vers le même résultat final. L'algorithme WOOT préserve les intentions de l'utilisateur i.e., tous les effets concurrents sont visibles dans la version finale de la page. Le résultat de la fusion automatique des modifications concurrentes sur la page nécessite sa révision par un utilisateur. Pour résoudre ce problème WOOT fait recours à un mécanisme de conscience de la concurrence (Alshattawi *et al.*, 2008). Ce mécanisme marque les pages fusionnées comme "produites par le serveur" et il marque aussi toutes les lignes provenant des opérations concurrentes dans le texte de la page wiki. Dans la figure 3, les lignes marquées dans l'état final sont générées par le système de la conscience de la concurrence. Elles reflètent les modifications concurrentes produites. WOOT a été conçu pour fusionner des structures linéaires de données sans tenir compte des annotations sémantiques dans les pages wiki. La fusion des pages wiki contenant des annotations sémantiques dans un wiki sémantique pair-à-pair que ce soit avec la règle de Thomas ou avec WOOT est automatique ce qui nécessite une révision de ces pages par les utilisateurs. Nous considérons que la meilleure solution adaptée au besoin des utilisateurs est celle qui permet aux utilisateurs de voir toutes les modifications présentes dans une édition concurrente. Cette solution est réalisable avec la stratégie de WOOT. Dans WOOT toutes les modifications concurrentes sont visibles dans la page

5. Un patch est un delta entre deux versions successives d'une page wiki. Il contient la séquence des opérations élémentaires nécessaires pour transformer une version en une autre.



**Figure 3.** Édition concurrente avec l’algorithme WOOT

wiki, ce qui permet à l’utilisateur de changer facilement le résultat de la fusion s’il le souhaite. Par contre avec la règle de Thomas, pour que l’utilisateur change le résultat d’une fusion, il est obligé d’accéder à des versions de l’historique de la page wiki.

## 5. L’approche SWooki

SWooki est un wiki sémantique pair-à-pair intégrant les technologies du web sémantique basé sur Wooki(Weiss *et al.*, 2007). Il est composé d’un ensemble de ser-

veurs wiki sémantiques interconnectés (des pairs) qui forment un réseau pair-à-pair. Les pages wiki et les annotations sémantiques dans ces pages sont répliquées sur tous les pairs. Chaque pair héberge une copie de toutes les pages wiki et un RDF store, il peut offrir le service du wiki d'une façon autonome. Dans ce réseau, chaque serveur possède un identifiant unique et joue le même rôle. Swooki est un réseau pair-à-pair non structuré, un serveur Swooki peut rejoindre ou quitter le réseau à tout moment. Les utilisateurs collaborent dans l'édition et dans l'annotation des pages wiki. Ils ajoutent les annotations sémantiques directement dans le texte des pages wiki. Les utilisateurs peuvent annoter toute page wiki sur n'importe quel pair, ces annotations seront propagées dans le texte à tous les pairs puis extraites et stockées dans le RDF store de chaque pair. Cette connaissance partagée éditée par collaboration améliore la structuration de Swooki, le contenu de ses pages devient processable par les machines. Ceci améliore la recherche, la navigation en accédant aux informations pertinentes dans les pages et aux relations entre les pages. En outre, cette connaissance produite dans les pages wiki peut être interrogée, extraite et réutilisée par des applications externes. Actuellement il n'existe aucun algorithme de réplication optimiste qui permet la convergence et qui préserve les intentions des utilisateurs pour le nouveau modèle de données (texte avec annotations). Toutefois, nous avons étendu l'algorithme de réplication optimiste WOOT utilisé dans Wooki afin de manipuler facilement les annotations sémantiques en les considérant comme une partie du texte (voir la section 5.3). Par conséquent, les pages wiki et la même base de connaissances finiront par converger sur tous les pairs. Swooki supporte tous les cas d'utilisation d'un wiki pair-à-pair spécifiés dans la section 2 grâce à sa réplication totale des pages wiki et à son algorithme de réplication optimiste.

