SWOOKI: A Peer-to-peer Semantic Wiki

Charbel Rahhal, Hala Skaf-Molli, Pascal Molli

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

SWOOKI: A Peer-to-peer Semantic Wiki

Charbel Rahhal — Hala Skaf-Molli — Pascal Molli

N° ????

Mars 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

INRIA
LORRAINE

SWOOKI: A Peer-to-peer Semantic Wiki

Charbel Rahhal*, Hala Skaf-Molli†, Pascal Molli ‡

Thème COG — Systèmes cognitifs
Projet ECOO

Rapport de recherche n° ???? — Mars 2008 — 14 pages

Abstract: Wiki systems have evolved in two different ways : semantic wikis and peer to peer wikis. Semantic wikis allow to embed formalized knowledge within wiki pages. P2P wikis offer support for massive collaboration, off-line editing and ad-hoc collaboration.

In this paper, we propose to combine the advantages of semantic wikis and P2P wikis in order to design a peer-to-peer semantic wiki. The main challenge is how to merge wiki pages that embed semantic annotations. Merging algorithms used in P2P wiki systems have been designed for linear text and not for semantic data. In this paper, we show how we can combine different optimistic replication algorithms to build a P2P semantic wiki.

Key-words: Semantic wikis, Peer to Peer wikis, optimistic replication, merging algorithms, P2P semantic Wikis

* charbel.rahal@loria.fr, ECOO Project, Nancy-University, LORIA, INRIA Centre - Nancy Grand Est
† skaf@loria.fr, ECOO Project, Nancy-University, LORIA, INRIA Centre - Nancy Grand Est
‡ molli@loria.fr, ECOO Project, Nancy-University, LORIA, INRIA Centre - Nancy Grand Est

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 —Télécopie : +33 3 83 27 83 19

SWOOKI: Un Wiki Sémantique Pair-à-Pair

Résumé : Les systèmes wiki ont évolués de deux manières différentes : soit vers des wikis sématiques, soit vers des wikis pair-à-pair. Les wikis sématiques permettent d'inclure une connaissance formalisée dans les pages wikis. Les wikis pair-à-pair offrent un support pour l'édition massive, le mode d'édition hors-ligne et une collaboration ad-hoc.

Dans ce papier, nous proposons de combiner les avantages des wikis sémantiques et des wikis pair-à-pair dans le but de concevoir un wiki sémantique pair-à-pair. Un des défis majeurs est la fusion des pages wikis contenant des annotations sématiques. Les algorithmes de merge utilisés dans les wikis pair-à-pair ont été conçus pour des structures linéaires et non pas pour des annotations sémantiques. Dans ce papier, nous présentons un moyen de combiner les différents algorithmes de réplication optimiste pour réaliser un wiki sématique pair-à-pair.

Mots-clés : Wikis sémantiques, wikis Pair-à-Pair, réplication optimiste, algorithmes de merge, Wikis sémantiques P2P

Contents

1 Introduction

Nowadays, Wikis are the most popular web-based collaborative writing tools. They allow users connected to the web to concurrently edit and modify wikis pages in a simple way. Simplicity and ease of use support for non experts are the keys of success of Wikis. Wikipedia, the free encyclopedia, is the most famous example of mass collaboration through Wiki.

In spite of the popularity, wikis suffer from the difficulty of navigation and information retrieval. This is the main consequence of the low structuring of Wiki pages. Structuring wiki systems to enhance navigation through easy access to the relevant information is a major need for pushing the traditional wiki systems to turn into semantic wikis.

Semantic Wiki (SW) is a wiki engine with technologies from semantic web[1] to embed formalized knowledge, content, structures and links in wiki pages [1]. Semantic technologies within a wiki form the base for powerful question answering interfaces [2]. SW tries to preserve basic advantages of a wiki, i.e the simplicity in creating and editing pages. Semantic Wikipedia is an example of mass collaboration using semantic wikis.

Popular semantic wikis are based on the client-server (centralized) architecture. All wiki pages reside on a single server that controls operations of distributed users. Consequently, scalability, performance, fault-tolerance and load balancing are major challenges for current semantic wikis. In addition, centralized architecture suffers from censorship problem and does not support off-line work.

An approach to solve these problems is to shift from centralized architecture to full distributed (peer to peer) one. Some researches have been done to build peer-to-peer(P2P) wiki systems [3, 4]. A P2P wiki system is P2P network of wiki servers. All wiki pages are replicated on each wiki server. *A P2P wiki system is correct if all replicas eventually converge to the same state while preserving user intentions* [5]. In order to ensure correctness, P2P wiki systems change subtly the behavior of wikis in case of concurrent editing. Merging concurrent changes are not performed by humans as in traditional wikis. Wiki servers merge concurrent modifications using automatic deterministic merging algorithms [6, 7] without any human intervention. Therefore, it is not possible to ensure the quality of merged pages. Humans are informed of merged pages *aposteriori* by using concurrency awareness mechanism [8].

