Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 244 (2009) 81–94

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

# Reengineering the Wikipedia for Reputation

Thomas Rune Korsgaard[1] , Christian D. Jensen[2]

*Informatics and Mathematical Modelling*
*Technical University of Denmark*
*Kgs. Lyngby, Denmark*

**Abstract**

The Wikipedia is a free online encyclopedia collaboratively edited by Internet users with a minimum of administration. Anybody can write an article for the Wikipedia and there is no verification of the author's expertise on the particular subject. This may lead to problems relating to the quality of articles, especially completeness and correctness of information, and inaccuracies in the Wikipedia have been rumoured to cause students to fail courses; innocent people have been associated with the killing of John F. Kennedy, etc. Providing a means to assess the correctness, completeness and impartiality of information in the Wikipedia is therefore vitally important for the users to build trust in the Wikipedia and ensure the continued success and growth of the system.

Integrating a reputation system into the Wikipedia would help users assess the quality of articles and provide a powerful incentive for authors to improve the quality of their articles. There are currently more than 7.5 million articles in the Wikipedia, and more than a thousand new articles are added daily, so the investment in the existing system is significant. The introduction of a recommendation system should therefore not require any modifications to the existing Wikipedia software.

In this paper we examine the problem of reengineering a large and popular system, in this case the Wikipedia, in order to include a reputation system. We propose a recommendation system, which allows Wikipedia users to calculate a personalised rating for any article based on feedback (recommendations) provided by other Wikipedia users. The recommendation system developed for the Wikipedia is based on a general architecture, which we believe applies to many existing applications for online collaboration. The proposed recommendation system is implemented in a proxy placed between the user's web-browser and the Wikipedia server, e.g., on the user's own machine, so there is no need to modify Wikipedia servers or software. A simple prototype of the proposed recommendation system is presented in this paper along with a preliminary evaluation of the prototype.

*Keywords:* Wikipedia, trust, reputation, filtering, recommender system

## 1 Introduction

The Wikipedia [6] is a free online encyclopedia collaboratively edited by Internet users around the world. The Wikipedia is based on the Wiki philosophy [22], where everybody is encouraged to contribute to a collaboratively authored document. This means that all articles in the Wikipedia can be edited by everybody, so content can be added, modified or deleted by anybody, but the Wiki system makes it easy to

[1] Email: thomas@korsgaard.eu

[2] Email: cdj@imm.dtu.dk

doi:10.1016/j.entcs.2009.07.040

restore documents to their previous state if modifications are considered undesirable. No prior qualifications are needed to write an article, so the Wikipedia is able to harness the collective intelligence of all Internet users, by allowing anyone who feels knowledgeable in an area to create or correct articles about subjects in that area. This increases the chances of finding an article on a given topic, even though there might only be a few experts in the world on this topic, so the Wikipedia is quickly replacing traditional printed references as a primary source of information.

The open and flexible nature of the Wikipedia has, however, also exposed one of the weaknesses of collaborative authoring, which is that malicious or incompetent users may compromise the integrity of the document by introducing erroneous entries or corrupting existing entries. Jimmy Wales, the co-founder of the Wikipedia, claims to receive 10 emails every day from students who failed their courses because the information cited from the Wikipedia turned out to be wrong [18]. Since many Wikipedia articles are written by ordinary people, they may not present an objective account of the subject. This is especially true when an article's topic is political, religious or racial in nature or in another way reflects social and cultural norms. This is best illustrated by the fact that a range of IP-addresses from the US Congress was banned from editing the Wikipedia in January 2006, because both the House and the Senate had been treating the Wikipedia as a personal battleground, fighting turf wars and repeatedly altering content about Congressmen listed on the site [21]. Another example of a misuse of the Wikipedia was the a conflict between Adam Curry and Dave Winer, who both believed themselves to be the father of pod-casting. An anonymous IP address kept making changes to the article on pod-casting leaving only Curry as the father of pod-casting; the IP address was later traced back to Curry [19]. It therefore appears obvious that a general mechanism to improve the quality of documents produced in open collaborative authoring systems is needed.

Reputation systems, where users provide feedback on the quality of the articles they read, provide a powerful incentive to improve the quality of the articles and help users avoid articles that are of low quality. In order to preserve both the large investment that authors have made in terms of time and effort and the familiarity of the user interface for occasional users, the collaborative authoring system must be considered a legacy system that cannot be modified and the reputation system should only be offered to users who opt in. This means that the reputation system should be offered as an external service, which users can include as an intermediary between their browser and the collaborative authoring system, in this case the Wikipedia.