### 5.1. *Modèle de données*

Une page Swooki possède un identifiant unique `pageid` qui est le nom de la page. L'identifiant est fixé lors de la création de la page. Si certains sites créent simultanément des pages avec le même nom, leur contenu sera fusionné directement par l'algorithme WOOT étendu. Une page wiki est considérée comme une séquence de lignes. Chaque ligne est considérée comme un quadruplet `< idl, contenu, degré, visibilité >`. `idl` est l'identifiant unique de la ligne. Les identifiants des lignes sont totalement ordonnés. L'attribut `contenu` est une chaîne de caractères représentant le contenu d'une ligne wiki lisible. Cet attribut est un mélange de texte et d'annotations sémantiques exprimant des métadonnées sur la page wiki. Ces métadonnées sont stockées dans le RDF store sous forme de triplets RDF ayant comme sujet l'identifiant de la page. L'attribut `dégré` d'une ligne est un entier fixé durant la génération de la ligne et utilisé par l'algorithme WOOT. La `visibilité` d'une ligne est spécifiée par l'attribut `visibilité`. Il est un booléen représentant si une ligne est visible ou non. Afin d'assurer la convergence des répliques, WOOT ne supprime jamais les lignes. Par contre, ces lignes sont toujours maintenues dans le modèle de la page ; elles sont tout simplement invisibles par l'utilisateur.

## 5.2. Les opérations d'édition

Swooki définit deux opérations  $\text{Insert}(\text{pageid}, \text{ligne}, \text{IP}, \text{IN})$  et  $\text{Delete}(\text{pageid}, \text{idl})$  d'une ligne. Une modification d'une ligne est considérée comme une suppression de la ligne suivie par une insertion d'une nouvelle ligne. Ces deux opérations sont définies comme suit :

1)  $\text{Insert}(\text{pageid}, \text{ligne}, \text{IP}, \text{IN})$  permet d'insérer la ligne  $\text{ligne} = \langle \text{idl}, \text{contenu}, \text{degré}, \text{visibilité} \rangle$  entre les deux lignes IP (la ligne précédente) et IN (la ligne suivante) dans la page wiki identifiée par pageid.

2)  $\text{Delete}(\text{pageid}, \text{idl})$  permet de rendre invisible la ligne identifiée par idl dans la page identifiée par pageid.

Chaque nouvelle page créée est initialisée contenant ID et IF. Elles sont des lignes spéciales indiquant le début et la fin d'une page. Quand un site  $x$  génère une opération d'insertion sur la page  $p$  entre les lignes ligneA et ligneB, il génère  $\text{Insert}(p, \langle x, ++ \text{clockx} \rangle, \text{contenu}, d, \text{vrai}, \text{idl}(\text{ligneA}), \text{idl}(\text{ligneB}))$  où  $d = \max(\text{degré}(\text{LigneA}), \text{degré}(\text{ligneB})) + 1$ . Les lignes ID et IF ont un degré de 0. Le degré est un paramètre interne nécessaire à WOOTO.

## 5.3. Algorithme de fusion dans Swooki

Dans ce paragraphe, nous présentons l'algorithme de fusion des pages wiki contenant des annotations que nous avons développé pour Swooki. Cet algorithme est une extension de l'algorithme WOOT qui prend en considération les annotations et qui assure leur convergence. La mise à jour des triplets dans un RDF store est faite par la mise à jour du contenu des pages wiki c'est-à-dire en utilisant seulement les opérations d'insertion et de suppression des lignes de texte.

– Durant la sauvegarde d'une page (voir figure 4) :

Lors de la sauvegarde d'une page wiki, un algorithme de Diff(Myers, 1986) calcule la différence entre la version sauvegardée et la version précédente de la page et génère un patch. Un patch est un ensemble d'opérations d'insertion et de suppression de lignes. Le patch est intégré localement et propagé vers d'autres sites afin d'être intégré et exécuté.

```

1 Save(page, oldPage)
2   let P := Diff(page, oldPage)
3   For each Op in P do
4     Receive(Op)
5   done
6   Broadcast(P)

```

**Figure 4.** La sauvegarde d'une page.



– Durant la réception d'une opération (voir figure 5) :

Lors de la réception d'une opération, l'algorithme de fusion teste si cette opération satisfait sa précondition. Si c'est le cas, deux types d'intégration sont produites, une intégration des annotations sémantiques et une autre textuelle. Dans la première, les annotations contenues dans la ligne de l'opération sont extraites au moyen de la fonction `ExtractRDF()`. Cette fonction traduit ces annotations en triplets RDF ayant comme sujet l'identifiant de la page, elle retourne un ensemble de triplets. Selon le type de l'opération, les triplets RDF contenus dans la ligne sont supprimés ou insérés dans le RDF store. Pour une opération de suppression, l'algorithme teste la visibilité de la ligne dans cette opération. En cas de suppression concurrente de la même ligne sur différents sites, ce test garantit que les annotations sur cette ligne sont supprimées une seule fois du RDF store. La deuxième intègre l'insertion ou la suppression d'une ligne selon le type de l'opération au moyen de l'algorithme WOOTO. Dans le cas de réception d'une opération qui ne satisfait pas sa précondition, elle sera mise en attente dans une file. Cette file est parcourue après intégration des opérations, ainsi les opérations satisfaites dans la file sont intégrées et retirées de cette file.

```

1 Receive(Op)
2   if isExecutable (Op) then
3     let S := ExtractRDF(Op)
4     if type(Op) = insert then
5       if S ≠ ∅ then
6         InsertRDF(S)
7       endif
8       IntegrateIns (Op)
9     endif
10    if type(Op) = delete and Op.visible = true then
11      if S ≠ ∅ then
12        DeleteRDF(S)
13      endif
14      IntegrateDel (Op)
15    endif
16  else
17    waitingLog := waitingLog ∪ {Op}
18  endif

