CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2015; **27**:2778–2795
Published online 7 October 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3420

SPECIAL ISSUE PAPER

P2Pedia: a peer-to-peer wiki for decentralized collaboration

Alan Davoust[1], Alexander Craig[1], Babak Esfandiari[1,*,†] and Vincent Kazmierski[2]

[1]*Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada*
[2]*Department of Law, Carleton University, Ottawa, Ontario, Canada*

SUMMARY

Existing Wiki systems such as Wikipedia depend on a centralized authority and cannot easily accommodate multiple points of view. We present P2Pedia, a social peer-to-peer wiki system, where users have their own local repository and can collaborate by creating, discovering, editing, and sharing pages with their peers but without synchronizing them. Multiple versions of each page can thus co-exist on each repository and across the network, which allows for multiple points of view.

Browsing or searching the wiki thus yields multiple page versions; to help the user's page selection process, the system annotates search results with trust indicators based on the distribution of each version in the peer repositories and the topology of the social network.

We describe an experimental study where the system was deployed for academic writing exercises, and we analyze the results to validate different aspects of this collaboration principle. Copyright © 2014 John Wiley & Sons, Ltd.

Received 20 February 2014; Revised 2 July 2014; Accepted 15 September 2014

KEY WORDS: wiki; decentralization; peer-to-peer; collaboration

1. INTRODUCTION

Wiki Wiki Web was created by Ward Cunningham as a simple content management system that promotes collaborative content creation on the Web [1].

Collaboration in a traditional wiki follows the principles of many groupware, defining a virtual space holding a shared collection of artifacts that all the contributors can edit. Typically, such processes involve limited teams, and Wikipedia is a successful experiment in breaking from this limitation, and opening up participation to anyone.

However, critics of Wikipedia argue that this open model allows for vandalism and misinformation, leading to untrustworthy content. Yet, despite a few high-profile cases [2], vandalism is usually quickly fixed [3].

A more difficult issue is 'edit wars' [4, 5]: articles are sometimes repeatedly edited and counter-edited by disagreeing contributors, and an arbitration process is necessary, often leading to articles being write-protected by senior editors. Issues at stake in these edit wars are often somewhat trivial, such as British versus American spelling.

Nonetheless, this ongoing problem shows the limits of a consensus-based approach; ultimately, there will be conflicts, requiring elaborate resolution processes and leaving some participants frustrated with their outcome.

In this paper, we describe P2Pedia, a peer-to-peer (P2P) wiki system aiming to accommodate multiple viewpoints, through truly decentralized control. In our approach, the system does not define

*Correspondence to: Babak Esfandiari, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario K1S 5B6, Canada.
†E-mail: babak@sce.carleton.ca.

a single shared repository, but is rather made up of a federation of individual repositories, owned and controlled by the different contributors, and organized in a social network. The artifacts (i.e., the wiki page versions) are shared following the principles of P2P file-sharing: users can search the global collection, and download pages shared by other users, which they can then edit in their own repository.

A number of different versions of any article may exist in the network, and user queries return all the relevant versions, from which users must then choose their preferred version to download and view. As users cannot directly modify content within each other's repositories, an edit war in P2Pedia would simply lead to (at least) two competing versions of the disputed page, which could both coexist in the wiki.

This approach is similar to the collaboration principle supported by distributed version control systems (DVCS) such as Git [6], with the additional feature that the global collection made up by the different repositories is viewed as a single wiki, which can be browsed and searched.

This approach raises a number of issues. In particular, by removing the editorial process, be it Wikipedia's consensus-based process or traditional expert editing, we transfer the burden of evaluating content to the reader, who must choose between the different versions of a page, a task that may seem daunting, particularly as the number of versions increases.

Our intuition is that the collective behavior of many participants can produce a 'wisdom of the crowd' effect [7], and lead to emergent properties in the distribution of wiki pages and page versions. For example, we expect some contributions to be agreed upon by a large proportion of the users, and others to be "popular" only within specific social circles: such aspects can be captured by specific *trust indicators*, exposed by the system as annotations to search results. The main indicator that can be applied to documents is *popularity*, while indicators applicable to users (peers) include *popularity* and *distance* in the social network, as well as the *similarity* of peer repositories.

We investigate the relevance of our proposed indicators based on a pilot deployment of P2Pedia for writing tutorials, as part of an early year undergraduate course at Carleton University. The participants were presented a writing problem, to which they produced many different solutions. We analyzed these solutions, broken down to a fine granularity, and demonstrate that the *popularity* of the contributions is significantly correlated with a measure of *quality* relevant to the setting. We also show that the *similarity* of users is meaningful, in the sense that the similarity of users measured on part of the problem was strongly correlated with the users' similarity on another part of the problem. In other words, past similarity is predictive of future similarity, which is a condition required for techniques such as collaborative filtering to be applicable.

The concept of P2Pedia was initially presented in [8], and the user study discussed in the second part of this paper was previously described in [9]. The system presentation has been completely rewritten, with additional insights from the comparison with W. Cunningham's related 'Smallest Federated Wiki' and a discussion of technical limitations; the results of the user study have been updated with additional data from a second user survey.

The rest of this paper is organized as follows. After a brief survey of related work in Section 2, we describe the functionality of P2Pedia in Section 3. Section 4 presents our user study and the collected data, then Section 5 details our analysis and results. Further insights from the users' feedback are discussed in Section 6. We discuss some limitations and future work in Section 7 and draw some conclusions in Section 8.

2. RELATED WORK

2.1. Physically distributed wikis

Traditional Wiki systems are simple content management systems, with a Web interface that supports easy editing from remote clients (Web browsers). Large scale wikis such as Wikipedia require complex and highly distributed infrastructures [10], but these distributed infrastructures are managed by a single organization.

In recent years, there have been a number of proposals for distributed wikis where the components of the infrastructure would be under decentralized control, that is, networked components controlled

DOI: 10.1002/cpe

by different organizations. Typically, the architecture proposed for such systems is a distributed hash table, that is, a structured P2P network used for distributed storage.

Piki [11] and Uniwiki [12] distribute the wiki contents over all the participating peers, allowing scalability, cost sharing, and use limited replication to ensure fault tolerance. Wooki [13], also based on a P2P network, fully replicates the wiki on each node, privileging maximal fault-tolerance, at the cost of low scalability. Uniwiki and Wooki use two variations of the same algorithm to handle version conflicts, propagating and automatically merging updates to all replicas in order to ensure their eventual consistency.

However, such distributed wikis can be described as *logically centralized*, in the sense that they aim to reproduce the functionality, including the content management model, of a traditional centralized wiki.

An exception is Distriwiki [14], deployed in a P2P network using the JXTA protocol. In this proposal, it appears that users are expected to search the network for the latest version of a page in order to edit it, and concurrent editing is left to the user to resolve. As such, this P2P wiki engine could lend itself to alternative collaboration models, such as the one discussed in this paper. However, these possibilities do not seem to have been explored.

