Synchronous Collaborative Text Editing in Wikis

Jessica Rubart
OWL University of Applied Sciences
An der Wilhelmshöhe 44, 37671 Höxter
Germany
jessica.rubart@hs-owl.de

ABSTRACT

Wikis are well-known for supporting collaborative writing. They are focusing on asynchronous collaboration. Today, synchronous text editing in the web is supported by several tools and approaches. However, this possibility is still missing in wikis. Based on a prototype implementation, this paper presents design considerations for a wiki integrating synchronous collaborative editing as a special kind of hypertext authoring.

CCS CONCEPTS

• **Human-centered computing → Collaborative and social computing systems and tools**; Wiki; Synchronous editors • **Human-centered computing → Interaction paradigms**; Hypertext / hypermedia; collaborative interaction

KEYWORDS

Wiki, collaborative editing, synchronous editor, real-time editor

ACM Reference format:

J. Rubart. 2017. Synchronous Collaborative Text Editing in Wikis. In *Proceedings of the Workshop on Narrative and Hypertext'17, Prague, Czech Republic, July 2017 (NHT'17)*, 5 pages, CEUR-WS.org.

1 INTRODUCTION

Wikis support asynchronous collaborative editing of web pages. The first wiki was invented by Ward Cunningham in 1995 [6][15]. It was called "WikiWikiWeb" as a substitute for quick web ("wikiwiki" is a Hawaiian word for "quick") [5]. The application scenario was editing and sharing software engineering design patterns. Design patterns aim at reusing object-oriented design structures, i.e. descriptions of communicating objects, interfaces, and classes to solve a general object-oriented design problem in a particular context [10]. Both, design patterns and wikis, have originally been developed as agile forms of technology following a simple reusable system [6]. The wiki idea was to provide a website, which allows users to quickly and easily share, modify, and improve information in a knowledge base collaboratively [15].

Today, wikis are not only used for best practice patterns in software engineering, but also for sharing experiences in several other domains, such as in project management or IT service management. Storytelling is, amongst others, a very important knowledge management instrument supporting the understanding and sharing of experiences.

The most famous wiki is the online encyclopedia Wikipedia [6]. It is based on the Wikimedia technology[2]. There are currently 286 active Wikipedia instances based on different languages. In total, there are presently more than 42 million articles [19] and the number is increasing strongly. However, if multiple users edit the same page at the same time, this causes an edit conflict [20]. The user saving the text at a later point in time than others needs to merge the changes manually. This can be error-prone. Through the versioning of wiki pages, errors can be resolved later, but this can cause much work.

Synchronous collaborative editing is a special case, but in the context of edit conflicts it might be very helpful. Today, synchronous text editing in the web is supported by several tools and approaches. However, this possibility is still missing in wikis, even though it can help supporting the agile idea of self-organizing teams.

In the following, this paper presents related work, a proof of concept implementation of a synchronous collaborative wiki, and first usage feedback. Afterwards, it focuses on design considerations for such a wiki. The paper ends with conclusions and future work.

NHT'17, Prague, Czech Republic, July 2017

[2] https://www.wikimedia.org/

2 RELATED WORK

Hypertext authoring has been worked on in several directions. In the context of a personal authoring environment, *NoteCards* [13] is an important approach, for example. In *NoteCards*, notes are represented as cards holding text and images, and being interconnected with typed links. Fileboxes are a kind of composite notes useful to organize large collections of notecards. In parallel, HyperCard [11] came up as a simpler hypertext authoring tool for users of Mac computers. Ward Cunningham was involved in the development of HyperCard. It influenced his first wiki project [5].

In gIBIS [3], hypertext networks based on the *Issue-Based Information System* (IBIS) approach can be created asynchronously in a group. IBIS is an argumentation-based approach focusing on problem solving. Cooperative hypermedia approaches, such as in [16], focus on collaborative structuring – both asynchronously as well as synchronously with fine-grained notifications of other users' interactions. In the context of hypertext narrative there is *Tinderbox* [2], for example. Tinderbox is a tool for making, analyzing, and sharing notes focusing on spatial hypertext, informal semantics, and web collage. Hypertext notes can be published on the web.

The real-time editors Etherpad[3] and Google docs[4] support web-based synchronous collaborative text editing based on operational transformations as replication and concurrency control mechanism [1]. This shows that the technology for synchronous collaborative text editing is basically available and can be integrated in wikis. Webstrates [4] is a research project and a development platform. It provides a Webstrates server that persists and synchronizes changes to the DOM of any page served by the Webstrates server. Based on this, flexible collaborative hypermedia systems can be built. For synchronous collaborative editing, the ShareJS[5] library is used.