```

**Figure 5.** La réception d'une opération.

Les opérations d'insertion et de suppression des lignes sont intégrées comme dans WOOTO en respectant leurs préconditions (voir figure 6). Une opération de suppression est intégrée lorsque la ligne à supprimer existe dans la page. Par contre, l'opération d'insertion est intégrée lorsque les lignes entre lesquelles la ligne doit être insérée existent dans la page. La fonction `containsL()` teste l'existence d'une ligne dans une page, elle retourne une valeur booléenne.

```

1 isExecutable (op)
2 let pageid := pageID(op), l := ligne (op),
3     IP := previousLine (op), IN := nextLine(op)
4 if type(op) = del then
5     return containsL(pageid, idl (l))
6 else
7     return containsL(pageid, idl (IP))
8     and containsL(pageid, idl (IN))
9 endif

```

**Figure 6.** La précondition d'une opération.

```

1 InsertRDF(S)
2 for each triple in S do
3     if Contains( triple ) then
4         triple .counter++
5     else
6         insertTriple ( triple )
7     endif
8 done

```

**Figure 7.** L'insertion dans le RDF store.

L'insertion des triplets RDF dans le RDF store est faite par la méthode InsertRDF. Cette méthode teste l'existence d'un triplet RDF dans le store avant de l'insérer, si c'est le cas il incrémente son compteur de un dans le RDF store ; sinon elle insère le triplet dans le RDF store avec un compteur égal à un. La fonction Contains() teste la présence d'un triplet dans le RDF store, elle retourne une valeur booléenne.

```

1 DeleteRDF(S)
2 for each triple in S do
3     triple .counter--
4     endif
5 done

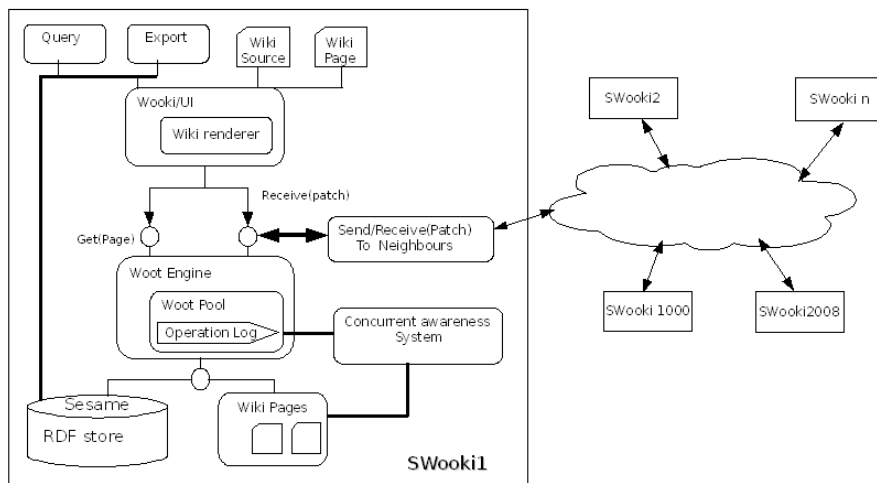
```

**Figure 8.** La suppression dans le RDF store.

La suppression des triplets RDF dans le store est faite par la méthode DeleteRDF. Cette méthode décrémente la valeur du compteur dans le store de un. Les triplets ne sont jamais supprimés du store, ceci est nécessaire pour assurer la convergence des RDF stores de tous les pairs. Les triplets ayant un compteur négatif sont ignorés durant l'exécution des requêtes sur le store.

## 6. Architecture et implémentation de Swooki

Cette section donne une vue globale de l'architecture de Swooki. Elle détaille ses composantes et décrit son implémentation en insistant sur ses caractéristiques. Un pair de Swooki est composé de trois composantes principales comme indiqué dans la figure 9 : la composante site Wooki, une composante sémantique et la composante de conscience de la concurrence.