2.2. Decentralized content management

With the advent of DVCS such as Git, Mercurial, or Bazaar, it was a natural step to adapt such version control systems to serve as a back-end for wikis. While such systems have not received much academic attention, several implementations are available: among others, we can cite git-wiki [15], Olelo [16], derived from the previous, and gollum [17], the wiki engine used in the Github online code repository.

Decentralized approaches to version control assume multiple repositories and define operations to exchange content between these repositories, that is, cloning, forking, and propagating changes. In other words, they define a *logically decentralized* form of content management.

However, in the DVCS-based wikis listed in the preceding text, the decentralized aspect of the underlying content management solution is not translated to new functionalities within the wiki interface. Any DVCS-specific operation on the wiki content, such as a clone or fork operation, must be realized through a separate mechanism, and is presumably not easily available to ordinary end-users.

One semantic wiki, DSMW [18], includes functionalities relevant to decentralized control management. This semantic wiki implements a multi-synchronous collaboration model [19], closely related to the DVCS idea. Participating peers host a full wiki, and subscribe to feeds of updates from other peers. The feeds are then automatically merged with an algorithm that ensures Causality, Convergence and Intention (CCI) consistency.

However, the consistency guarantees restrict the possibilities of divergence. The multi-synchronous workflow is intended to lead to eventual consistency, and although a participant can choose not to integrate a set of changes from another participant, this will prevent it from integrating any changes made later on, virtually disconnecting the two versions. Our approach is more flexible, encouraging divergence while allowing exchange of content at any time between any participants, regardless of their history.

2.3. Federated wikis

More closely related to our work is the Smallest Federated Wiki (SFW) project,[‡] spearheaded by Ward Cunningham, the creator of the original wiki.

In Cunningham's view [20], wikis should be interoperable and users should be able to copy content from one to another, in the same spirit as one can 'fork' a software project using a DVCS. In this view, as in our own, consistency between different versions of a page is neither a goal nor even desirable.

‡https://github.com/WardCunningham/Smallest-Federated-Wiki. Note that we describe this project based on its implementation and available documentation, as it has, so far, not been discussed in any academic publication.

In SFW, each participant has a wiki server. Connections with neighbouring nodes are maintained by links in the history of the pages: when a page is forked from one node to another, the fork is recorded in the history of the version in the receiving node, with an indication of where the page was forked from.

The main difference with previous distributed wiki approaches is that the user of SFW can explicitly navigate the different nodes of the federated wiki. By following wikilinks, the user navigates between the pages of her own wiki, and when the link points to a page that she does not have, the system automatically selects a version from a neighbour, and directs the user to view this version. However, as with ordinary Web pages, this version is not saved locally (i.e., on the user's 'home' node); if the user wishes to save it, she must explicitly 'fork' it using a button in the page footer.

The Uniform Resource Identifiers (URIs) of pages include two parts: one part to identify the site that they are being viewed from (the viewer's 'home' node), and a second part that indicates the provenance of the page being viewed, and the page title.

Federated wikis support keyword searches, and pages returned in a search may come from the local node, but also from some of the neighbouring nodes.

The main differences between our approach and Cunningham's SFW are the following:

- While SFW is designed for a set of permanent servers, nodes in P2Pedia are P2P applications that may join or leave the network at any time.
- The network neighbourhood in P2Pedia is managed manually by the user, rather than by the system.
- Page versions in SFW are identified by their location, whereas in P2Pedia, replicated pages share the same identifier.
- In SFW, when following a wikilink, the choice of pages is not up to the user (as is the case in P2Pedia); instead, it is built into the system, much like in a traditional wiki. This helps smooth navigation, but limits the opportunities for comparing multiple versions.
- Searches in SFW are limited to the local node and its direct neighbours at query time, whereas in P2Pedia, they are propagated through the network.
- Any page viewed in P2Pedia is saved locally by default.
- In SFW, each node may only host one page (one version) with a given title, whereas in P2Pedia, a user may keep any collection of versions of a page.

2.4. *Online community question and answer sites*

Wikis in which multiple competing versions are allowed (e.g., P2Pedia or SFW) present a similarity with community Question and Answer sites such as StackOverïňĆow or Yahoo! Answers.

On these sites, users can post questions, and other users can provide answers, and vote on how good those answers are. This allows different perspectives and opinions for the same question, while comments and votes help identify the most valuable answers. On some sites (notably StackOverflow) , users can garner 'reputation points', when their answers receive positive votes. Reputation may simply serve as a gratification system to motivate the users, or it can provide special privileges, such as extra editorial rights (e.g., the right to delete other answers...).

However, answers from reputable users are not identified or promoted in any particular way, and their votes do not carry more weight. It is interesting to note that the reputation system is centrally managed (the rules are defined by the host company) and the points are an *absolute* (or *global*) type of trust metric: that is, it provides a single trust value for each user, rather than allowing *relative* (or *local*) user-to-user trust [21]. This suggests that the quality of contributions is assumed to be an objective metric, as opposed to *opinions*: on a site where users review products (e.g., epinions.com), users can decide which other users they trust. This establishes a network of relative trust.

P2Pedia allows for different perspectives in a wiki-like setting, and our decentralized control model makes it more natural to let the users establish relative trust based on implicit ratings, and express this trust as social connections.

For question answering, it is interesting to think what a 'traditional wiki'-like approach would produce: there would be only one answer to each question, and the users would cooperatively edit this single answer.

DOI: 10.1002/cpe

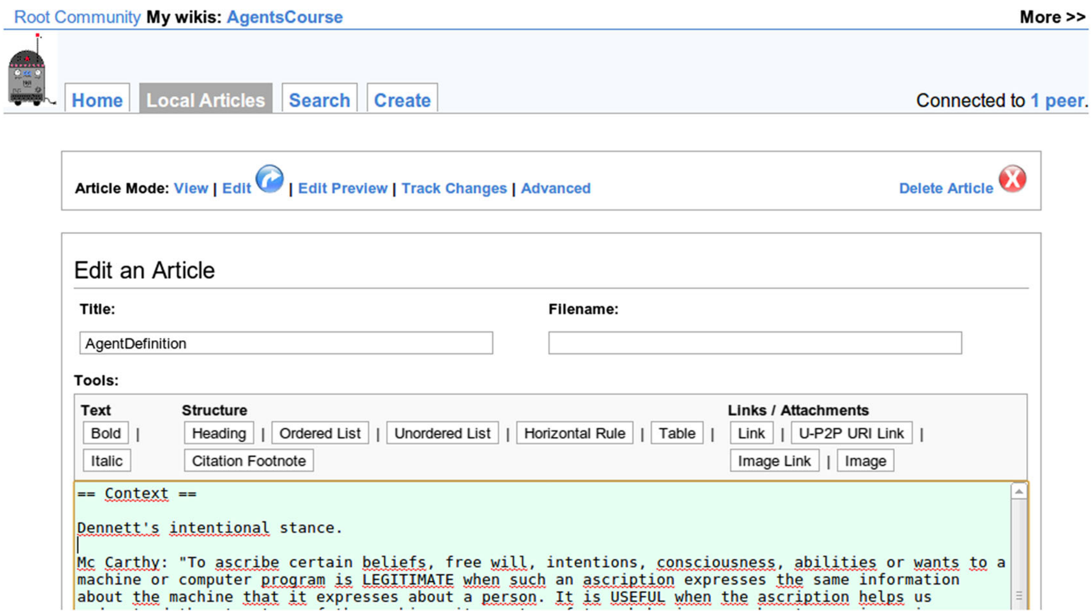


Figure 1. A screenshot of the P2Pedia interface. The user is editing a page.