In this paper, we address the challenge of transforming a P2P wiki system into a P2P semantic wiki system. In fact, merging algorithms used in P2P wiki systems have been designed for linear text and not for semantic data. In this paper, we show how we can combine different optimistic replication algorithms to build a P2P semantic wiki.

The paper is organized as follows. Section 2 details different possible use cases of P2P wikis. Section 3 presents related works. We focus on available P2P wikis and some semantic wikis. Section 4 presents the general approach of SWOOKI, a P2P semantic wiki that we propose. In this section, we present three different ways to manage concurrent editing of text and semantic annotations. The last section concludes the paper.

2 Use cases for P2P semantic wikis

In this section, we detail three interesting use cases for P2P wiki systems. Our objective is to also support these scenarios with a P2P semantic wiki.

- **Massive Collaboration** In this case, a P2P wiki system is deployed as Usenet network [9]. For instance, thousands of wiki servers can be deployed within organizations or universities. Any user can connect to any wiki server. This deployment allows :
  - to handle a large number of users by dividing the load on the whole network,
  - to tolerate many faults. A crash of one wiki server does not stop the service.

[1] www.w3.org/2001/sw

  - to share the cost of the infrastructure. Wikis are set up and maintained by different organizations. Therefore, it is not necessary to collect funds just to maintain the infrastructure. For instance, Wikipedia foundation has to collect 150000$ every three months just to maintain the Wikipedia infrastructure.
  - to resist to censorship. An organization controls only one wiki server and not all data.

- **Off-line work.** Adding off-line capabilities to web applications is currently a major issue. For instance, the development of Google gears [10] and Firefox3 off-line capabilities demonstrate the need of the off-line work. Wikis are web applications and the need for off-line wiki editing is real. Current technologies for adding off-line capabilities to web applications focus on Ajax applications. However, the off-line mode of these web applications does not provide all features available in the on-line mode. This can be an obstacle for a wiki system. For instance, the off-line mode of the wiki allows navigation but it does not allow editing.

  A P2P wiki tolerates naturally off-line work by means of an integrated merge algorithm. With such system, it is possible to travel with a complete wiki system on a laptop, make changes off-line and re-synchronize with the P2P network as soon as an Internet connection is available.

- **Ad-hoc Collaborative Editing** this scenario is derived from the previous one. Imagine several off-line wiki users have a meeting. Unfortunately, there is no Internet connection available in the meeting room. Therefore, they decide to set up an ad-hoc network within the meeting room. A P2P wiki is able to propagate changes within the ad-hoc network and allows collaborative editing just for these off-line users. Of course, when the meeting is finished and users return to their organizations, their wiki systems will re-synchronize with the whole P2P network.

The above scenarios present interesting use cases of a P2P wiki. Such use cases are obviously extensible to a P2P semantic wiki. However, these scenarios illustrate the importance of merging algorithms. P2P wiki systems rely on complex synchronization algorithms. These algorithms handling linear text. In this paper, we study the possibilities to handle semantic annotations with these algorithms.

3 Related Work

In this section, we detail available peer-to-peer wikis. We show their limits with respect to our use cases and then present some available semantic wikis.

In the literature, there are three available Peer-to-peer wikis : DistriWiki[4], Co-op[11] and Wooki[3].

The basic idea of DistriWiki [4] is to store wiki pages in a Distributed Hash Table (DHT). In order to handle failures, a page is replicated several times using different hash functions. Therefore, it is possible to store the same page on different nodes of the DHT. If one DHT node is down, another node handles requests for this page. If a page is replicated 4 times, the system can tolerate 3 faults. While, this approach is simple to set up, it has the following drawbacks:

- The off-line work and ad-hoc collaboration use cases are not supported by DistriWiki. The system is running only if the DHT is accessible.

- The authors of DistriWiki do not explain clearly what happens in case of concurrent editing of the same page. However, the general strategy on a DHT is to retrieve always the freshest replica. This means that some changes are not visible in the last version of the wiki page. This is serious drawback of the system. It is well established in the CSCW community that concurrent collaborative editing has to preserve intentions [5], i.e. *if an operation has produced an observable effect on one site, this effect has to be observable on all sites.* The strategy of the freshest replica is clearly incompatible with the principle of user intentions.

- DistriWiki has been designed for traditional wiki pages. As wiki pages contain semantic annotations, these semantic annotations will be distributed within pages on the DHT. In this case, it is not possible to perform semantic queries. If we choose to have a separate DHT storage for semantic annotations, this implies that each time we need to access semantic data, we perform an access on a DHT with $O(nLog(n))$ complexity for retrieving data (n is the size of the DHT). This complexity is not compatible with intensive access to semantic data.

Co-op[11] is a commercial product. The authors claim that it offers P2P wiki features. However, there are no available research papers describing Co-op algorithms. Co-op relies on Distributed version control system (DVCS) [12] to build a P2P wiki system. DVCS have all required features to replicate textual data. They have been designed for distributed software development, it is obviously possible to use Code co-op outside this original scope. However, DVCS systems never claimed that they ensure convergence or user intentions. In these systems, a site can integrate a remote change performed on any another site at any time. This is clearly a feature required to build a P2P wiki. However, such integration is done under the control of a human. In case of conflicts generated by concurrent changes, the system generates conflict blocks in order to make the local user aware about conflicts. This strategy is suitable in case of concurrent software engineering.