**Figure 9.** L'architecture de Swooki

### 6.1. Site Wooki

Un site Wooki est composé de trois composantes principales. La composante de base est l'algorithme WOOT étendu qui est en charge de la génération et de l'intégration des opérations d'édition sur les pages wiki. Un site Wooki produit le même résultat indépendamment de l'ordre de réception des opérations. Les opérations sont intégrées lorsque leurs préconditions sont satisfaites, dans le cas contraire elles seront en attente dans une file d'attente. Une autre composante est en charge de l'interface de l'utilisateur. Les utilisateurs éditent une vue du modèle qui présente seulement le contenu des lignes visibles et non pas directement le modèle. Les opérations d'insertion et de suppression des lignes dans WOOT sont détectées par un algorithme de Diff (Myers, 1986) qui calcule la différence entre la page demandée et la page modifiée par l'utilisateur. Ces opérations sont intégrées localement et diffusées aux autres sites de Wooki. La dernière composante est chargée de diffuser les opérations locales et la récupération des opérations distantes. Elle combine un algorithme de propagation probabiliste (Eugster *et al.*, 2003) avec un algorithme d'anti-entropie (Demers *et*

*al.*, 1987). Le premier algorithme propage rapidement les mises à jour sur le réseau pair-à-pair et gère d'une manière dynamique l'appartenance des pairs au réseau. Le dernier est responsable de récupérer les mises à jour manquantes des sites déconnectés après leur reconnexion.

## 6.2. La composante sémantique

**Annotation :** Dans SWooki les annotations sémantiques sont intégrées directement dans le texte. Les utilisateurs ajoutent facilement les annotations dans le texte en utilisant un langage de balisage spécial, c'est similaire à ajouter des liens typés dans les pages wiki. La syntaxe utilisée est très similaire à celle utilisée dans Semantic MediaWiki (voir la figure 2). SWooki adopte cette façon d'annotation pour deux raisons : d'une part, pour faciliter la phase d'annotation pour les utilisateurs sans avoir besoin de changer d'interfaces pendant l'édition et l'annotation et d'autre part, pour intégrer facilement l'aspect sémantique dans Wooki sans changer son modèle de données. WOOT a été conçu pour synchroniser des répliques de données linéaires. En intégrant les données sémantiques dans le texte, WOOT assure aussi leur convergence.

**Extraction et Stockage :** Nous avons choisi d'utiliser Sesame(Sesame, 1997) comme RDF store pour le stockage des annotations sémantiques dans SWooki. Un RDF store est en fait une interface API qui stocke des triplets RDF de différentes manières et qui permet l'exécution des requêtes. Le langage de requêtes utilisé est SPARQL(SPARQL, 2008) recommandé par le W3C. La version de Sesame implémentée dans SWooki est 2,0. Les RDF stores actuels ne dupliquent pas les triplets RDF dans le but de respecter les sémantiques RDF c'est-à-dire un graphe minimal décrit une ressource sans le besoin de dupliquer les mêmes triplets. Les wikis sémantiques intégrant les annotations dans le texte tel que Semantic Wikipedia parsent le contenu d'une page après chaque sauvegarde et mettent à jour ses annotations dans le RDF store. Par exemple, la suppression d'une annotation dans une page nécessite la suppression de toutes les annotations de la version précédente de la page du RDF store et de les rajouter toutes dans le RDF store sauf l'annotation supprimée. Afin de garantir une efficacité dans la mise à jour des RDF stores dans un wiki sémantique pair-à-pair, seules les annotations modifiées dans les pages sont modifiées dans les stores. Pour réaliser ceci en assurant la convergence des annotations dans les RDF stores de SWooki, nous avons transformé Sesame à un multi-ensemble de triplets RDF tout en attribuant des compteurs à ces triplets i.e. l'ajout d'un triplet est équivalent à l'ajout d'un quadruplet < sujet, prédicat, object, compteur >. L'ajout d'une annotation plusieurs fois dans une page incrémente son compteur dans le RDF store. Similairement sa suppression décrémenté son compteur. Cette solution respecte bien les sémantiques des RDF sans avoir une vraie duplication des triplets. La composante sémantique fournit aussi une fonctionnalité pour exporter un graphe RDF correspondant aux annotations d'une page Wiki.

**Navigation et Recherche :** Dans SWooki, les utilisateurs peuvent naviguer entre les pages en utilisant les annotations sémantiques dans ces pages (voir la figure 10).