This paper examines the problem of reengineering a large and immensely popular legacy application to include a reputation system. The paper presents a reputation system for the Wikipedia, which provides a central repository for feedback that allows individual users to calculate their own subjective reputation value for a given article in the Wikipedia. The system does not calculate or distribute reputation values, but simply the recommendations (signed feedback) from other users, so we refer to our system as the Wikipedia Recommender System (WRS). Through the recommendations from other people, the local component of the recommender system on the user's own machine is able to calculate a rating, which indicates the quality of the article. The recommender system then receives feedback from the reader, which allows the WRS to determine whether the recommendations were useful and to identify the recommenders whose feedback coincided with the user. The recommender system uses this information to update the active users profile, in order to decide how recommendations should be interpreted and provide a more precise recommendation next time. A proxy based prototype of the WRS has been developed, which allows us to evaluate the feasibility of the proposed architecture. Preliminary experiments indicate that the computational overhead involved in verifying the recommendations and storage overhead needed by the recommendations are acceptable.

The rest of this paper is organised in the following way: Section 2 examines the problem of reenginering a legacy software system to include a reputation system. An overview of the Wikipedia and the structure of Wikipedia articles is presented in Section 3. An outline of the Wikipedia Recommender System is presented in Section 4. Section 5 presents our preliminary evaluation of the developed prototype and Section 6 presents a discussion of related work. Finally, our conclusions and some directions for future work are presented in Section 7.

## 2 Reengineering for reputation

Traditional reputation systems [14] are either closely integrated with the application in which they are used [2,5] or they are implemented by an external reputation server, which collects and collates feedback from users and calculates a reputation value [8,4]. The first solution generally requires significant modifications to an existing system, so we focus on the second solution in this paper. A generic architecture for an external reputation system is shown in Figure 1.

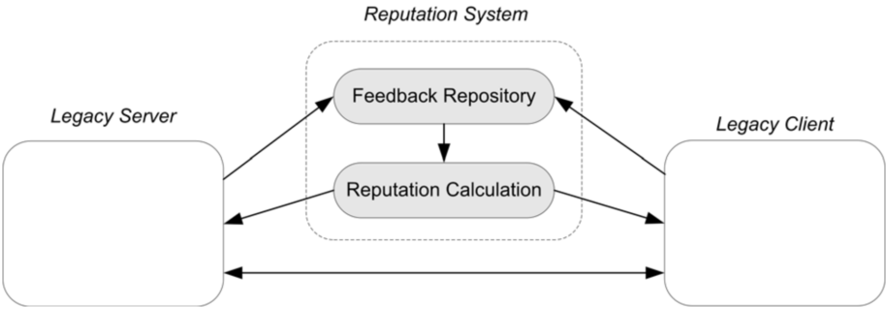

Fig. 1. Generic reputation system architecture.

The figure shows a legacy application consisting of a client and a server [3] and an external reputation system. Both clients and servers may request reputation ratings about the other party from the reputation system in order to decide whether they

[3] This example trivially extends to most distributed application architectures, because one party generally initiates the interaction by requesting a service and the other party responds to this request.

should interact with the other party. If both parties are satisfied with the reputation of the other party, the direct interaction can take place according to the existing protocols of the legacy system. Once the result of the interaction is known, each party may return feedback to the reputation system, which stores it in the feedback repository. Records of relevant interactions, stored in the feedback repository, are used to calculate the reputation values that are forwarded to clients and servers upon request. The functionality required to request reputation ratings from and return feedback to the reputation system can either be implemented directly in the communication primitives of the legacy application, in the middleware used to connect clients with servers or as wrappers that encapsulate both clients and servers.

Reputation is always linked to an identity, so both parties require a mechanism to verify the identity of the other party. Most distributed systems, however, already rely on authenticated identities when granting access to protected resources, so it is often possible to base the reputation mechanism on these identities.

The generic reputation system architecture presented above does not have to be embodied directly in a single server. Instead, both the feedback repository and the reputation calculator may be implemented by either a centralised or distributed system. The actual locations of the functional components depend on the architecture of the system being reengineered.

We have thus defined the following logical components that have to be considered when a legacy system is reengineered for reputation:

**Feedback Repository** The reputation system needs a mechanism to store opinions about other users and store feedback from interactions with other users. The structure of this feedback repository may either be centralised or distributed.