The context of a P2P wiki is slightly different. In a P2P wiki system, the wiki server is continuously integrating remote changes from its neighbors. If the integration mechanism generates changes when integrating remote changes, the P2P system can start an infinite loop. This is why many DVCS just freeze the local workspace in case of conflict. This means all nodes of the P2P network containing conflicts cannot handle requests. In this case, the first use case scenario for massive collaboration is not possible. The second problem with DVCS approach is that currently they replicate file systems and text files. Therefore, they will merge semantic data as text.

Wooki [3] is a P2P wiki system composed of a set of interconnected wiki servers that form a P2P overlay network. In this overlay, each server plays the same role. As in any P2P network, membership is dynamic. Wooki servers can join or leave the network at any time. Wiki pages are replicated over all members of the overlay. Each server hosts a copy of pages and can autonomously offer the wiki service. Page copies at each site are maintained by an optimistic replication mechanism that disseminates changes and ensures consistency. This replication mechanism called WOOT [13] has been designed to replicate linear data. Where a wiki page is considered as a sequence of lines. WOOT ensures the CSCW principles of convergence and user intentions [5]. WOOT ensures that:

- **convergence:** all wiki servers eventually will converge to the same value of wiki pages.
- **user intention preservation:** a visible effect observed when a change is generated at one site will be observable on all sites. For example, for a wiki page, this means that if a line has been inserted between two lines on one site, this line will appear between these two lines in all sites, in spite of concurrent operations (even deletion).

WOOKI cannot handle concurrent editing as a traditional wiki as in a traditional wiki concurrent editing is detected when a user saves a page. In Wooki, concurrent editing is detected *aposteriori*. Consequently, Wooki cannot ensure that all visible pages are produced by a human. Some pages are safe i.e. reviewed by a human, others are automatically merged and need a human review. Wooki integrates a concurrency awareness system to notify users about the status of wiki pages[8].

WOOKI supports all use cases for a P2P wiki system, but some pages are automatically produced by a merge algorithm. Even though the quality of these pages cannot be ensured, users are aware about the existence of these pages. These pages are flagged as unsafe and concurrent modifications are highlighted for easing the review process. The main problem with the Wooki approach is that the replication algorithm is designed to merge linear text structure and not semantic data. Preserving intentions on semantic data has not yet been defined.

If we try to combine P2P wiki system with Semantic wiki systems, it is very important to know how semantic wikis represent their semantic data and how they combine textual parts with semantic parts. There are currently many different semantic wiki systems. We make a distinction between two approaches of semantic Wikis [14]: *The use of wikis for ontologies* and *The use of ontologies for wikis*. Few semantic wiki engines merge both approaches. Due to the lack of space, we cannot explore all approaches within this paper. We decide to focus on semantic wikis that follow the first approach, and more precisely on Semantic MediaWiki.

Semantic wikis that fall into the first category [2, 15, 16] are a straightforward combination of existing Wiki systems and the semantic web knowledge representation paradigms. They consider wiki pages as concepts. The edition of the semantic annotations is either directly in the text of the page, usually as typed links or separately in a semantic data wiki page. Typed links are considered as relations between concepts or attributes for concepts. For example, in Semantic MediaWiki (SMW), semantic annotations are integrated immediately in the text. This locality ensures better readability. In SMW users add annotations to the wiki text via special markup. The proposed markups are easy to use, adding annotations being similar to adding links into wiki pages.

Statement about element	Syntax in wiki source
object property	[[propertyName::ObjectName]]
attribute property	[[propertyName:= DataTypeValue]]
rdf:type class name	[[Category: ClassName]] (on article page)

Table 1: annotations syntax in Semantic MediaWiki [17]

SMW uses Object property, DataTypeProperty and Class as ontological elements. These elements are translated into OWL elements by the system for reuse. Individual represents an article page. Each article annotated or not is an individual of the ontology (an owl individual). Each annotation in an article is a statement about this article. An article is classified based on its topic using the *Category: className*.

1. *Category* is an annotation allowing users to classify articles.
2. *Object property* is an annotation describing relationships between two individuals.
3. *Attribute property* is an annotation that associates individual with values.

The example in figure 1 shows a page about France in SMW.

```
'''France''', officially the French Republic, is a country whose
metropolitan territory is located in [[Located in:: Western Europe]] and that also
comprises various overseas islands and territories located in other continents.
 France is one of the founding members of the [[member of:: European Union]].
[[Category:Country]].
```

Figure 1: France page in Semantic MediaWiki

Semantic wikis following *the use of Wikis for ontologies* approach are not intended as a general purpose ontology editors and they do not impose any restrictions on the semantic annotations. A formal ontology emerges during editing the wiki pages. Therefore, the wiki becomes the front-end of the ontology maintenance system.