Figure 10. L'interface utilisateur de Swooki

Ces annotations placées dans un factbox en bas des pages wiki améliorent la structuration du Swooki. La recherche dans Swooki est au-delà des mots-clés, elle est basée sur la sémantique des pages wiki et sur les relations entre ces pages. Une interface de recherche est utilisée (voir figure 11), les utilisateurs peuvent rechercher toutes les pages contenant une certaine annotation ou toutes les annotations d'une page wiki. Par exemple, trouver tous les pays qui sont situés en Europe. Les utilisateurs peuvent exécuter des requêtes SPARQL. Cette fonction est assurée par le module des requêtes de Sesame. L'exécution des requêtes dans Swooki est très efficace et passe à l'échelle, une requête peut être exécutée sur n'importe quel pair. Chaque pair possède localement toutes les annotations des pages wiki dans son RDF store. D'une part, ceci ne nécessite plus de trouver et de transférer les annotations d'autres pairs lors de l'exécution d'une requête. D'autre part, la charge d'exécution des requêtes est distribuée sur tous les pairs de Swooki. Swooki permet aussi de générer du contenu dynamique dans des pages wiki à l'aide des requêtes intégrées dans ces pages. Cette fonction est actuellement assurée dans Swooki pour des requêtes simples, mais elle peut être facilement étendue pour d'autres plus complexes.

### 6.3. La composante de conscience de la concurrence

Le traitement de l'édition concurrente dans un wiki pair-à-pair diffère de celui des wikis traditionnels ou sémantiques. Dans le cas d'une édition concurrente dans un wiki centralisé, une nouvelle version d'une page est créée pour l'utilisateur qui sau-

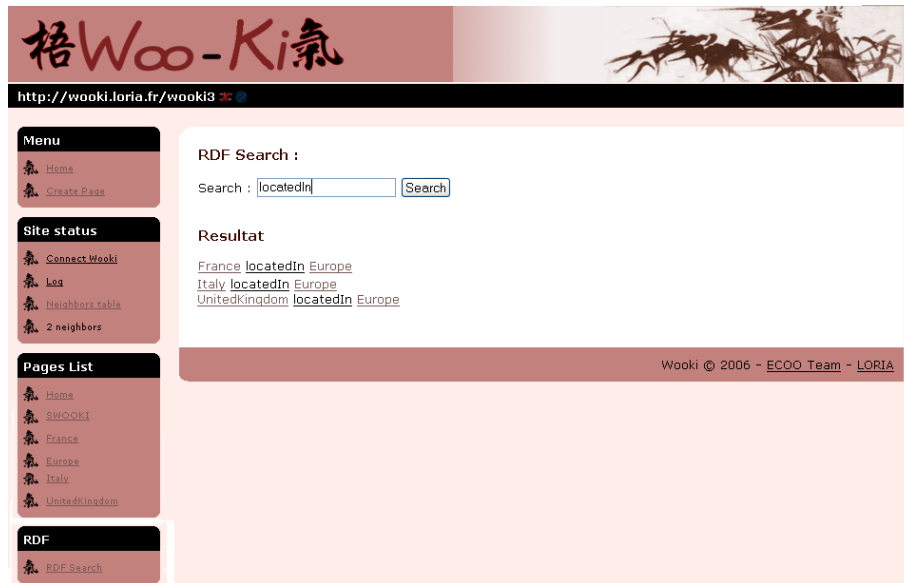


Figure 11. L'interface de recherche dans SWiki

vegarde premièrement ses changements. Ce comportement est utilisé dans Semanti MediaWiki. Le serveur détecte un changement concurrent et demande aux utilisateurs de fusionner manuellement leurs changements avec les changements concurrents du premier utilisateur. Lorsque la fusion est faite par les utilisateurs et il ne reste aucune version concurrente, le dernier utilisateur sauvegarde et une nouvelle version de la page est produite. La fusion des changements concurrentes est toujours sous le contrôle d'un utilisateur. Ainsi, toutes les versions d'une page wiki sont produites par un utilisateur et ont été possiblement révisées par un utilisateur avant d'être créées. Dans un wiki pair-à-pair, les changements concurrents peuvent avoir lieu et peuvent être sauvegardés sur les différents pairs. Les fusions ne sont pas exécutées juste après la sauvegarde des pages mais lors de la réception des changements distants par chacun des pairs. Pour assurer une convergence éventuelle, les fusions sont faites automatiquement par les pairs du wiki. Par conséquent, il peut avoir des pages wiki visibles produites par des fusions de pairs et leur contenu nécessite une révision. Dans le but de traiter ce problème, SWiki fait recours à un système de conscience de la concurrence (Alshattnawi *et al.*, 2008) pour les wikis pair-à-pair. Ce système a pour but d'informer les utilisateurs sur l'état d'une page wiki concernant la concurrence et marque les lignes concurrentes dans cette page. Durant l'intégration des modifications distantes sur un pair, le système calcule l'ensemble des opérations concurrentes en utilisant une horloge plausible (plausible clock) (Torres-Rojas *et al.*, 1999). L'horloge plausible est un moyen pour détecter la concurrence qui passe à l'échelle. Si l'ensemble des opérations concurrentes n'est pas vide, la page wiki est marquée comme