**Reputation Calculation** The reputation system must be able to calculate a reputation value based on the opinions and feedback stored in the reputation repository. This calculation may either be performed in a central component, where one algorithm is used to calculate the reputation for all agents, or distributed in many components, where each component is free to implement their own reputation calculation algorithm. It is, however, important that all reputation calculation components have full access to the feedback repository.

**Identity Management** The types of identities used in a reputation system include from authenticated physical identities, semi-permanent identifiers, such as IP-addresses, or virtual identities,[4] such as nicknames or pseudonyms. It is important to note that interactions based on reputation do not necessarily require authentication of a legal identity; it is often sufficient to recognise the virtual identity of the other party [20].

**Interception Mechanism** It will be necessary to modify some parts of the legacy system in order to interface with the reputation system. This modification may

[4] A virtual identity is an identifier that may be individually chosen by the agent and cannot necessarily be authenticated through a third part, e.g., if an agent generates a public-/private-key pair, publishes the public-key and consistently signs all messages with the private-key, then the public-key may be considered a virtual identity.

include rewriting the application code, changing the middleware, add new aspects [15], implement interposition agents [13] or interpose a set of proxies between the client and the server.

# 3 The Wikipedia

The basic philosophy behind the Wikipedia is that everyone should be allowed to edit everything, but that it should be easy to restore the document to its prior state if the modifications are considered undesirable. Detection is left to the users and the only means of response is to restore the previous page.

## *3.1 Wikipedia Architecture*

The Wikipedia is accessed through an ordinary web browser and the Wikipedia appears, to the user, as an ordinary website. The Wikipedia HTML pages are dynamically created by the MediaWiki software. MediaWiki is written in the PHP programming language, and can use a relational database management system to store the articles.

An article on the Wikipedia has several sub-pages, which are useful in the WRS. The structure of these sub-pages is shown in Figure 2.

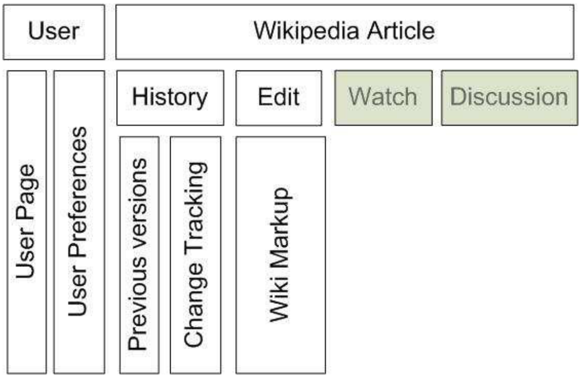

Fig. 2. Simplified structure of the Wikipedia articles

Each article has the main article presented to the viewer. Furthermore there are 4 pages that are related to each article: the history page, the edit page, the watch page, and the discussion page. The watch page is not visible unless the user is logged in. The edit page allows an active user to alter the content of the article without these modifications being visible to everyone before the page is saved. The history page provides access to previous versions of the page. Every prior version of an article, up to the existing version of the article, can be found on the history page. The history page also offers the functionality of reverting the page to a previous version if the current version has been vandalised, and of showing the difference between two versions. Furthermore, there is a user page available to users who are logged in. The userpage allows the user to set preferences for the Wikipedia and to keep a personal homepage. According to the Wikipedia philosophy other users may still edit the content of the user page.

### *3.2 Trust in the Wikipedia*

Trust in the Wikipedia is ultimately a question about the quality of the articles that it contains. The quality of a Wikipedia article is determined by a few simple properties, i.e., is the article complete, correct and unbiased. However, these properties are difficult to determine automatically and despite some promising work in this area [12,23], we do not believe that these techniques are sufficiently mature at the moment. Instead, we propose to rely on feedback from the users, i.e., to use some reputation system similar to the ones used by Amazon [2], IMDb [5] or the “WOT” plugin for Firefox [8]. A reputation system cannot prevent undesirable content from entering the Wikipedia, but it may help readers assess the quality of Wikipedia articles and allow them to decide whether to believe the article or look for more reliable information elsewhere. Moreover, the introduction of a reputation system is in line with the Wiki philosophy, where we find few mechanisms to prevent malicious or accidental modification of a Wiki page; detection is left to the users and the only means of response is to restore the previous page.