Semantic wikis belonging to the second approach *the use of ontologies for wikis* can be considered as tools for ontology engineering. These wikis require loading some existent ontologies before starting using them. For instance, Ike Wiki[18] and SweetWiki [1] follow this approach.

In this paper, we investigate how we can combine the WOOKI approach with the Semantic Media wiki approach. We called this combination SWOOKI. The main issue that we address is how to merge wiki pages that contain semantic annotations. The question that arises is if this combination changes the behavior of the Semantic Media Wiki.

4 SWOOKI Approach

In this section, we focus on the construction of a peer to peer semantic wiki that follows the first approach of semantic wiki: *The use of wikis for ontologies*. These semantic wikis do not impose any restriction on semantic annotations. Therefore, it is possible to adapt text merging algorithms to semantic data. Of course, our final objective is to build a peer to peer semantic wiki that integrates both approaches.

To integrate semantic web technology, Swooki follows the philosophy of Semantic Media Wiki[2, 17]. The content is combined with semantic annotations, semantic annotations being embedded in the wiki text. In this paper, we concentrate on the integration of concurrent modifications in peer to peer semantic wikis.

In the next subsections, we detail three different ways to manage concurrent editing of wiki pages that mix textual and semantic data. We apply the *The last writer wins strategy and Woot strategy* because they are the only available merging algorithm for peer-to-peer wikis.

4.1 Last Writer Wins Strategy

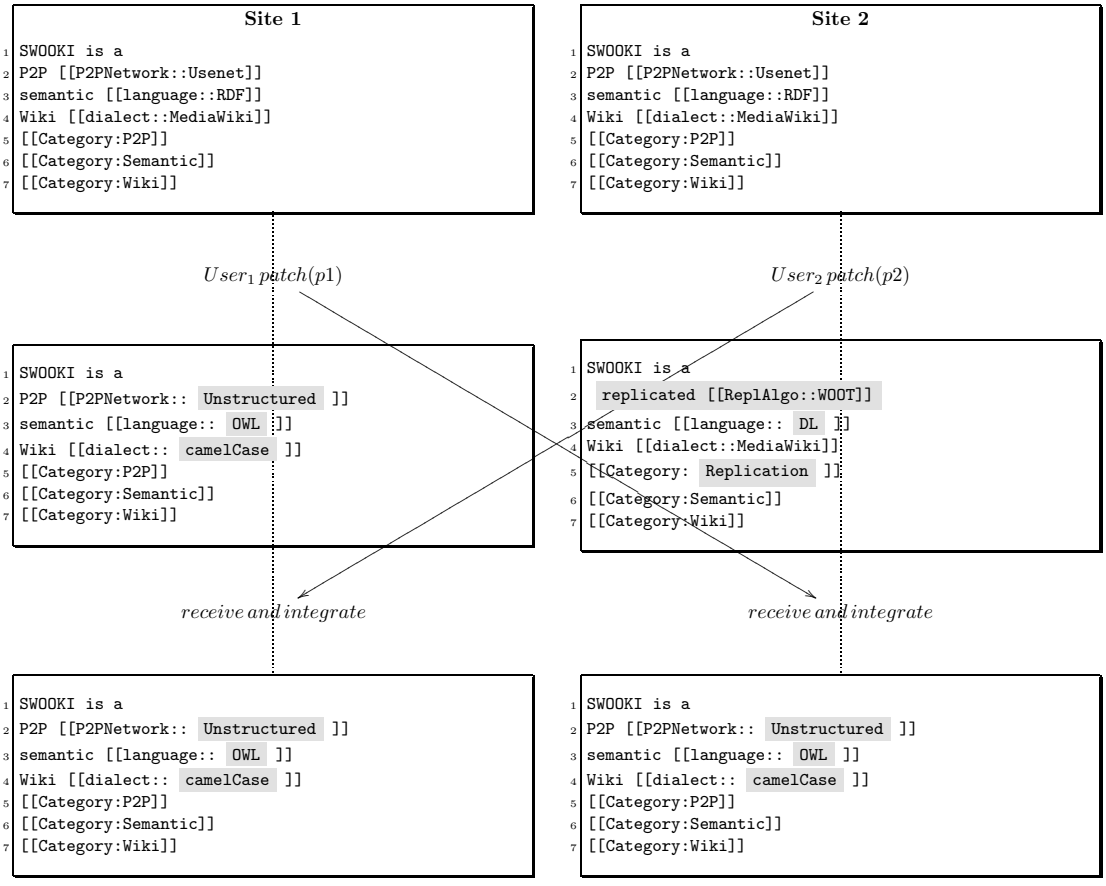

Figure 2: Concurrent editing with Thomas's rule

The "last writer wins strategy" has been introduced in 1976 by Johnson and Thomas [19]. This replication algorithm ensures eventual consistency [20] for replicated data. It has been designed for large scale replicated systems. Each site maintains a set of pairs $(identifier, value)$. This can be easily extended to data such as RDF triples $(subject, predicate, object)$. Each modification is

decorated with 2 timestamps, namely the creation timestamp and the last modification timestamp. In case of concurrent updates on the same data (same data is detected by comparing identified creation timestamps), the incoming change is applied only if the last modification timestamp is more recent than the current one on the local copy.