une page produite par un pair c'est-à-dire elle a été fusionnée automatiquement par un pair et doit être révisée par les utilisateurs, sinon la page est dans un état produite par un utilisateur. Le mécanisme de conscience de la concurrence calcule l'état à partir duquel les modifications concurrentes ont commencées. Ensuite, il insère des directives pour les lignes modifiées d'une façon concurrente lors de la demande de la page par les utilisateurs. Les régions marquées dans la page (voir figure 12) informent les utilisateurs sur les parties modifiées concurrentement dans la page wiki. Le mécanisme de conscience de la concurrence est implémenté dans Swooki et il peut être testé dans la version actuelle du prototype de Swooki.

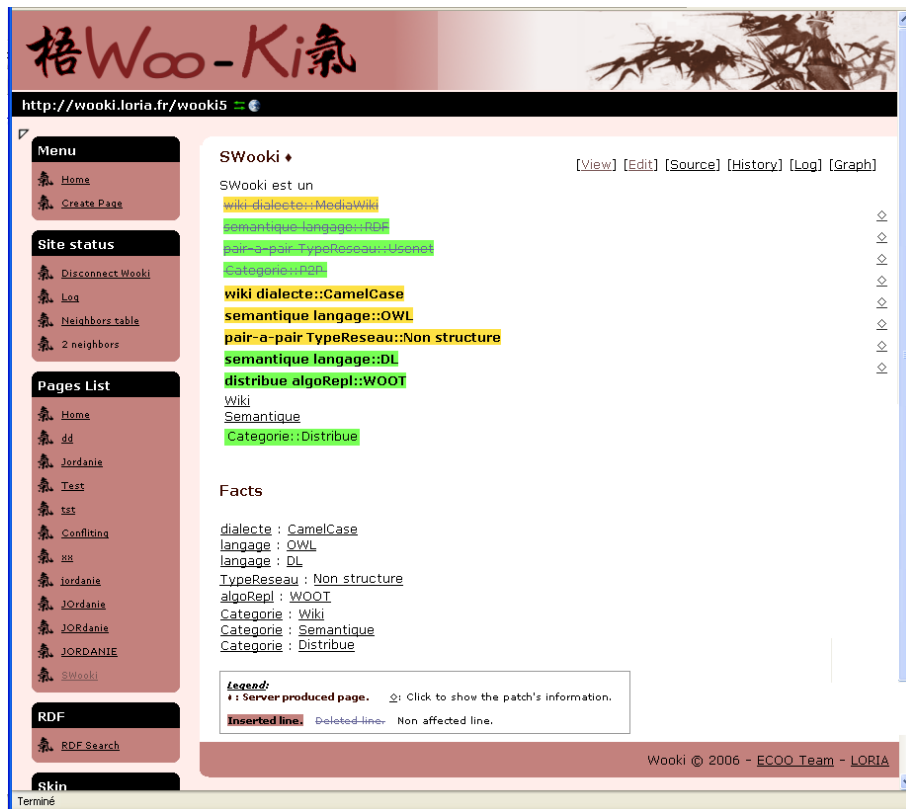


Figure 12. La détection des modifications concurrentes dans Swooki

Le prototype Swooki a été implémenté dans des servlets Java sur un serveur Tomcat. Ce prototype est disponible avec une licence GPL sur sourceforge à l'adresse <http://sourceforge.net/projects/wooki>. La possibilité de tester Swooki est également disponible en visitant un pair de Swooki sur <http://wooki.loria.fr/wooki5>.

## 7. Analyse de performance

Le nombre de pairs dans le réseau ne fait pas parti des paramètres de la complexité de l'algorithme de Swooki. Par conséquent, Swooki passe à l'échelle vis-à-vis au nombre de pairs dans le réseau. Toutefois, Swooki est limité par la capacité de stockage de chaque pair puisque les données sont répliquées totalement sur tous les pairs. Mais ceci ne pose pas un vrai problème pour les raisons suivantes : en prenant Wikipedia comme exemple, nous découvrons qu'en 2008, la taille totale de la Wikipedia française est inférieure à 2,5 Go. La taille de la Wikipedia française grossit lentement, par exemple sa taille a grossi de 3% entre les mois de mai et d'avril en 2008<sup>6</sup>. Nous analysons les performances de Swooki par rapport au nombre de messages nécessaires pour exécuter une requête, pour propager une modification et pour synchroniser les données.