### *3.3 Reputation in the Wikipedia*

The number of articles in the Wikipedia is growing rapidly. There are currently close to 1.8 million articles in the English Wikipedia alone and around 2,000 new articles emerge every day [7]. It appears obvious that modifying such a large and dynamic system is difficult and should therefore not be attempted unless absolutely necessary, so the existing Wikipedia software base should be considered as a legacy system that cannot be modified. Fortunately, the functions of a reputation system is orthogonal to the basic functions of the Wikipedia, so it is possible to provide these functions, i.e., management of feedback and calculation of reputation values, from external servers.

## 4 Wikipedia Recommender System

The Wikipedia Recommender System (WRS) has been designed to operate on top of the existing Wikipedia without requiring modifications to the MediaWiki installation or the underlying Wiki engine.

### *4.1 System Overview*

Implementation of a recommender system on top of a legacy web-based system requires the ability to rewrite the content read from the server (to insert the recommendations) and to capture and store the feedback from the clients. A simple way to do this is to insert a web-proxy between the user and the Wikipedia. This architecture is shown in Figure 3, where the proxy executes on the user’s own computer along with the browser.

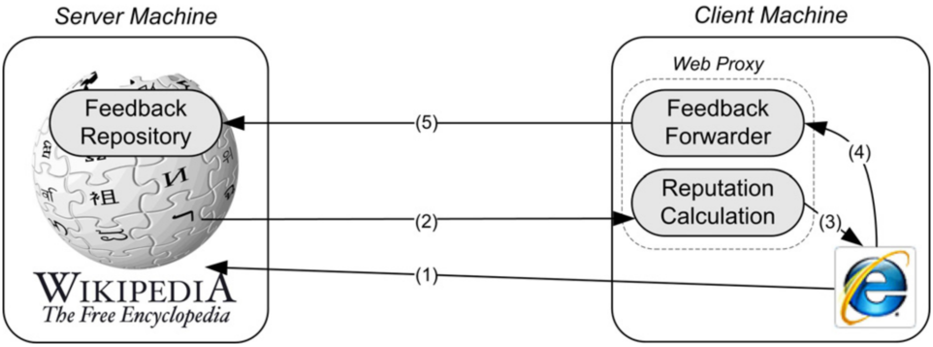

Fig. 3. Overview of the Wikipedia Recommender System

The browser must be configured to use the local web proxy (this is how users opt in), which intercepts all requests to the Wikipedia (1). The proxy retrieves the article from the Wikipedia (2) along with the feedback (cf. Section4.2), which is used to calculate the reputation score for the article. The page [5] is rewritten to include the reputation score and forwarded to the browser (3). The user now has an indication of the quality of the page and may decide to provide feedback regarding the quality of the page and the utility of the reputation rating (4). The user's indication of the utility of the score is used by the proxy to build trust in the other Wikipedia users who recommended this article and the user's own rating is stored in the feedback repository in the Wikipedia (5). The different components are described in greater details in the following.

### *4.2 Recommendation Repository*

The recommender system requires a repository that stores feedback (the recommendations) from users and makes these recommendations available to other users. This repository may either be implemented as a distributed database running on all the clients or as a single centralised database running on a separate server. Neither of these solutions are desirable because they introduce additional complexity and dependencies into the system. However, the Wikipedia itself provides a repository of information that can be read and modified by all its users, so it should be possible to store recommendations on the Wikipedia servers. We do not wish to modify the WikiMedia system to include recommendations, but instead propose to store recommendations as HTML comments on the edit or discussion pages of the Wikipedia articles. This allows us to store the recommendations in a central location, available to everyone, without modifying the WikiMedia system or interfering with Wikipedia users who do not wish to avail themselves of the recommender system (the HTML comments will not be rendered by the user's browser). This solution also allows the WRS, running on the user's proxy, to create a subset of these ratings [6] based on trust in the recommender and calculate a reputation value for the article based on

[5] Articles in the Wikipedia are contained in web pages, so we generally use the term *article* to refer to the logical content and *page* to refer to the physical data structure.

[6] The terms recommendations and ratings will be used interchangeably in this paper.

these trusted recommendations (cf. Section 4.3).

As Wikipedia articles, including the HTML comments, can be modified by anyone, so ratings have to be secured against masquerading, modification and deletion. Masquerading and modification can be prevented by introducing cryptographic measures, e.g., the user signs the content of the rating, which means that we need a repository for public-keys for all users who provide recommendations; this is examined in greater detail in Section 4.4. The Wikipedia philosophy states that everybody can delete everything, which makes the WRS open to DoS attacks. However, once the ratings have been set on a page, they will always be available through the history page, where the prior versions are stored. Normally when a page is vandalised the page restore quite fast [3].