3. P2PEDIA, A DECENTRALIZED WIKI

3.1. Overview

P2Pedia is accessed through a Web browser, and its interface is similar to that of most existing Web-based Wikis. Users may browse pages, perform keyword searches (on the page title and/or on the textual content), and edit any page, as in a traditional wiki, by clicking on a tab at the top of the page, and editing the Wikitext (text with formatting markup) in a text box, using the Creole Wiki Syntax [22].

A screenshot of the P2Pedia interface, where a document is being edited, is shown in Figure 1.

The main difference with traditional wikis is that P2Pedia does not define a single shared space containing all the pages; it is rather a federation of personal wikis that are made to interoperate. The central principle of this interoperability is that users can *read* (browse and search) pages from any of the wikis, seen as a single global collection, but any pages they *write* will be stored in their own personal wiki. 'Editing' a page means creating a new version of this page, which is stored locally and co-exists with the old version.

In practice, P2Pedia is implemented as a P2P application, and each user's local installation has a repository where the user's personal collection of wiki pages is stored. Unlike most P2P networking approaches but as in online social networks, the connections in P2Pedia are under control of the user, who can thus establish them based on personal preference.

3.2. Wiki data model and operations

In this section, we specify the functionality of P2Pedia in a more formal way.

3.2.1. Notations. We introduce the following notations.

We consider a set of users $U = \{u_i\}$ and the graph $G_t = \langle U, E_t \rangle$ of their social acquaintances at time t.

Each user u_i has a repository R_i, which contains a set W of *wiki pages*. Each wiki page $w_{id} \in W$ is identified by a globally unique identifier id (generated using a hash of the page content), and has the following properties:

- a title;

DOI: 10.1002/cpe

- content (wikitext);
- attachments (embedded media objects);
- a history: the history is the ordered list of all its parent and ancestor versions, identified by their unique identifiers.[§]

It is worth noting that these wiki pages w_{id} are immutable (with respect to their identifier). In the terminology commonly used for traditional wikis, our concept of a wiki page corresponds to *page revisions*. In the traditional sense, a page is a variable pointing to the 'current' revision of the page.

In P2Pedia, the wiki pages are used like single-assignment variables. For a given title, there can be any number of wiki pages with this title, with no notion of 'current' wiki page, and no particular ordering between them. What may distinguish these wiki pages is their distribution among the users, as we will discuss further in the paper.

3.2.2. Write operations. We first consider the 'write operations', whereby the wiki is modified.

As in traditional wikis, pages can be created (from a blank editing form), deleted, or edited. However, as outlined in the preceding text, these operations will have specific semantics in P2Pedia: wiki pages are immutable, and 'editing' consists in creating a new wiki page, 'child' of the edited one. Deleting a page only removes a local copy of that page, rather than all the available copies.

In addition, P2Pedia defines an additional operation, *download*, that means to copy a page from a remote repository to the local repository.[¶] These operations are specified by the pseudocode in Algorithm 1.

Algorithm 1 Wiki operations available to a user u_i. (Modifying operations)

function create(*title, content, attachments*):
  $id \leftarrow$ generateId(*title, content, attachments*)
  $w_{id} \leftarrow \langle$*title, content, attachments*, []$\rangle$
  $R_i \leftarrow R_i \cup \{w_{id}\}$

function delete(id):
  $R_i \leftarrow R_i \setminus \{w_{id}\}$

function edit(id, op): {op is a modification to the page}
  $\langle$*title, content, attachments, history*$\rangle \leftarrow w_{id}$
  content2 $\leftarrow$ *op*(*contents*)
  history2 $\leftarrow$ [*id, history*] {prepend id to *history*}
  $id2 \leftarrow$ generateId(*title, content2, attachments, history2*)
  $w_{id2} \leftarrow \langle$*title, content2, attachments, history2*$\rangle$
  $R_i \leftarrow R_i \cup \{w_{id2}\}$

function download(id):
Require: $\exists u_j \in U, w_{id} \in R_j$
  $R_i \leftarrow w_{id}$

3.2.3. Basic read operations. The way wiki pages are viewed, browsed, and searched for is also different in P2Pedia. Again, as pages are not uniquely defined by their title, retrieving a specific wiki

[§]An alternative implementation could include only the first parent, or a bounded number of them, reducing the size of pages but introducing a risk of losing part of the history information (the difference is only on the 'filiation' information, as the contents of the previous versions are not maintained with any guarantees, in either case).

[¶]For our current implementation of P2Pedia, 'local' and 'remote' can be taken in their literal sense, but the concept would easily extend to a different physical deployment, where 'local' then refers to the repository of the user executing the operation, and 'remote' refers to another user's repository.

DOI: 10.1002/cpe

page is performed using its id. In addition, the location of a particular page is not known *a priori*; that is, there may be any number of copies of the page, in arbitrary user repositories.

The basic read operations in P2Pedia are the *lookup* of a page based on its unique id, and the *search*, that returns all the available page versions matching a particular logical condition. These operations are given in Algorithm 4.

The lookup first checks the local repository. If the page is not in the local repository, a *search* is performed to find any copy available in other repositories. The search that may be performed within the lookup operation uses an exact match on the page id as a search condition.

However, in general, a search can take any boolean condition. The search is not necessarily applied to the entire wiki: P2Pedia uses the social network to disseminate the query, following a protocol that does not guarantee that the full repository will be searched. The social network operations and the broadcast operation used in P2Pedia are given in Algorithm 2.

Algorithm 2 Basic read operations in P2Pedia.

```
function lookup(ident):
if w_ident ∈ R_i then
   return w_ident
else
   return search(id = ident)
end if

function search(condition):
for all j ∈ broadcast(i) do
    {The broadcast function returns a subset of nodes in G_t}
   for all w_id ∈ R_j do
      if matches(w_id, condition) then
            Hits ← Hits ∪ w_id
      end if
   end for
end for
return Hits
```

3.2.4. Wikilinks and page history search. In traditional wikis, the most recent version of a page is considered the 'authoritative' version of the page. Alternative versions are maintained as part of the history, but are not considered valid alternatives, and can no longer be edited.

In P2Pedia, users may choose to keep different versions of a page in their repository, and can edit any of these versions, regardless of whether it is the most recent one. When a page has multiple, parallel edits, in version control terminology, this creates a 'fork' in the page history.