In [17], patterns for the design of computer-mediated interaction between humans are described focusing on reusable best practices. One pattern is called "Shared Editing" and focuses on the provision of an editor, with which users can manipulate shared artifacts simultaneously. Studies, such as in [18], focus on investigating user behavior during real-time collaboration. It shows that there are also differences in cultures. For example, in the context of slide sharing, U.S. participants preferred individual navigation through the slides, while Japanese participants wished there was a moderator.

3 PROOF OF CONCEPT

As a proof of concept, a wiki providing synchronous collaborative text editing has been implemented [7]. Figure 1 shows a screen dump of the current prototype. Articles, i.e. wiki pages, can be selected, edited, newly created, or deleted. Presence awareness is provided, i.e. user names currently editing one and the same article are visualized. In addition, editing awareness is provided through the visualization of remote cursors.

3.1 Implementation

For integrating synchronous collaborative editing possibilities, we have used ShareJS. The ShareJS server uses operational transformations [1] [8], an operation-based synchronization mechanism for real-time collaboration. ShareJS does not yet support shared cursors so that the visualization of remote cursors is implemented in an application-specific way based on the Ace[6] editor.

For this prototype implementation, MeteorJS[7] has been used as a platform for developing the wiki in JavaScript. In addition, it provides hosting and deployment features for testing and evaluating applications. Based on this service, first users have been able to access the prototype. MeteorJS is written using NodeJS[8]. We have integrated MongoDB[9] as the database backend. As stated above, ShareJS is used for synchronous editing. Three editors have been integrated: A simple textarea, the Ace editor, and CodeMirror[10]. The Ace editor and CodeMirror can be customized with various themes. The input given in the editor is assumed to be in the markdown[11] format. The application converts the given markdown text to HTML. As soon as two or more users edit the same article, they get synchronized.

TogetherJS[12] is another JavaScript library, which supports collaborative text editing. In addition, it already supports shared cursors and integrates a text chat as well as an audio chat via WebRCT[13]. In our prototype, the communication features of TogetherJS have been integrated. While ShareJS synchronizes JavaScript models, TogetherJS integrates directly through the DOM (Document Object Model)[14].