### *4.3 Reputation Calculation*

The calculation of a reputation value for an article is based on the reputation repository, which stores all the ratings that other users have given the article. Each rating consists of five elements:

**The mark** The rating that the user has given the article.

**The user** The registered Wikipedia user name of the user who gave the mark. The name is chosen by the user when registering with the Wikipedia, but there may be no link to the user's real identity.

**The version** The version number of the article that the rating relates to.

**The article name** The name of the article is inserted into the rating, in order to prevent that ratings are copied to other articles.

**The hash** The hash protects the integrity of the the user name, the mark and the version. The title of the page, the mark, and the version are concatenated and signed with self-signed certificate, where the public key is kept at the user's personal user page. This prevents ratings from being tampered with, moved to other pages or moved to a later version of an article.

The first three elements are used to calculate the rating, while the two last elements are included to ensure the integrity and authenticity of the rating.

There are many possible ways to calculate a reputation value based on the information contained in a rating. The simplest way would be to calculate the arithmetic average of all the marks, but we believe that this is too simple because it does not allow for ethnic, religious, cultural or political differences between the users of the WRS. Instead, we proposed to calculate a weighted average of the marks, where the weights reflect the trust in the recommender (the Wikipedia user name stored in the rating).

In order to calculate these weights, the proxy maintains a list of *friends*[7] that include users whose recommendations have been used in the past. The list of friends

[7] We use the term friends to denote recommenders that have been seen before and whose recommendations have been classified as useful. Real friendship or simply knowledge about the other user's identity is not assumed.

is stored in a database maintained in the local proxy, which we refer to as the *Ring of Reviewers* (RoR).

When a user provides feedback about an article, the WRS will examine the user's RoR and update the trust values of the friends who provided recommendations. The trust value of recommenders that have provided recommendations similar to the users feedback will increase, while the trust in friends with very different ratings will fall. The indication of utility provided by the user along with the feedback helps determine how different the two ratings have to be in order to decrease trust in a recommender.

Another problem that has to be considered is that the contents of an article change over time, so ratings that were submitted for a prior version of an article may no longer be valid, because the contents of the article have changed. It is obvious that ratings should still be valid if the changes are minor (e.g., typos, adjectives, punctuation etc), but the rating should no longer be valid if the changes are extensive or contains words that may completely change the meaning of the text (e.g., words like not, no, don't, without, does not, un-). Determining the semantic difference between two documents is a difficult problem, which is beyond the scope of this paper.

In the current implementation, we have implemented a simplistic approach based on a threshold, which defines a limit to acceptable change. Preliminary experiments have indicated that texts remain fairly similar if less than 15% of the text is changed, but this does not account for negations or changing the names of people or places mentioned in the text. It is, however, important to note that the history page provides a complete revision history, which makes it easier to identify modifications and determine whether changes are likely to have altered the meaning significantly. Moreover, we believe that techniques developed in the areas of plagiarism and copy detection [11] may also help solve this problem, which we plan to address in future work.

### *4.4 Identity Management*

Most authors who contribute to the Wikipedia creates an account, which gives them a name and provides them with their own user page. It is therefore natural to use this Wikipedia user name as the identity in the reputation system and require that all users of the WRS register with the Wikipedia first.

Registering with the Wikipedia gives both a user name and an individual page for the user on the Wikipedia, the so-called user page. This is a standard Wiki page that is accessible to all users of the Wikipedia, so it provides a convenient vehicle for distribution of cryptographic keys.

All users of the WRS need a public-/private-key pair to sign ratings and allow other WRS users to verify their signatures, so before being able to make recommendations, the user must generate a public-/private-key pair and upload the public-key to their user page. The user page is a standard Wiki page, which anybody can edit, so it is important that other users (and their proxies) only use the latest public-key uploaded by the owner of the user page; this may require a search of the history pages as well. It is possible to verify that the public-key was uploaded by the owner, because the Wikipedia records the user name of the person modifying a page and logging in to the Wikipedia under a given user name requires a standard login/password authentication.

### 4.5 Interception Mechanism

The Wikipedia Recommender System only uses recommendations to provide feedback about the quality of articles. This means that we only need to intercept interactions on the client side, which is implemented through a proxy running on the users machine[8] as shown on Figure 3.