The result is that the history of a page may branch out into a tree, as illustrated in Figure 2(b), whereas in traditional wikis, it is linear (Figure 2(a)).

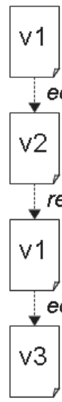
The semantics of wikilinks in traditional wikis are the following: a wikilink is defined by its *label*, and the label is the title of another page. Clicking on the link returns the most recent revision of the page bearing this title, or else a blank form to create the page.

In P2Pedia, the wikilink conceptually points to the entire *set* of wiki pages sharing this title. Unlike the traditional setting, these wiki pages are not an ordered sequence of revisions, but possibly very different versions maintained by different users, and expressing opposite points of view. The user can then retrieve all or any of these pages.

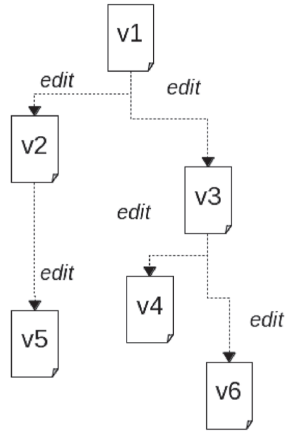
Therefore, following a wikilink in P2Pedia returns *all* the available page versions with the requested title. This behavior is implemented as a *search* on the page title, as shown in Algorithm 3.

The version tree of a P2Pedia page can also be explored, as one might want to view the history of a page in Wikipedia. P2Pedia supports a graph query language based on graph traversal: a path pattern can be defined, starting from a given page version, in order to retrieve related versions. Using the

DOI: 10.1002/cpe



(a) Versioning in a centralized Wiki: v3 is the "current" version.



(b) Versioning in P2Pedia: v4, v5 and v6 are "leaf" versions in the versioning tree.

Figure 2. The different versioning processes available in a centralized and decentralized collaboration model produce different versioning hierarchy structures. Each figure shows an example hierarchy of versions for a single page. (a) Versioning in a centralized Wiki: v3 is the 'current' version; (b) versioning in P2Pedia: v4, v5, and v6 are 'leaf' versions in the versioning tree.

'child/parent' metaphor to refer to consecutive versions of a page, relevant queries include finding all the ancestor versions of a page, the descendant versions, or the sibling versions (child versions of the same parent).

As the version graph of a page can be distributed across multiple repositories, graph traversals are implemented using recursive *search* operations, using the identifiers in the *history* attribute of pages. As an example, the 'sibling' query is given in Algorithm 3.

Algorithm 3 Browsing and History Search in P2Pedia

function wikilink(*label*):
return search(*title* = *label*)

function sibling(*id*):
  page ← *lookup*(*id*)
  parentid ← *page.history*[0] {get parent version}
return search(*history*[0] = *parentid*) {find all children of the parent}

3.2.5. Social network operations. The social network is maintained by each peer, who can establish or drop connections to others. These operations are specified in Algorithm 4. The social network connections are used to broadcast search messages, which are propagated along its edges, up to a certain horizon. However, the choice of search protocol – that is, given a search initiator node, what subset of peer repositories should be searched – does not affect our general approach. As we discuss in Section 2, the similar project *Smallest Federated Wiki* adopts a different search approach, only searching a node's direct neighbours.

3.3. Trust indicators

Searches and wikilink navigation may return many different versions of the same page. In order to help the user make sense of these results and choose a version to download, P2Pedia offers the user *trust indicators*.

These indicators, which apply to peers or to documents, are not an absolute assessment relying on a central authority; rather, they are relative and depend on local criteria such as the querying peer's

DOI: 10.1002/cpe

Algorithm 4 Social Network Functions

```
function connect(i, j):
    E_{t+1} ← E_t ∪ (i, j)

function disconnect(i, j):
    E_{t+1} ← E_t \ (i, j)

function broadcast(i): {a crude limited breadth-first search}
    Neighbours ← {i}
for i = 1 to TTL do
    {TTL is a constant parameter}
    for all j ∈ Neighbours do
        for all (j, k) ∈ E_t do
            Neighbours ← Neighbours ∪ {k}
        end for
    end for
end for
return Neighbours
```

network neighbourhood, or on the local repository contents. In addition, they are simply indicators that users can combine in order to make a personal assessment of peers and documents.

P2Pedia offers the following trust indicators, which are provided along with the results of all searches.

3.3.1. Document popularity. The main indicator to assess the value of a page version is its *popularity*. The underlying assumption is that if many users keep a particular version in their repository, it is an indication of quality.

Formally, document popularity is (|| denotes set cardinality) defined as follows:

$$trust_{popularity}(w_{id}) = |\{i, w_{id} \in R_i\}| \tag{1}$$

3.3.2. Social networking trust indicators. By giving the users control of their connections (instead of using a structured network, for example), users have the opportunity to connect to trusted acquaintances. The social network G_t can be used as a basis to assess the trustworthiness of other peers. For a considered peer u_i (issuing a query), the following indicators will be attached to a responding peer u_j:

- Peer popularity: The number of users that have connected to u_j, that is, the indegree of node u_j in G_t.
- Network distance: The distance (length of the shortest path) from u_i to u_j in the social graph G_t. Assuming that trust has some level of transitivity, the network distance can be a good trust indicator.

3.3.3. Similarity. The similarity between two users is measured by the similarity of their repositories. The rationale of this trust indicator is the same principle used in collaborative filtering (or recommender systems): the fundamental assumption [23] is that if two users tend to rate items similarly, they share similar tastes, and hence will rate other items similarly.

Here, we do not use explicit ‘ratings’ of pages, but consider that if a user keeps a page version in her repository, it expresses an implicit positive rating of this page. The ‘rating similarity’ of two users u_i and u_j is then measured by the Jaccard coefficient of the sets of page versions in their repositories:

$$trust_{sim}(u_i, u_j) = \frac{|\{R_i \cap R_j\}|}{|\{R_i \cup R_j\}|} \tag{2}$$

DOI: 10.1002/cpe

We can infer from these indicators that for a user to be deemed trustworthy, and to fully be able to use the indicators for her own benefit, she must actively maintain her content (by keeping documents that it 'endorses' and deleting the others) and her neighbourhood (by connecting to similar peers and disconnecting from those that she does not trust anymore).

4. ACADEMIC WRITING TUTORIALS WITH P2PEDIA

We now present an experimental validation of P2Pedia as a collaboration tool.

In a classroom setting, a common practice is to have the students first work individually or in small groups on a problem, then open up the discussion to the full classroom, in order to compare and discuss the different solutions obtained by the different students or groups.

In an e-learning setting, using a traditional groupware system such as a wiki makes it difficult to expose students to multiple solutions, and may suggest that there is always a single, best solution.