In figure 2, we apply the rule of Thomas to merge concurrent editing [19]. Starting from a common initial text, two users edit concurrently (collaboratively) a wiki page about SWOOKI, each user being connected on a different site.

At site1, User1 updates the *P2PNetwork* property to $[[P2PNetwork :: Unstructured]]$, the semantic annotations *language* property to $[[language :: OWL]]$ and the *dialect* property to $[[dialect :: camelCase]]$. All these edits are highlighted in figure 2. A patch p_1 is the new content of the page generated by modifications made by User1. p_1 is executed locally at site1 and sent with its new last time modified timestamp to site2 in order to be integrated.

Concurrently, at site2, User2 replaces original line 2 by a new line "replicated [[ReplAlgo::WOOT]]". She updates the language property to DL and changes category P2P to replication. A patch p_2 generated by modifications made by User2 is locally executed at site2 and sent with its last modified timestamp to site1 in order to be integrated.

If we consider a wiki page as a pair $(pageid, value)$, the two concurrent patches p_1 and p_2 (see figure 2) will be decorated by a creation timestamp represented by *pageid* and a last modification timestamp. In our example, we suppose that p_1 is more recent than p_2. When p_2 is received on site 1, it is simply ignored. When p_1 is received on site 2, it is more recent than the current version of the page, so p_1 is applied on site 2. This strategy ensures eventual consistency and ensures also that any visible page on a wiki server has been reviewed by a human.

Of course, the main problem of this strategy is that the effect of p_2 is not visible on the current state of the wiki page. It requires an awareness system to aware users about this concurrent editing (this awareness aspect is not currently managed by the Thomas's write rule). User should perform a manual merge using versions of the wiki page to integrate some concurrent modifications. This is not really compatible with simplicity principle of wikis. This is also not compatible with the concept of user intentions preservation [5] of groupware system. User intentions preservation property ensures that an effect observed on one site at generation time must be observed on all sites. The Thomas's write rule strategy is not compatible with user intention preservation.

4.2 Woot strategy

WOOT manages a wiki page as a sequence of lines. WOOT considers updating of a line $l1$ as a deletion of the old line $l1$ followed by an insertion of a new line with the new content between the line before $l1$ and the line after $l1$. In order to ensure eventual consistency, WOOT never really deletes lines. Deleted lines are still maintained in the model of the page but they are just not invisible for the user.

In figure 3, we execute the same scenario of the figure 2,by using the WOOT[13] algorithm to perform the merge.

At both sites, the modifications made by each user are detected by the system as patches [2], p_1 on site1 and p_2 on site 2. Each patch is executed locally at a site and sent and integrated on the other site.

Due to the property of convergence of WOOT, both sites converge to the same final result. We describe the behavior of concurrent editing using the Woot algorithm at line level. It is also possible to use the Woot algorithm at word level or at character level. We made this choice for simplicity of reading.

The quality of the result of the WOOT algorithm is subject to discussion. However, WOOT algorithm preserves user intentions, all concurrent effects are visible in the final version of the page. The main disadvantage is that this page has not been reviewed by a human and cannot be considered as safe. The concurrency awareness [8] mechanism will mark the page as "server

[2] A patch is a delta between two successive versions of a wiki page and contains the sequence of elementary operations required to transform one version into another.