[3] http://etherpad.org/

[4] https://docs.google.com/

[5] https://github.com/share/ShareJS/

[6] https://github.com/ajaxorg/ace/

[7] https://www.meteor.com/

[8] https://nodejs.org/

[9] https://www.mongodb.com/

[10] https://codemirror.net/

[11] https://tools.ietf.org/html/rfc7763/

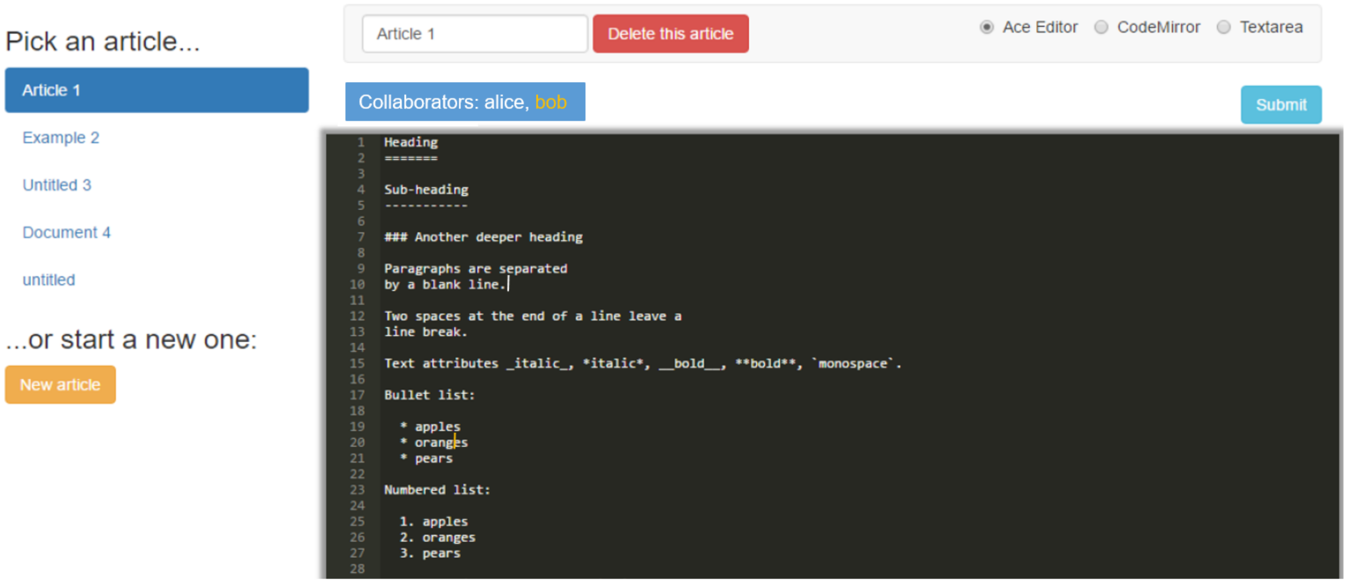
[12] https://togetherjs.com/

[13] https://webrtc.org/

[14] https://www.w3.org/DOM/

3.2 First Usage Feedback

We did a first evaluation with a group of students using Groupware Observational User Testing [12]. With this evaluation method, evaluators observe users performing collaborative tasks in a laboratory setting. Problems are monitored and users are asked to think aloud about what they are doing. In addition, we have asked about feedback through a final questionnaire. The students were distributed in different rooms and were asked to edit several web pages – similar to articles in Wikipedia. Mostly, they have used the chat to communicate and coordinate themselves. The audio chat did not work properly on all browsers. Communication is very important in synchronous collaboration settings.



● Ace Editor ○ CodeMirror ○ Textarea

Collaborators: alice, bob

```
Heading
=======

Sub-heading
-----------

### Another deeper heading

Paragraphs are separated
by a blank line.

Two spaces at the end of a line leave a
line break.

Text attributes _italic_, *italic*, __bold__, **bold**, `monospace`.

Bullet list:

  * apples
  * oranges
  * pears

Numbered list:

  1. apples
  2. oranges
  3. pears

```

Figure 1: Proof of Concept Implementation Using ShareJS

As a result, the prototype system basically illustrates the advantages of integrating synchronous editing in wikis using operational transformations. Conflict situations and complex merging efforts can be avoided. However, this first evaluation also identified some drawbacks of our prototype system:

For using the communication tool, the users needed to manage a session by themselves, i.e. to invite the other users whereas the shared editing was coupled automatically.

Not all integrated editors included the visualization of remote cursors. The cursors are important awareness instruments. Without remote cursors, the users did not know immediately where the others were editing.

Students, who did not know the markdown syntax well, needed a lot of time to search and write according to the syntax. For them, the synchronous collaboration helped to learn the language.

In a few cases the response time of the editor was considered slow.

4 DESIGN CONSIDERATIONS

In this section, we will look at and discuss design considerations from two perspectives, which are detailed in the following subsections:

- Collaborative writing strategies

as well as

- Groupware and hypermedia design issues

4.1 Collaborative Writing Strategies

There are five well-known collaborative writing strategies [14].

Firstly, a single author might write on behalf of a team. In this case, from the technical point of view collaborative authoring features can be reduced to sharing the results.

In sequential single writing, the shared document is passed from one author to the next one, after the author has written her or his part of the document. For this strategy, traditional wikis are already well-suited as edit conflicts cannot occur.

In parallel writing, authors work on their part of the document at the same time, respectively. This can be managed with a traditional wiki assigning each group member a separate starting wiki page. However, if parallel writing does not mean separate sections, but different roles in the authoring process, such as author, editor, or reviewer, then edit conflicts might occur (dependent on the work process).

In reactive writing, a document is created in real-time without substantial preplanning. It is also known as joint writing, consensus writing, or reflective writing. Authors react to each other's changes or additions, e.g. by creating new sections or changing existing ones. In reactive writing, a high level of consensus is needed and can be developed. Version control is more complex and synchronous collaborative writing is necessary.

Finally, in mixed mode two or more of the collaborative writing strategies are combined so that synchronous collaborative writing can avoid edit conflicts and support consensus.

4.3 Groupware and Hypermedia Design Issues

According to well-known Groupware experiences, such as in [9], it is important to think about awareness features – features that make visible other users and other users' interactions, such as presence awareness and shared cursors. Also, in asynchronous collaboration, awareness features are important, e.g. to show which relevant wiki pages have been updated since the last login.

Another important issue is floor control. Shall authors be able to write simultaneously or is there a moderator or a predefined process passing the floor to the different authors? In our proof of concept implementation, we give authors, editing the same page at the same time, simultaneous writing possibilities. In this way, reactive writing is possible so that consensus can be built and editing conflicts can be avoided.