In the following sections, we report our experience using P2Pedia, as part of an undergraduate course at Carleton University, on research methods for legal studies. The tutorials were meant to expose common flaws in academic writing. Students were given a poorly written text, and asked to improve its style. This is a good example of a problem, which does not have a single solution, but several possible ones.

The participants worked first independently, then in small groups where they collaborated to put together their individual contributions. We collected and analyzed the documents written by the students and obtained some additional feedback through a Web survey.

4.1. Research questions

Our analysis of the documents aimed primarily to validate the trust indicators of P2Pedia and assess whether this decentralized approach to collaborative writing could benefit their learning experience.

More precisely, it addresses the following research questions:

How can the students identify valuable contributions from their peers?

In order to answer this question, we analyze the different users' contributions and validate the underlying assumptions of P2Pedia's trust indicators. Our findings show that document popularity and user similarity can help students identify valuable contributions. As the network topology was imposed, we did not validate our social networking trust indicators.

Do the students benefit from having their own solutions, versus agreeing on a common solution?

In order to answer the second question, we used some elements of our data analysis, and feedback collected from the survey. Our analysis of the documents shows that the best contributions (as measured by 'popularity') could not all be combined into a single ideal solution. Creating a single solution would have therefore required leaving out many valuable contributions. Feedback from both the students and the course instructor lead us to believe that this would have been detrimental to the students' learning experience.

4.2. Tutorial setup and workflow

The tutorials took place as part of an early year undergraduate course in the Department of Law at Carleton University, in the Fall and Winter semesters of 2011–2012. In each semester, there were 12 tutorial groups, of approximately 20 students each, and each group had a separate tutorial session. The tutorial sessions were held in classrooms with one computer per student, and one for the instructor, and lasted 80 min.

The tutorial workflow was as follows.

- Initially, the provided text was published on the instructor's node. Students were grouped in small groups of three to five students, so that there were four subgroups in each tutorial groups. The connections between the nodes were set up so that the students could access the work of their neighbours in the small groups.

DOI: 10.1002/cpe

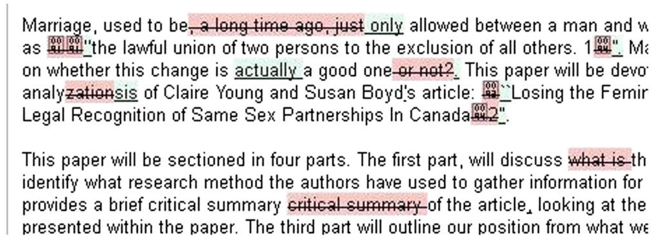


Figure 3. P2Pedia screenshot: 'Diff' highlighting of the changes between two versions.

- Each student downloaded the initial document, then worked individually to improve its style.
- After approximately 20 min, the students were asked to search the network and download the versions written by their peers from their subgroups. The students were given 10 min to read the edits made by their peers, which could be highlighted by a 'diff' functionality, as illustrated by the screenshot in Figure 3.
- The students then got together by subgroups, and collaboratively created a 'master' version merging their different edits. For this step, they sat together and worked on a single computer. This second phase of editing lasted approximately 15 min.
- The instructor then led a group discussion with the whole classroom, of the different solutions proposed by the different subgroups.

After the tutorials, the different versions of the documents written by the students were collected, and the students were asked to fill out an online survey.[‖]

We note that the documents analyzed here come from the first semester only (12 tutorial groups), but the survey answers were collected from both sessions.

4.3. *Collected documents*

A total of 241 different versions of the document were collected, 208 versions from the individual editing phase, and 35 versions from the group editing phase.

The first sentence of the document to be edited, which we will be used as an example throughout this paper, was the following:

> Marriage, used to be, a long time ago, just allowed between a man and woman, but it is now defined as "the lawful union of two persons to the exclusion of all others."

4.4. *Atomic edits*

The different versions written by the peers were analyzed using a 'diff' tool, which tracks the changes between two versions of a document by listing the minimal additions and suppressions to go from the first version to the second. We applied the diff tool at the granularity level of whole words and tracked the exact location of each atomic edit with respect to the original text. For example, an atomic edit might list the words 'Marriage, used to be', as being removed at character position 0, or the word 'was' being inserted at position 38 (the position refers to the position with respect to the original text; the inserted word may be only at position 10 in the new text).

This way, each document version can be represented as a set of atomic edits. As we will see, the collection of atomic edits has much more interesting statistical properties than the collection of documents: due to the limited editing time, each student only produced one or two versions of the document, and only downloaded those of her immediate neighbours to read them, as per the tutorial workflow; on the other hand, each document version contained dozens of atomic edits, and these edits were often made by many students: some appeared in over 50% of the document versions.

[‖]We note that this research has received clearance by Carleton University's Ethics Review Committee. The students did not receive grades for the work performed during the study, they could opt-out of having their data collected, without the course instructor's knowledge; none did), and the survey was online and anonymous. We believe that it is very unlikely that any participant felt any pressure related to their grade in the course.

DOI: 10.1002/cpe

Table I. Evaluation of the sample documents by the class instructor and by automatic scoring functions. The letters A, B, . . . J identify the documents of the sample.

Rank	Expert	$S1$	$S2$
1	A	A	A
2	B, C	B	B
3		D	F
4	D, E	F	D
5		G	E
6	F, G, H, I	H	C
7		E	G
8		C	H
9		I	I
10	J	J	J

Our dataset of 208 versions created in the first phase produced around 5500 different atomic edits. Interestingly, we observe that the frequencies of the different edits appear to follow a distribution very similar to that of words in a natural language, which follow the so-called 'Zipf's law' [24]: the frequency of the n^{th} most frequent word is inversely proportional to n.

5. EXPERIMENTAL RESULTS

5.1. Edits and document quality

We now address the following research question: is the *popularity* of the edits contained in a document an indication of the *quality* of the document?

In the specific context of these writing tutorials, the students' goal was to improve the language of the documents, that is, the grammar, spelling, and general writing style. Therefore, the quality of these particular texts can be evaluated in a somewhat objective way by a human expert.

In order to evaluate the correlation between popularity and quality, we automatically assigned scores to the documents, based on the popularity of their edits, and compared a sample of these scores with the evaluation of a human expert, namely the class instructor.

Our scoring function $S1$ uses the frequencies of the edits: for each edit in a text, we assign a score equal to the popularity of this edit (i.e., the number of times it occurs across the different document versions), and sum the scores of all the edits in each version. We then ranked the documents by this score and chose ten documents corresponding to scores at the 10%, 20%, .. 100% percentile ranks, and submitted these documents to our expert for evaluation.

We also experimented with variations of our scoring function, which mostly produced identical or nearly-identical rankings of the papers. We include here the results of a second scoring function $S2$, which considered only the most popular edits: each edit was assigned 1 point if it was among the top 300 popular edits (meaning it appeared in at least 10% of the documents), and 0 otherwise. The edit scores were then added up for each document, as in the scoring function $S1$.