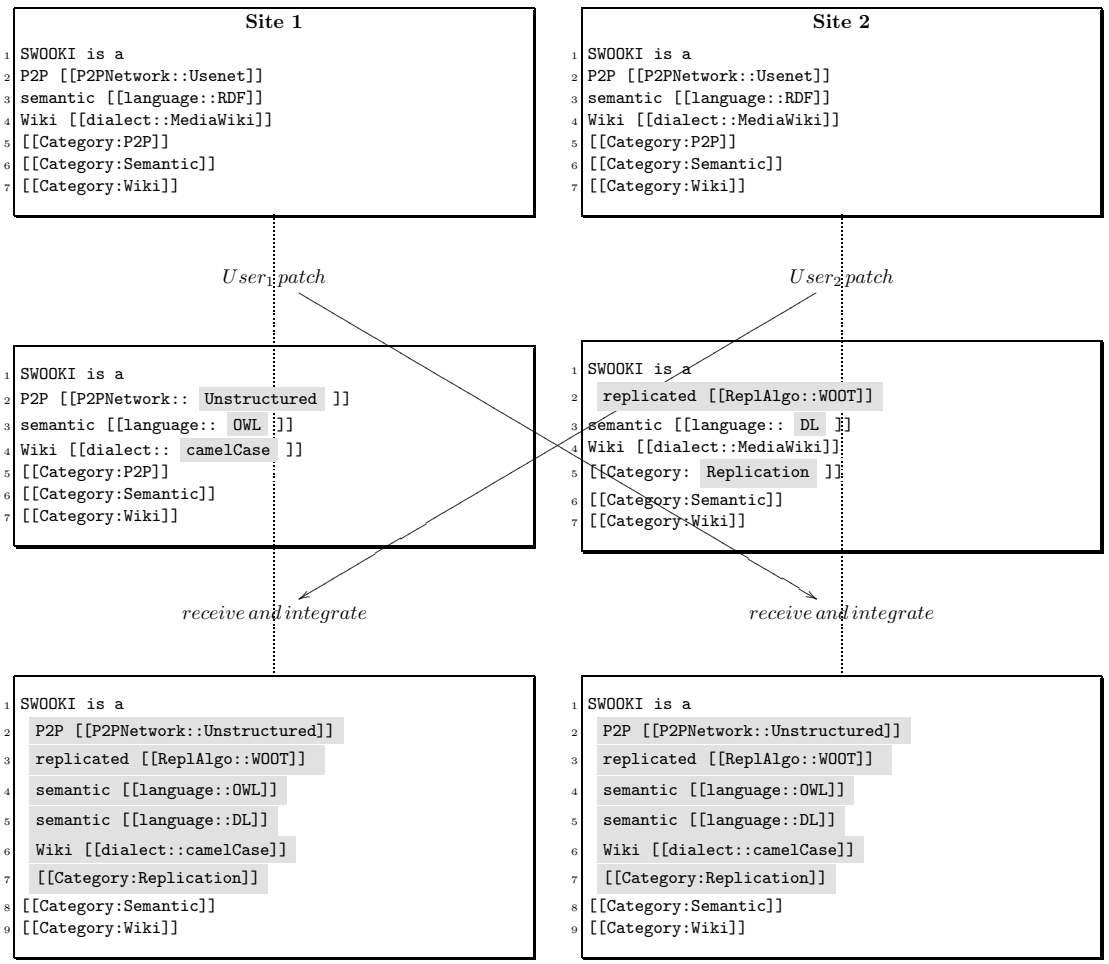

Figure 3: Concurrent editing with WOOT algorithm

produced" and highlight all concurrent operations in the final page. In figure 3,the highlighted lines in the final state reflect the concurrent modifications produced. These highlights are generated by the concurrency awareness mechanism. The main problem with the WOOT approach is that a wiki page is merged as a sequence of lines without taking into account the nature of semantic data and without any human reviewing.

Even if the result in figure 2 looks more accurate than this of figure 3, both pages are generated by an automatic merge and must be reviewed by a human.

4.3 Mixing WOOT and Thomas

In figure 4, we execute the scenario of the figure 2 and 3, but here we apply the Thomas's write rule for merging operations on semantic annotations and WOOT for managing operations on text.

If we try to apply the principle of user intentions to semantic annotations, we find that it is impossible to preserve intentions. The original definition [5] is :

- (Intention of an Operation). The intention of an operation O is the execution effect which can be achieved by applying O on the document state from which O was generated.