- L'exécution des requêtes : Chaque pair dans Swooki peut exécuter chaque requête sémantique localement sans générer du trafic réseau pour la résoudre.

- Livraison des messages : puisque l'algorithme de Swooki ne génère aucun trafic pour assurer la convergence, le coût du trafic dans notre système est le coût du Wooki classique qui est le coût du trafic de l'algorithme de propagation probabiliste Lpbcast (Eugster *et al.*, 2003) et celui de l'algorithme anti-entropie. Dans (Eugster *et al.*, 2003), les auteurs ont montré que leur algorithme passe à l'échelle. Un site envoie toujours un message à un sous ensemble de pairs du réseau. Le coût de l'anti-entropie est plus important que celui de Lpbcast mais il est exécuté dans des cas particuliers tels que lorsque un site déconnecté rejoint le réseau. Le volume des changements générés par la Wikipedia française est d'environ 100 Mo par mois entre septembre 2006 et mai 2008, ce qui montre que le trafic moyen est faible.

- Synchronisation des données : La complexité de l'intégration d'une opération est  $O(L^2)$  (Weiss *et al.*, 2007), où L est le nombre de lignes qui ont été insérées dans la page wiki. Puisque les lignes supprimées sont marquées comme supprimées et qu'il n'y a pas un algorithme pour purger l'historique compatible avec les contraintes du pair-à-pair, la taille de la page wiki peut grandir à l'infini. Toutefois, les wikis comme Wikipédia conservent toutes les modifications dans l'historique et ne les suppriment jamais. Par conséquent, dans le contexte d'un wiki, notre solution est acceptable.

## 8. Conclusion

Swooki est le premier wiki sémantique pair-à-pair. Il combine les avantages des wikis sémantiques et des wikis pair-à-pair. Les pages wiki et les annotations sémantiques dans ces pages sont répliquées sur tous les pairs. Les utilisateurs collaborent dans l'édition et dans l'annotation des pages wiki. Ces annotations seront propagées dans le texte à tous les pairs puis extraites et stockées dans le RDF store de chaque pair. L'algorithme de réplcation optimiste de Swooki fusionne les annotations sémant-

6. <http://stats.wikimedia.org/EN/TablesDatabaseEdits.htm>



tiques sous forme de texte. Il assure la convergence des pages wiki et des RDF stores sur tous les pairs. SWoiki traite les problèmes des wikis sémantiques en supportant la tolérance aux pannes, l'édition massive, un mode d'édition déconnectée, et permet le partage des coûts de l'infrastructure grâce à sa réplication totale des pages wiki et à son algorithme de réplication optimiste. SWoiki suit le type d'approches des wikis sémantiques qui utilisent le wiki pour la production d'une base de connaissances. Cette base distribuée est éditée par collaboration et exploitée par des requêtes. SWoiki n'est pas conçu comme un éditeur d'ontologie spécialisé. SWoiki fournit les mêmes fonctionnalités de base de la Semantic Wikipedia. Dans SWoiki, la détection et le traitement des modifications concurrentes sont supportés par un système de conscience de la concurrence pair-à-pair. Dans le futur, nous envisageons d'étudier la possibilité d'intégrer des ontologies dans SWoiki afin d'améliorer la cohérence des annotations. Nous menerons des études d'usage dans SWoiki dans le but d'évaluer son efficacité.