The results are summarized in Table I, which shows the ranking of the documents according to the expert, and according to our two automatic scoring functions. The expert considered that several documents were of equivalent quality, and instead of assigning absolute scores to each document, he ranked them in five tiers, from the best to the worst.

In order to estimate numerically the correlation between these different evaluations, we evaluate the correlation between the rankings that they produce, using Spearman's rank correlation coefficient.[**]

[**] Spearman's rank correlation coefficient is obtained by replacing each value of each random variable by the rank of that value, then applying Pearson's correlation coefficient to the obtained 'rank' variables.

DOI: 10.1002/cpe

The correlation value of $S1$ with the expert correlation is 0.61, and the correlation of $S2$ with the expert evaluation is 0.77. We estimated the significance of these figures by estimating the correlation of the expert ranking with 1000 randomized rankings. Our true correlation coefficients were found to be within the top 5% of all the values obtained by randomization, which confirms the statistical significance of our figures with at least 95% confidence.

We can thus conclude that the most popular edits are the most likely to make the best quality documents. This suggests that it could be valuable to aggregate the most popular edits to make a synthetic 'ideal' version. We explore this question in the next section.

5.2. Edits and multiple solutions to a problem

One important observation here is that many of the edits are incompatible; that is, they cannot be applied to the same text. A simple example of incompatible edits would be a pair of edits where different words are inserted at the same position.

Using simple compatibility criteria, such as the one stated in the preceding text, we analyzed the compatibility of the most popular edits. Among the top 100 popular edits, we found 33 pairs of incompatible edits. Among the top 300 popular edits, we found 324 pairs of incompatible edits. This is consistent with the characteristics of the problem: as the class instructor noted,

> "[the] goal was not to have students create a 'perfect' edited version that matched some pre-established notion of what the text should look like. Rather, the goal was to challenge students to identify ways to improve the sample text. There was no single 'right' answer to this exercise."

As the most popular edits were incompatible, this implies that it is not possible to make a single 'ideal' document, but that several potential 'best' documents could be made by combining the compatible edits together. An interesting question is as follows: what is the minimal number of documents that can accommodate the top k most popular edits?

This problem can be expressed as a graph coloring problem, and is thus NP-complete but it can be solved for a small value of k. We found that the 100 most popular edits can be accommodated in three different versions. The 300 most popular edits – corresponding to all the edits that occur in at least 10% of the documents – require at least seven different versions. We note that mathematical solutions to the problem would not necessarily lead to grammatically correct text, which simply means that the minimum number of documents with good English is likely to be higher.

In a way, this analysis provides elements to answer our main research question, regarding the value of our decentralized collaboration model. In this exercise, if the participants had been forced to choose a single solution, they would have had to give up on many of the popular (and valuable) changes that they had identified in the individual editing phase.

Again in the words of the class instructor,

> "The major benefit of the P2P system for our class was that students could see their peers' work in real time and then use the knowledge they gained from that experience to improve their own work. By using the P2P platform we were able to give students access to each other's work, while also encouraging group collaboration. Ultimately, this could be used to allow students to improve their work through exposure to the work of their peers, while also promoting the importance of the students' individual work."

In addition, the feedback from the students (Section 6) also seems to indicate that a large proportion of the students would not have submitted the group-edited versions for evaluation, but would have rather made their own tweaks to these versions. This suggests that the students also see value in having their own solution, which is usually not possible with traditional groupware.

5.3. Peer similarity

We study here the following question: if two peers make similar edits on one document, are they likely to make similar edits on a different document? If this is true, then the similarity between users' documents can be used as basis for collaborative filtering; for example, the application could

DOI: 10.1002/cpe

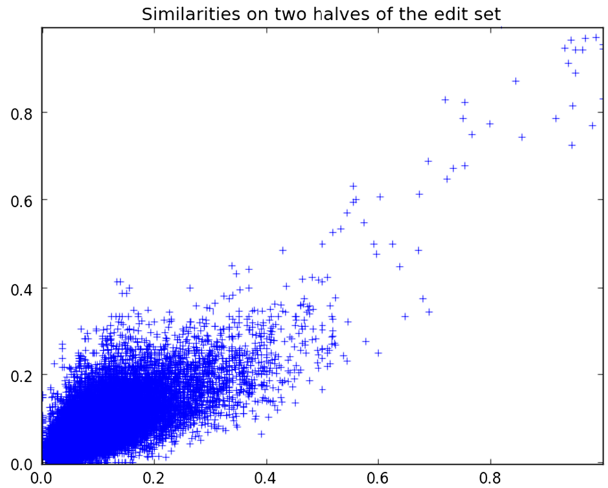


Figure 4. Similarity of the edits on the end of the documents, plotted as a function of the similarity of the edits on the beginning of the documents.

recommend edits that a user is likely to incorporate. More generally, we hope – although we cannot analyze this in this setting – that this extends to documents: users that have similar collections of documents will tend to agree on their 'rating' of other documents, that is, their decision to keep those documents or not.

In order to address this question, we divide the set of edits in two parts: the edits that apply to the beginning of the text (up to a certain character position), which we will call the set A of edits, and those that apply to the end of the text, the set B of edits.

Then, for each pair of documents,[††] we calculate how similar their edits were on the beginning of the text, that is, how similar their edits from set A were. Then, for those documents that have similar edits on the beginning of the text, we can estimate the similarity of the edits they made on the end of the text, that is, the similarity of their edits from set B.

We note $u_{d,A}$ the set of edits from A in a document d, and $u_{d,B}$ the set of edits from B in the same document.

The similarity of d_1 and d_2 on the beginning of the text is obtained by comparing the vectors $u_{d_1,A}$ and $u_{d_2,A}$, and their similarity on the end of the text is obtained by comparing the sets $u_{d_1,B}$ and $u_{d_2,B}$. We measure the similarity between sets of edits using the Jaccard coefficient (defined in Equation 2, in Section 3.3), as it is the similarity metric that P2Pedia offers to measure the similarity between peers. For all pairs of documents (d_1, d_2), we can thus compute two random variables X_A and X_B:

$$\begin{aligned} X_A &= jaccard\left(u_{d_1,A}, u_{d_2,A}\right) \\ X_B &= jaccard\left(u_{d_1,B}, u_{d_2,B}\right) \end{aligned} \tag{3}$$

We can then measure the correlation of the random variables X_A and X_B. As we are particularly interested in the cases where $u_{d_1,A}$ and $u_{d_2,A}$ are similar, we keep only the top 50% most similar pairs of documents (with respect to set A).

Calculating Pearson's correlation coefficient for X_A and X_B gives us a value of 0.44, that is, a high positive correlation. This correlation can also be seen graphically, by plotting one X_B as a function of X_A. This plot is shown in Figure 4.

Again, we can assess the statistical significance of this value by applying a randomization test. Here, we computed the correlation of X_A with 1000 random permutations of X_B, and obtained

[††]Note that with a few exceptions, the documents and peers are matched one-to-one.