When the proxy receives a HTTP request, it first determines whether the request is for the `wikipedia.org` domain and simply passes it on if it is not. If the request is to the Wikipedia domain, the proxy retrieves the HTML page from the Wikipedia server and extracts the URL of the edit page and the history page. The edit page is used to obtain the ratings and the history page is used to retrieve previous versions of pages in order to check if recommendations given to earlier versions of the article should still be considered for the existing page.

The edit page is passed on to the Reputation Calculator, which extracts the recommendations from the HTML page, verifies the recommendations (cf. Section 4.3) and stores the verified recommendations in a temporary Session Rating Database. The Rating Calculator module now calculates a personal rating based on the ratings in the temporary Session Rating Database and the trust values in the RoR. This rating is inserted into the HTML page along with the feedback option to the user and the HTML page is passed on to the browser.

The user now has the choice to decide on whether to give feedback or not. This feedback will be used to update the trust values in the RoR and store a recommendation in the Wikipedia.

A complete description of the design and implementation of the Wikipedia Recommender System is found in Korsgaard's M.Sc. Thesis [16].

## 5 Evaluation

We have implemented a simple prototype of the Wikipedia Recommender System, which allows us to determine the feasibility and performance of the proposed architecture.

### 5.1 Impact on the Wikipedia

The size of the ratings depend on the length of the username and the title of the article that is rated. The average size of usernames and article titles, indicates that a rating has an average size between 110 and 140 characters. Assuming a worst

[8] In this paper we generally assume that the proxy runs on the user's own computer, but it is possible to have the proxy hosted by an external service provider, which makes it accessible to the user when she is not using her own machine, e.g., from an Internet Café.

case scenario, we presume that a rating is 140 characters. If all of the articles on the English Wikipedia, has 20 ratings each. This gives us:

$$1.8 \text{ million articles} \cdot 20 \text{ ratings pr. article} \cdot 140 \text{ bytes} = 5040 \text{ million bytes}$$

An estimated 5 GB will be spent on storing the ratings in the Wikipedia. From the MediaWiki Foundation download page [1], a compressed database dump can be downloaded. This dump takes up 84.6 GB, and users are warned that uncompressed it will take up 20 times more space. This gives an uncompressed size of the Wikipedia of 1692 GB. This suggests that the ratings will take op 0.3% of the total size of the Wikipedia, which we regard as acceptable. Even with 100 ratings per article, the recommendations would only account for 25 GB, which corresponds to 1.5% of the total storage requirement of the Wikipedia.

### *5.2 Benchmarking the Proxy*

In the prototype implementation the free open source proxy implementation, SCONE [9] is used. SCONE is based on the WBI Development Kit from IBM [10].

In the benchmark test the proxy and the browser are located on the same network and are connected to the Internet through a 100 Mbit/s Fast Ethernet connection. The proxy is running on a AMD Athlon 2000+ machine with 512 MB ram, which represents a modest configuration for modern PCs.

Load times have been measured with Ethereal [11] and the load times can be found in Table 1. Two kind of benchmark test have been made. One for measuring the load times of a webpage and one for measuring the time it takes to initialize the proxy. The proxy takes a long time to initialize because all cached certificates are verified to ensure that no certificates have expired or been revoked, i.e., all cached certificates have to be downloaded and verified. The initialization of the proxy is normally only done when the proxy is rebooted, so it is not considered a problem that these load times are quite large. Load times for initialization can be seen in Table 2.

| Test no. | Proxy | No. of Ratings | Time for a page to load |
|---|---|---|---|
| 1 | No | 0 | 2.74 s |
| 2 | Yes | 0 | 7.16 s |
| 3 | Yes | 3 | 6.2 s |
| 4 | Yes | 12 | 6.6 s |
| 5 | Yes | 100 | 8.5 s |

Tab. 1. Load times of webpages.

| Test no. | Proxy | No. of certificates | Initialization of proxy |
|---|---|---|---|
| 1 | No | 0 | 2.74 s |
| 2 | Yes | 0 | 7.16 s |
| 3 | Yes | 3 | 9.9 s |
| 4 | Yes | 12 | 21.54 s |
| 5 | Yes | 100 | 96.27 s |

Tab. 2. Proxy initialization times.