## 9. Bibliographie

- Allen L., Fernandez G., Kane K., Leblang D., Minard D., Posner J., « ClearCase MultiSite : Supporting Geographically-Distributed Software Development », *Software Configuration Management : Icse Scm-4 and Scm-5 Workshops : Selected Papers*, 1995.
- Alshattnawi S., Canals G., Molli P., « Concurrency Awareness in a P2P Wiki System », *Proceedings of CTS 2008, The 2008 International Symposium on Collaborative Technologies and Systems*, Irvine, California, USA, May, 2008.
- Androutsellis-Theotokis S., Spinellis D., « A survey of peer-to-peer content distribution technologies », *ACM Computing Surveys*, vol. 36, n° 4, p. 335-371, December, 2004.
- Buffa M., Gandon F., « SweetWiki : semantic web enabled technologies in Wiki », *WikiSym'06 : Proceedings of the international symposium on Symposium on Wikis*, ACM Press, Odense, Denmark, p. 69-78, August, 2006.
- Co-op, « Reliable Software », [http://www.relisoft.com/co\\_op](http://www.relisoft.com/co_op), 2006.
- Decker B., Ras E., Rech J., Klein B., Hoecht C., « Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis », *Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE)*, November, 2005.
- Demers A., Greene D., Hauser C., Irish W., Larson J., Shenker S., Sturgis H., Swinehart D., Terry D., « Epidemic Algorithms for Replicated Database Maintenance », *Proceedings of the ACM Symposium on Principles of Distributed Computing*, ACM Press, Vancouver, Canada, p. 1-12, August, 1987.
- Eugster P. T., Guerraoui R., Handurukande S. B., Kouznetsov P., Kermarrec A.-M., « Lightweight probabilistic broadcast », *ACM Trans. Comput. Syst.*, vol. 21, n° 4, p. 341-374, November, 2003.
- Gears, « Gears - Enabling Offline Web Applications », <http://gears.google.com/>, 2008.
- Johnson P., Thomas R., « RFC0677 : Maintenance of duplicate databases », *Internet RFCs*, 1975.
- Morbidoni C., Tummarello G., Erling O., Bachmann-Gmür R., « RDFSyc : efficient remote synchronization of RDF models », *Proceedings of the ISWC/ASWC2007*, vol. 4825, p. 533-546, November, 2007.

- Morris J., « DistriWiki : a distributed peer-to-peer wiki network », *WikiSym '07 : Proceedings of the 2007 international symposium on Wikisp.* 69-74, 2007.
- Myers E. W., « An O(ND) Difference Algorithm and Its Variations », *Algorithmica*, vol. 1, n° 2, p. 251-266, March, 1986.
- Oaks S., Gong L., *Jxta in a Nutshell*, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2002.
- Oster G., Urso P., Molli P., Imine A., « Data Consistency for P2P Collaborative Editing », *Proceedings of the ACM Conference on Computer-Supported Cooperative Work - CSCW 2006*, ACM Press, Banff, Alberta, Canada, November, 2006.
- RDF, « Resource Description Framework », <http://www.w3.org/RDF/>, 2004.
- Saito Y., Shapiro M., « Optimistic Replication », *ACM Computing Surveys*, vol. 37, n° 1, p. 42-81, 2005.
- Schaffert S., « IkeWiki : A Semantic Wiki for Collaborative Knowledge Management. », *WE-TICE*, IEEE Computer Society, p. 388-396, 2006.
- Sesame, « openRDF.org », <http://www.openrdf.org/>, 1997.
- Souzis A., « Building a Semantic Wiki », *Intelligent Systems, IEEE*, vol. 20, n° 5, p. 87-91, 2005.
- SPARQL, « SPARQL Query Language for RDF W3C Recommendation », January, 2008.
- Spencer H., Lawrence D., *Managing Usenet*, O'Reilly, January, 1988.
- Sun C., Jia X., Zhang Y., Yang Y., Chen D., « Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems », *ACM Trans. Comput.-Hum. Interact.*, vol. 5, n° 1, p. 63-108, 1998.
- Tazzoli R., Castagna P., Campanini S., « Towards a semantic wiki wiki web », *Proceedings of the International Semantic Web Conferenc (ISWC)*, 2004.
- Torres-Rojas F. J., Ahamad M., « Plausible clocks : constant size logical clocks for distributed systems », *Distrib. Comput.*, vol. 12, n° 4, p. 179-195, 1999.
- Tummarello G., Morbidoni C., Petersson J., Puliti P., Piazza F., « RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications. », *P2PKM*, CEUR Workshop Proceedings, CEUR-WS.org, August, 2004.
- URI, « URIs, URLs, and URNs : Clarifications and Recommendations 1.0 », <http://www.w3.org/TR/uri-clarification>, 2001.
- Völkel M., Groza T., « SemVersion : An RDF-based Ontology Versioning System », *Proceedings of IADIS International Conference on WWW/Internet*, IADIS, Murcia, Spain, October, 2006a.
- Völkel M., Kröttsch M., Vrandecic D., Haller H., Studer R., « Semantic Wikipedia », *Proceedings of the 15th international conference on World Wide Webp.* 585-594, May, 2006b.
- Weiss S., Urso P., Molli P., « Wooki : a P2P Wiki-based Collaborative Writing Tool », *Web Information Systems Engineering*, Springer, Nancy, France, December, 2007.