DOI: 10.1002/cpe

correlation values confined to the interval $[-0.02, 0.02]$. This shows that this correlation is statistically significant with very high confidence.[‡‡]

6. USER PERCEPTION

After the tutorial, we proposed an online poll to the students in order to collect direct feedback on the usability of the tool, and on the collaboration model. We report here the results of this survey. Unfortunately, as only a small proportion of the students took the time to answer (62 out of 330 participants in total,[§§] that is, 19%.), we will not attempt to draw any strong conclusions from these figures. We hope to conduct more systematic user studies in the future.

6.1. Usability

A total of 89% of respondents found the interface easy to use (11% unsure); 82% found it easy to search and download their peers' versions (8% disagree and 10% unsure).

6.2. Benefit of the activities for the students' learning experience

A total of 79% thought that reading their peers' work contributed to their learning experience (6% disagree and 15% unsure); 62% thought that the group editing activity contributed to their learning experience (13% disagree and 25% unsure). Overall, 71% thought that the writing exercise was useful, and 90% thought that P2Pedia was an appropriate tool for the exercise.

6.3. Single versus multiple solutions

We introduced several questions for the purpose of evaluating whether it was beneficial to allow, ultimately, multiple solutions to the problem.

A total of 74% thought that it was easy to reach an agreement (a solution that everybody agreed on) in the group editing phase (8% disagree and 18% unsure); if they had to submit the result of the tutorial for evaluation, the students would either submit the group edited version (38%), or make their own tweaks to the group-edited version (56%), and 7% would choose to write their own version entirely. Finally, regarding the solutions, 64% thought that the instructor should select the best solutions and offer them as (multiple) solutions, whereas 36% thought the instructor should either write their own or select the best one to post as a single solution.

These responses suggest that while many students see a benefit in collaboration, either by being exposed to the solutions of others, or by actively working together in a group editing activity, a large part of them would still like to submit their own, personal solution; and it seems reasonable that the students would also want to see multiple solutions provided by the instructor.

7. LIMITATIONS AND FUTURE WORK

7.1. Technical limitations to scalability

As a wiki and more generally a collaborative application, it is highly desirable that P2Pedia can scale to many users, and large amounts of content.

The main technical limitation in this regard is the network traffic caused by P2Pedia usage, primarily browsing wikilinks, which translate to searches in the underlying P2P network.

[‡‡]We note that these extremely low values for the randomization test may be surprising, compared to those that could be obtained when validating the correlation of our scoring functions; these values are due to the fact that the variables X_A and X_B have very many values (several thousand), which makes it extremely unlikely that high correlations will occur by chance. In contrast, with only 10 values, there is approximately a 5% chance of obtaining a correlation above 0.5.

[§§]The data analyzed in the previous section was from a single session, while the survey was proposed to students after both sessions.

DOI: 10.1002/cpe

P2Pedia searches use a Gnutella-like broadcast, and are sent a maximum of two times along each edge of the social graph. The number of edges in the graph is, in the worst case, quadratic with respect to the number of nodes, so each search generates $O(n^2)$ messages in the network (n being the size of the network). Note that in a realistic 'small-world' topology, the number of edges is linear with respect to the number of nodes. However, these $O(n^2)$ messages are distributed over the entire network, in such a way that each node handles, in the worst case, only $O(n)$ of these messages, ($O(1)$ in the small-world scenario).

Assuming that all the participating nodes generate searches, the traffic figures in the preceding text are then multiplied by n: search activity in a network of size n will thus require each node to handle, in the worst case, $O(n^2)$ messages, or $O(n)$ in a small-world topology.

These results, although they ignore the mitigating factor of the bounded number of hops allowed in Gnutella, imply that beyond a certain size, the network would get congested and slow. Estimating the maximum number of participants that could be supported, in practice, is challenging, as it would require detailed models of user activity.

However, it is worth noting that despite predictions of doom and internet meltdown [25], Gnutella file-sharing networks still exist, with a 2007 crawl finding over 1.2 million users over a 24-h period [26].

In our experimental deployment described in this paper, university rules prevented us from installing P2Pedia in the labs, and we ran all the nodes on the same server, which the students could access. We had approximately 45 nodes running concurrently, and the searches generated the same traffic (not between machines but between ports of the same machine) as they would in a distributed deployment. We observed this workload to be quite manageable by a single machine, with some delays (up to a few seconds to see results appear) when all the users tried to search at the same time (due to the tutorial workflow).

7.2. Scalability of the decentralized collaboration model

In the experiments described in the paper, the users read all of the different versions submitted by their peers. As the number of users increases, this becomes infeasible. An important limitation of our study is that we have not observed what users might do when faced with very large numbers of versions.

This is when we would expect the users to rely on the trust indicators to rank and filter the versions, and thus identify those most relevant to them. This is similar to the millions of results of a Web search, of which users only explore the first few pages. The main difference is that Web search engines apply their sorting (ranking) function behind the scenes and outside the users' control, whereas we expect P2Pedia users to choose among several sorting criteria, and influence the ordering itself by their social behavior (e.g., 'friending' or 'following' particular users then using social distance as a sort function).

In future work, we hope to study this in larger deployments of P2Pedia.

7.3. Scenarios for future work

Further validation of P2Pedia's collaboration model is necessary. We do not have any data showing how the system may be used outside of the controlled workflow offered in the study. In particular, we have not been able to study the social networking aspect (allowing the users to choose their neighbours), and the associated trust indicators.

In the following, we outline scenarios where P2Pedia's collaboration model appears particularly relevant. These scenarios could be used for additional user studies.

7.3.1. Collaborative note taking. One application that appears very relevant is *collaborative note-taking*. Students are usually expected to take notes individually, and note-taking is an opportunity for the students to personalize their learning material, by noting salient points, keeping a trace of

DOI: 10.1002/cpe

oral explanations, and generally selecting material that they feel helps their understanding of the topic at hand.

Students may benefit from one another's lecture notes, and still want to study a version that is largely their own. Experimental deployment of P2Pedia for note-taking at Carleton University is one of our planned directions for future work. Our results could be compared with courses where students were expected to contribute material to traditional wikis [27–29].

7.3.2. Teaching material. Extending the previous idea, the course instructor could also participate in a note-taking wiki, using her own node, and collect (crowd-source) detailed notes from the students at the end of the course. This learning material could then in turn be shared with instructors of similar courses in other universities.

The Wikimedia Foundation project Wikiversity is a project inspired by Wikipedia, where university-level learning material is designed collaboratively, following the principles of traditional wikis.

In practice, we believe that instructors rarely make use of a single source of learning material, and often prefer to combine various textbooks, selecting material according to their personal views on the topic. A P2Pedia-based approach could be a better fit for this scenario.

8. CONCLUSION

Ward Cunningham, who invented the original wiki, considered that its centralized nature was a mistake [20]. Although centralized tools are the norm for computer-supported collaborative work, his intuition of a decentralized approach is interesting.