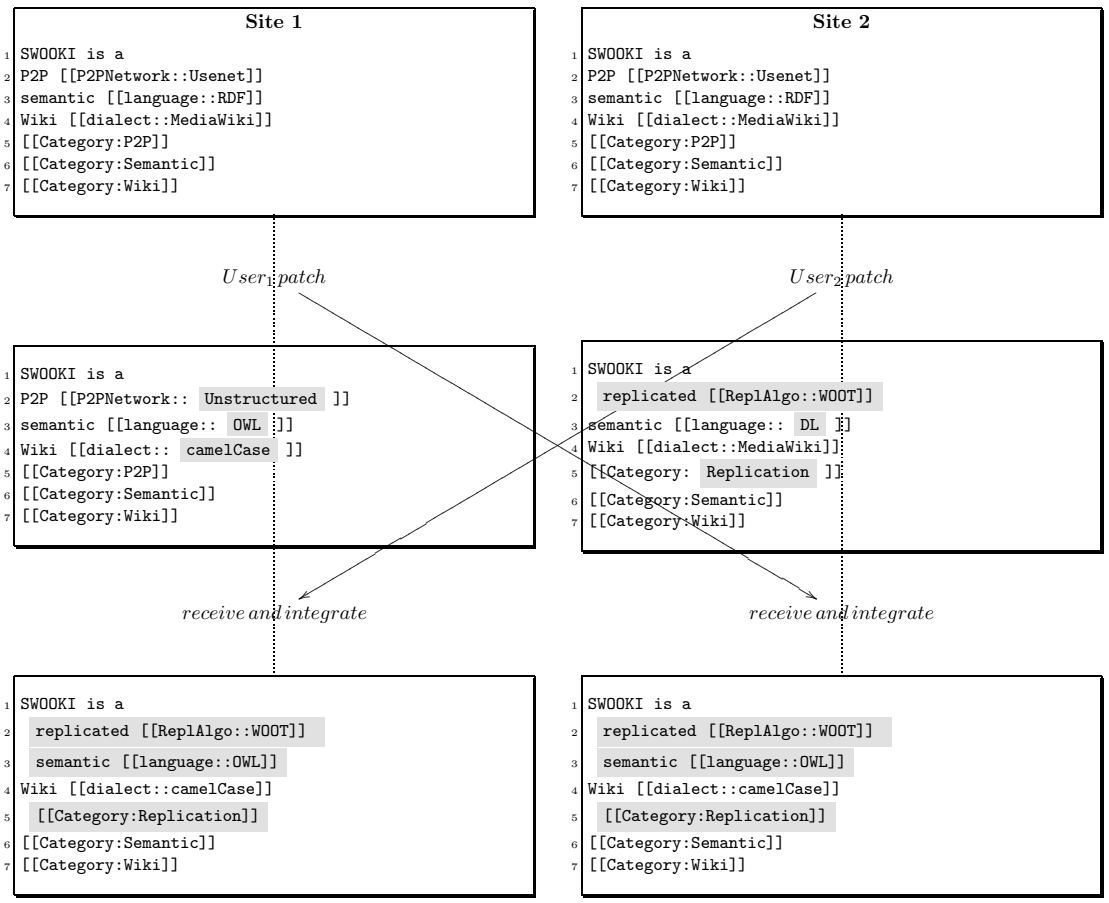

Figure 4: Concurrent editing with WOOT for text and Thomas's rule for semantic

- (Intention preservation) : For any operation O, the effects of executing O at all sites are the same as the intention of O, and the effect of executing O does not change the effects of independent operations.

For example, let us consider the operation $updateProperty(pname, value)$. An instance of this operation is generated at site 1 with $op_1 = updateProperty(\text{“}language\text{”}, \text{”}OWL\text{”})$. The intention of the operation is obviously to replace the old value "RDF" by the new value "OWL". Concurrently, $user_2$ on site 2 is executing the operation $op_2 = updateProperty(\text{”}language\text{”}, \text{“}DL\text{”})$. The intention of op_2 is to replace "RDF" by "DL". It is not possible to preserve intentions of both operations according to the above definition. This means that intention preservation is just not defined for semantic annotation data types with an operation $updateProperty(pname, value)$. If intention preservation cannot be defined for semantic annotation, then the Thomas's write rule is an acceptable algorithm for managing semantic annotations.

Now, we consider a wiki page as a sequence of lines. Inserting a line l between two consecutive lines $l1$ and $l2$ can be represented as an operation $op = ins(l1 \prec l \prec l2)$. The intention of op is to observe l between $l1$ and $l2$. Suppose two concurrent operations $op_1 = ins(l1 \prec \text{”}semantic[[language : OWL]]\text{”} \prec l3)$ and $op_2 = ins(l1 \prec \text{”}semantic[[language : DL]]\text{”} \prec l3)$.

Next, if we apply both operations, we obtain a state where the effect of op_1 and the effect of op_2 are observed between $l1$ and $l3$. For this insert operation, the notion of intention preservation is defined and can be maintained by the system.

As a consequence, we believe it is reasonable to use WOOT for managing text and Thomas's write rule for managing semantic annotations.

In this scenario, two types of patches are generated, patches corresponding to operations on the semantic annotations and patches corresponding to the operations on lines.

We can explain the final result of figure 4 as follow:

- The "Unstructured" property does not appear because it has been applied to the original line $l2$. Line $l2$ has been marked as invisible because it has been replaced by line "replicated [[replAlgo::WOOT]]".
- On line 3, we applied the last writer wins strategy and we consider that site 1 won.
- On line 5, there is no particular problem.

Also in this scenario the final result of the page is generated by an automatic merge and must be reviewed by a human.

4.4 Discussion

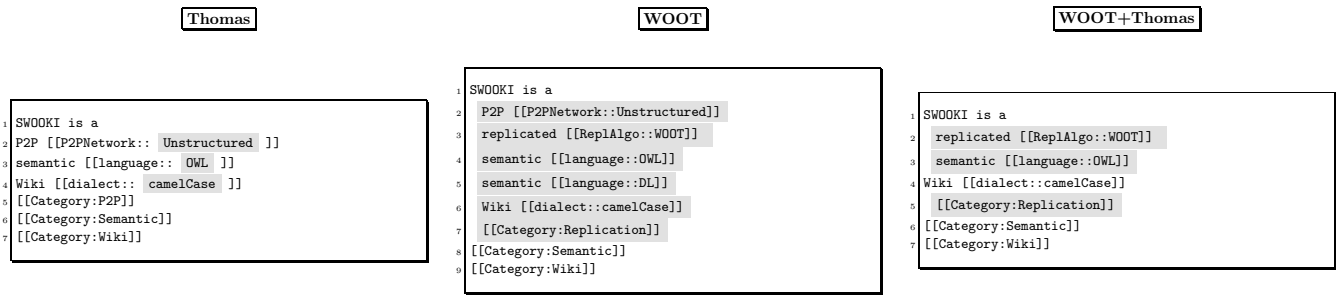

Figure 5: Comparison of merge strategies : Thomas - WOOT - WOOT+Thomas

Figure 5 presents the different results of the same scenario. The left part of the figure is the result with Thomas's write rule, the center part with WOOT and the right part with WOOT+Thomas. At the first sight, it seems that the smarter merge is the result of the application of WOOT+Thomas strategy.