The load times for the webpages are quite good. As described by Mc Nee et al. [17] an entity must have around 80 ratings in order to give a confident result. If we assume that a general article in the Wikipedia have around 100 ratings. This

[9] http://www.scone.de/

[10] http://www.alphaworks.ibm.com/tech/wbidk/

[11] http://www.ethereal.com

gives us an extra load time of 1.5 second, which is quite acceptable.

## 6 Related work

In the following, we examine related work that aims to determine or improve the quality of information in the wikipedia or introduce a reputation system to an existing application.

Dondio et al. [12] proposes a system to determine the quality of articles in the Wikipedia. This system looks at the history of an article in the Wikipedia and evaluates the article based on its revision history. Several properties in the history are examined, such as whether the article is written by an expert, shows clear leadership in the evolution, is constantly reviewed by authors, is stable and well referenced. These properties are used to calculate a trust value that should reflect the quality of the article. We believe that an evaluation of the content based solely on historical attributes are insufficient to evaluate the quality of the content. Such an automated system leaves no room for ”soft” issues, such as whether the article is well written, is unbiased and presents up-to-date information. These soft values can only be detected by human readers, so we believe that ratings based on feedback will be the better choice to determine trust.

In a similar approach, Adler et al. [10,9] proposes a mechanism to calculate the reputation of authors based on the longevity of their contributions. An author who contributes text that remains in the system with few modifications will get a high reputation, while an author who contributes text that is highly modified, or maybe even rolled back, will get a low reputation value. Only the revision history of the Wikipedia articles that an author has contributed to is required to calculate the reputation of an author, so there is no feed-back from the readers and no need for a central repository. The reputation value of all the authors who contributes to a Wikipedia article may then be used to calculate a trust value for that particular article. This work implicitly addresses some of the soft issues identified above, because text that remains unmodified in the system is likely to be considered accurate and unbiased. We do, however, believe that it does not addresses all problems relating to the style of writing as well as problems with articles that are very unevenly edited, e.g., where some paragraphs are written at a very general level while others provide a lot of painstaking details. Such text is clearly accurate and unbiased, but may not be considered well written. Feedback from human readers on the other hand, is able to address all the soft issues identified above.

IWTrust [23] is used to introduce trust into a question answering environment. The idea is that IWTrust tries to create proofs that an answer to a given question is correct. The IWTrust introduces a TrustNet, which is a network of trusted users. Every user that is connected to this network has a trust value - a degree of how much this user is trusted. We adopt the idea of having a network of trusted users, where each member has a trust value, into the Wikipedia Recommender System. With this trust value users find other users that are similar to themselves.

Although some simple tools based on reputation, such as the WOT extension for the Mozilla Firefox browser [8], are starting to appear, they typically rely on a central database for all reputation information, so all user feedback is collated into the same database and the server that hosts this database will also calculate the reputation values. We believe that there are obvious problems with relying on a single centralised data base of feedback for a reputation system, which should provide useful information across national, political, social, religious and cultural boundaries. Storing recommendations in one single database gives the advantage that the recommendations are always available and tamper proof, but it also gives the disadvantage that the centralised calculation of reputation values favours the majority of the users.

## 7 Conclusion

In this paper we addressed the problems of incomplete, incorrect or biased articles in the Wikipedia. Simple attribute based article verification schemes are not considered adequate, so we proposed a mechanism based on feedback from the users of the Wikipedia. This requires reengineering of the Wikipedia to include a reputation system that will manage the feedback.

We examined the general problem of reengineering for reputation and identified four major areas that have to be addressed by the reengineering process: construction of a feedback repository, reputation calculation, identity management and development of an appropriate interception mechanism.

We presented the design and implementation of a recommendation system that can be used to assess articles in the Wikipedia. The recommender system provides the user with a recommendation about whether a given Wikipedia article should be trusted or whether she should seek information elsewhere.

A prototype of the recommendation system is build on top of the existing Wikipedia system without requiring modifications to the WikiMedia software. Existing articles can be rated by active users, which may help other users to determine the credibility of the rated articles. Users store information about the ratings of recommenders whom they have seen before and use this information to calculate an individual and subjective trust value for that recommender. This information is used to calculate the weight of feedback from that recommender in future trust calculations.

We have demonstrated the ability to seamlessly integrate a recommender system with an existing Wiki implementation, where recommendations are stored in invisible comments on the Wiki pages. This type of recommendation system allows clients to implement a subjective ranking algorithm, which prevents a (possibly artificially created) majority of users from dominating the recommendations made by the system. Moreover, the recommender system is designed so that it will be transparent to users who do not wish to use this recommender system.