We have designed and implemented a P2P-based decentralized wiki system, P2Pedia, that allows for a fairly unique approach to collaboration. The main principle of this collaboration model is to allow users to work on the same problem while adopting different solutions.

In this study, we have demonstrated a scenario where the decentralized collaboration model was relevant and applicable: the considered problem could be solved in a collaborative writing environment such as a wiki, but has the following characteristics:

- The setting did not require a single, ideal solution (i.e., this was not the expected outcome by the instructor).
- The participants saw value in having their own solution, but also in having access to other good solutions.
- The similarity of participants on past contributions was indicative of future similarity: in other words, similarity of contributions would be a meaningful metric for users to find like-minded peers.
- The popularity of contributions (considered at a fine granularity) is indicative of academic value.
- These contributions, even the most popular ones, were not compatible; that is, they could not be merged into a single, ideal solution.

However, several relevant questions remain unanswered. We hope that wider deployments of P2Pedia, or of similar projects, such as Cunninghamn's SFW, might allow for more in-depth studies of decentralized collaboration.

We see a particular potential for our approach in an e-learning context.

The source code and an online demo of P2Pedia are currently accessible online from the project website `http://p2pedia.sf.net`.

ACKNOWLEDGEMENTS

This work is supported by the Carleton Strategic Academic Initiatives Fund.

REFERENCES

1. Leuf B, Cunningham W. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 2001.
2. Seigenthaler J. *A False Wikipedia 'Biography'*. USA Today, November 2005. (Available from: http://www.usatoday.com/news/opinion/editorials/2005-11-29-wikipedia-edit_x.htm) [Accessed on 30 September 2014].
3. Viégas FB, Wattenberg M, Dave K. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM: Vienna, Austria, 2004; 575–582.
4. Sumi R, Yasseri T, Rung A, Kornai A, Kertész J. Edit wars in Wikipedia. *SocialCom/PASSAT*, Boston, MA, USA, 2011; 724–727.
5. Chaudhuri S. Editors won't let it be when it comes to 'the' or 'The'. *The Wall Street Journal* October 2012. (Available from: http://online.wsj.com/article/SB10000872396390444657804578048534112811590) [Accessed on 30 September 2014].
6. Swicegood T. *Pragmatic Version Control Using Git*. Pragmatic Bookshelf, 2008.
7. Surowiecki J. *The Wisdom of Crowds*. Anchor, 2005.
8. Craig A, Davoust A, Esfandiari B. A distributed wiki system based on peer-to-peer file sharing principles. *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2011, Lyon, France, August 23 - 25, 2011*, Lyon, France, 2011; 364–371.
9. Davoust A, Craig A, Esfandiari B, Kazmierski V. Decentralized collaboration with a peer-to-peer wiki. *Proceedings of CTS'2012*, Denver, Colorado, USA, 2012; 286–293.
10. Bergsma M. Wikimedia infrastructure, 2007. (Available from: http://www.nedworks.org/~mark/presentations/san/Wikimediaarchitecture.pdf) [Accessed on 30 September 2014].
11. Mukherjee P, Leng C, Schurr A. Piki - a peer-to-peer based wiki engine. *P2P '08. Eighth International Conference on Peer-to-Peer Computing, 2008*, Aachen, Germany, 2008; 185–186. DOI: 10.1109/P2P.2008.33.
12. Oster G, Mondéjar R, Molli P, Dumitriu S. Building a collaborative peer-to-peer wiki system on a structured overlay. *Computer Networks* 2010; **54**(12):1939–1952.
13. Weiss S, Urso P, Molli P. Wooki: a p2p wiki-based collaborative writing tool. In *Proceedings of the 8th International Conference on Web Information Systems Engineering, WISE'07*. Springer-Verlag: Berlin, Heidelberg, 2007; 503–512.
14. Morris JC. DistriWiki:: a distributed peer-to-peer wiki network. In *WikiSym '07: Proceedings of the 2007 International Symposium on Wikis*. ACM: New York, NY, USA, 2007; 69–74. DOI: 10.1145/1296951.1296959.
15. Rozet S. git based wiki, 2008. (Available from: http://atonie.org/2008/02/git-wiki) [Accessed on 30 September 2014].
16. Mendler D. Olelo, a wiki with a git backend, 2013. (Available from: http://www.gitwiki.org/) [Accessed on 30 September 2014].
17. Gollum. A wiki built on top of git, 2010. (Available from: https://github.com/github/gollum.git) [Accessed on 30 September 2014].
18. Rahhal C, Skaf-Molli H, Molli P, Weiss S. Multi-synchronous collaborative semantic wikis. In *WISE*, vol. 5802, Vossen G, Long DDE, Yu JX (eds)., Lecture Notes in Computer Science. Springer, 2009; 115–129.
19. Dourish P. The parting of the ways: divergence, data management and collaborative work. *European Conference on Computer Supported Cooperative Work*, Stockholm, Sweden, 1995; 213–213.
20. Finley K. *Wiki Inventor Sticks a Fork in His Baby*. Wired Enterprise, 2012.
21. Chandrasekaran P, Esfandiari B. A model for a testbed for evaluating reputation systems. In *Proceedings of the 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, TRUSTCOM '11. IEEE Computer Society: Washington, DC, USA, 2011; 296–303. DOI: 10.1109/TrustCom.2011.40.
22. Creole wiki markup specification v1.0, 2007. (Available from: http://www.wikicreole.org/wiki/Creole1.0) [Accessed on 30 September 2014].
23. Goldberg KY, Roeder T, Gupta D, Perkins C. Eigentaste: a constant time collaborative filtering algorithm. *Information Retrieval Journal* 2001; **4**(2):133–151.
24. Adamic LA. Zipf, power-laws and pareto - a ranking tutorial, 2000. (Available from: http://ginger.hpl.hp.com/shl/papers/ranking/ranking.html) [Accessed on 30 September 2014].
25. Ritter J. Why gnutella can't scale. no, really, February 2001. (Available from: http://www.cs.rice.edu/~alc/old/comp520/papers/ritter01gnutella-cant-scale.pdf) [Accessed on 30 September 2014].
26. Shavitt Y, Weinsberg E, Weinsberg U. Mining music from large-scale, peer-to-peer networks. *IEEE MultiMedia* 2011; **18**(1):14–23.
27. Wheeler S, Yeomans P, Wheeler D. The good, the bad and the wiki: evaluating student-generated content for collaborative learning. *British Journal of Educational Technology* 2008; **39**(6):987–995. DOI: 10.1111/j.1467-8535.2007.00799.x.
28. Cole M. Using wiki technology to support student engagement: lessons from the trenches. *Computers & Education* 2009; **52**(1):141–146. DOI: 10.1016/j.compedu.2008.07.003.
29. Minocha S, Thomas PG. Collaborative learning in a wiki environment: experiences from a software engineering course. *New Review of Hypermedia and Multimedia* 2007; **13**(2):187–209. DOI: 10.1080/13614560701712667.