In all scenarios, the results are produced by the server, it needs a human for reviewing the final state. With Thomas's rule, the user has to integrate concurrent changes *aposteriori*. With WOOT and WOOT+Thomas, the user has to verify the accuracy of the result.

If we consider the cognitive effort for the user, we believe that the best solution is to see all concurrent changes. With the Woot strategy all concurrent changes are really visible in the wiki page. If the user wants to change the result of the merge, he can do that easily. With the two others solutions, user needs to access versions in the wiki page history.

5 Conclusion

Swooki is the first attempt to build a peer to peer semantic wiki. It follows the *use wikis to ontologies* approach. Swooki provides the same functionalities of any server-based semantic wiki due to the total replication of the data and its merging algorithm. In this paper, we focus on merging of semantic data. Compared to traditional semantic wikis, SWOOKI allows massive collaboration, off-line editing and ad-hoc collaboration.

Integrating semantic annotations into a P2P wiki can be easily done. At first sight, managing semantic annotations with Thomas's write rule and textual part with WOOT looks like a good solution. We believe that managing all the wiki pages as text can ease the review process of merged pages. Choosing this solution allows to build a P2P semantic wiki very easily. We are

currently working on the implementation of the ideas presented in this paper. We need to carry out user studies to validate our ideas.

P2P semantic wiki can support a large number of users by using our approach. This approach allows to balance the load of queries. However, it does not solve the problem of scalability for queries and reasoning. We believe that we provide a cheap way to have many replicas of the same semantic wiki. Maybe this total replication of semantic data can be used to distribute semantic queries on different replicas.

References

[1] Buffa, M., Gandon, F.: Sweetwiki: semantic web enabled technologies in wiki. In: WikiSym'06: Proceedings of the international symposium on Symposium on Wikis, New York, NY, USA, ACM Press (2006) 69–78

[2] Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic Wikipedia. Proceedings of the 15th international conference on World Wide Web (2006) 585–594

[3] Weiss, S., Urso, P., Molli, P.: Wooki: a p2p wiki-based collaborative writing tool. In: Web Information Systems Engineering, Nancy, France, Springer (2007)

[4] Morris, J.: DistriWiki:: a distributed peer-to-peer wiki network. Proceedings of the 2007 international symposium on Wikis (2007) 69–74

[5] Sun, C., Jia, X., Zhang, Y., Yang, Y., Chen, D.: Achieving Convergence, Causality Preservation, and Intention Preservation in Real-Time Cooperative Editing Systems. ACM Transactions on Computer-Human Interaction **5**(1) (1998) 63–108

[6] Molli, P.: Optimistic data replication on decentralized and p2p systems (invited talk). 6th franco-mexican school on distributed systems, Colima, Mexico (2007)

[7] Ignat, C., Oster, G., Molli, P., Cart, M., Ferrie, J., Kermarrec, A.M., Sutra, P., Shapiro, M., Benmouffok, L., Busca, J.M., Guerraoui, R.: A comparison of optimistic approaches to collaborative editing of wiki pages. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). Number 3, White Plains, NY, USA (2007)

[8] Alshattnawi, S., Canals, G., Molli, P.: Concurrency awareness in a p2p wiki system. In: Proceedings of CTS 2008, The 2008 International Symposium on Collaborative Technologies and Systems, Irvine, California, USA. (2008)

[9] Spencer, H., Lawrence, D.: Managing Usenet. O'Reilly Sebastopol (1998)

[10] : Gears - enabling offline web applications. http://gears.google.com/ (2008)

[11] Reliable Software: Code Co-op. (2006) `http://www.relisoft.com/co{_}op`.

[12] 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)

[13] Oster, G., Urso, P., Molli, P., Imine, A.: Data consistency for P2P collaborative editing. Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (2006) 259–268

[14] Buffa, M., Gandon, F.L., Ereteo, G., Sander, P., Faron, C.: Sweetwiki: A semantic wiki. J. Web Sem. **6**(1) (2008) 84–97

[15] Tazzoli, R., Castagna, P., Campanini, S.: Towards a semantic wiki wiki web. Proceedings of the International Semantic Web Conferenc (ISWC) (2004)

[16] Souzis, A.: Building a Semantic Wiki. Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications] **20**(5) (2005) 87–91

[17] Vrandecic, D., Krötzsch, M.: Reusing ontological background knowledge in semantic wikis. In Völkel, M., Schaffert, S., eds.: SemWiki. Volume 206 of CEUR Workshop Proceedings., CEUR-WS.org (2006)

[18] Schaffert, S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. 1st International Workshop on Semantic Technologies in Collaborative Applications (STICAÇ06), Manchester, UK (2006)

[19] Johnson, P., Thomas, R.: RFC0677: Maintenance of duplicate databases. Internet RFCs (1975)

[20] Saito, Y., Shapiro, M.: Optimistic replication. ACM Comput. Surv. **37**(1) (2005) 42–81

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399