Preliminary experiments with the implemented prototype indicate that the overhead of the proposed architecture is negligible (both with respect to computational overhead and storage space) and the system is unlikely to affect the users experience while reading articles in the Wikipedia.

# References

[1] *enwiki dump progress on 20070402*, http://download.wikimedia.org/enwiki/20070402/ (2007), *accessed 29 May 2007.*

[2] *Amazon web-site*, http://www.amazon.com (2008), *accessed 2 June 2008.*

[3] *eBlogger: On vandalism*, http://eblogger.blogsome.com/2005/10/24/on-vandalism/ (2008), *accessed 2 June 2008.*

[4] *European consumer center denmark: Howard the shopping assistant*, http://www.forbrugereuropa.dk/english/facts-and-advice/ecommerce/assistant/ (2008), *accessed 2 June 2008.*

[5] *Internet movie database web-site*, http://www.imdb.com (2008), *accessed 2 June 2008.*

[6] *Wikipedia*, http://en.wikipedia.org/wiki/Wikipedia (2008), *accessed 2 June 2008.*

[7] *Wikipedia:size of wikipedia*, http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia (2008), *accessed 2 June 2008.*

[8] *Wot: Against intuition inc*, http://www.mywot.com (2008), *accessed 18 May 2008.*

[9] Adler, B. T., J. Benterou, K. Chatterjee, L. de Alfaro, I. Pye and V. Raman, *Assigning trust to wikipedia content*, Technical Report UCSC-CRL-07-09, School of Engineering, University of California, Santa Cruz, U.S.A., http://works.bepress.com/luca_de_alfaro/5 (2007).

[10] Adler, B. T. and L. de Alfaro, *A content-driven reputation system for the wikipedia*, in: *Proceedings of the 16th International World Wide Web Conference*, Banff, Alberta, Canada, 2007, pp. 261–270.

[11] Brin, S., J. Davis and H. García-Molina, *Copy detection mechanisms for digital documents*, in: *Proceedings of the ACM SIGMOD Annual Conference*, San José, U.S.A., 1995, pp. 398–409.

[12] Dondio, P., S. Barrett, S. Weber and J.-M. Seigneur, *Extracting trust from domain analysis: A case study on the wikipedia project.*, in: *ATC*, 2006, pp. 362–373.

[13] Jones, M. B., *Interposition agents: Transparently interposing user code at the system interface*, in: *Secure Internet Programming*, Lecture Notes in Computer science **1603**, Springer Verlag, 1999 pp. 339–369.

[14] Jøsang, A., R. Ismail and C. Boyd, *A survey of trust and reputation systems for online service provision*, Decision Support Systems **43** (2007), pp. 68–644.

[15] Kiczales, G., J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier and J. Irwin, *Aspect-oriented programming*, in: *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, Jyväskylä, Finland, 1997, pp. 220–242.

[16] Korsgaard, T. R., "Improving Trust in the Wikipedia," Master's thesis, Departmen of Informatics and Mathematical Modelling, Technical University of Denmark (2007).

[17] McNee, S. M., S. K. Lam, C. Guetzlaff, J. A. Konstan and J. Riedl, *Confidence displays and training in recommender systems* (2003).

[18] Orlowski, A., *Avoid wikipedia, warns wikipedia chief, it can seriously damage your grades*, The Register (2006).

[19] Podcasting News, *Wikipedia Caught in Podfather Turf War*, http://www.podcastingnews.com/archives/2005/12/wikipedia_caugh_1.html (2007), *accessed 23 May 2007.*

[20] Seigneur, J.-M., S. Farrell, C. Jensen, E. Gray and Y. Chen, *End-to-end trust in pervasive computing starts with recognition*, in: *Proceedings of the First International Conference on Security in Pervasive Computing*, Boppard, Germany, 2003.

[21] Utter, D. A., *Wikipedia bans congress*, http://www.webpronews.com/topnews/2006/01/30/wikipedia-bans-congress (2007), *accessed 28 May 2007.*

[22] *What is wiki*, http://www.wiki.org/wiki.cgi?WhatIsWiki (2008), *accessed 2 June 2008.*

[23] Zaihrayeu, I., P. P. da Silva and D. L. McGuinness, *Iwtrust: Improving user trust in answers from the web*, in: *Proceedings of the Third International Conference on Trust Management*, Rocquencourt, France, 2005, pp. 384–392.