Session control is another issue to consider. Do authors need to invite others? In our prototype system, all authors, editing the same page at the same time, are automatically in a shared editing session. It could also be useful to ask writers, with which writing strategy they would like to continue. In this context, a voting mechanism can be helpful. A shared session should also establish a communication channel automatically.

A joint editing session can either be planned or take place by chance. In any case, it is helpful to provide support for latecomers, i.e. people joining later than others. Can they understand what happened? In our prototype system, they can scroll through the document and see what has been edited. In addition, versioning support is usually useful. Furthermore, dedicated latecomer functionality might be helpful, such as recording and playing the session history.

Another issue are customized views. In our prototype, we have integrated three different editors, but the authors see and work on the same text. In addition, they can scroll individually. In this context, it needs to be decided which parts of the shared data need to be coupled – the text or the text and the navigation? Or can this be customized?

It is also interesting to think about the integration of WYSIWYG editors as not all authors want to use markup languages.

Finally, are there any privacy concerns that need to be considered? Usually, it is important to give users control over which information gets shared and what remains private.

According to Halasz' seven issues for hypermedia systems [13], versioning is an important feature to maintain and manipulate a history of changes. Wikis already incorporate versioning support. When integrating synchronous collaborative writing, one needs to differentiate versions necessary for concurrency control from document versions, which the authors would like to submit. Here, it might also be useful to integrate decision support to help authors to decide when to submit a new joint version. Alternatively, the system could be able to submit sections of a wiki page separately as new versions.

Regarding collaborative work, Halasz points out that a high degree of concurrent work is important. From the viewpoint of the collaborative writing strategies, this is the case with reactive writing. In asynchronous collaboration, locking mechanisms shall be fine grained. Synchronous collaborative editing capabilities complement this view.

5 CONCLUSIONS AND FUTURE WORK

We have motivated the integration of synchronous collaborative editing capabilities in wikis and presented a proof of concept implementation. Based on this, we have described first usage feedback and design considerations for this special kind of hypertext authoring.

Currently, we are exploring further JavaScript libraries for this hypertext authoring purpose to improve the prototype implementation.

In our future work, we would like to make a user study among Wikipedia authors to get more feedback and requirements regarding synchronous editing in wikis.

REFERENCES

[1] Mehdi Ahmed-Nacer, Claudia-Lavinia Ignat, Gérald Oster, Hyun-Gul Roh, and Pascal Urso. 2011. Evaluating CRDTs for Real-time Document Editing. In *Proceedings of the* ACM *Symposium on Document Engineering (DocEng'11)*, Mountain View, California, USA, ACM Press, New York.

[2] Mark Bernstein. 2003. Collage, Composites, Construction. In *Proceedings of Hypertext'03*, ACM Press.

[3] Conklin, J., and Begemann, M.L. gIBIS 1988. A Hypertext Tool for Exploratory Policy Discussion. In *Proceedings of Computer Supported Cooperative Work (CSCW)*, ACM Press, pp. 140-152.

[4] Niels Olof Bouvin and Clemens N. Klokmose. 2016. Classical Hypermedia Virtues on the Web with Webstrates. In *Proceedings of Hypertext and Social Media 2016 (HT'16)*, ACM Press.

[5] Ward Cunningham. 2014. *Wiki History*. http://wiki.c2.com/?WikiHistory. Retrieved June 15, 2017.

[6] Ward Cunningham and Michael Mehaffy. 2013. Wiki as Pattern Language. In *Proceedings of the 20th Conference on Pattern Languages of Programs (PLoP'13)*, Monticello, Illinois, USA, ACM Press, New York.

[7] Sandeep Reddy Dareddy. 2016. *Integrating Operational Transformations in Wikis: Synchronous Editing in Wikis*. Master's thesis. OWL University of Applied Sciences, Lemgo, Germany.

[8] C. A. Ellis and S. J. Gibbs. Concurrency Control in Groupware Systems. In: *SIGMOD Record: Proceedings of the ACM SIGMOD Conference on the Management of Data - SIGMOD '89*, 18(2):399-407, May 1989.

[9] C. A. Ellis, S. J. Gibbs, and G. Rein. 1991. Groupware: Some Issues and Experiences. In *Communications of the ACM*, Vol. 34, No. 1.

[10] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

[11] D. Goodman. Complete HyperCard 2.0 Handbook. Random House Inc., 1990.

[12] Carl Gutwin and Saul Greenberg. 2000. The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. In *Proceedings ot the 9th IEEE Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '00),* 98–103. IEEE Comp. Society

[13] F.G. Halasz. 2001. *Reflections on NoteCards: seven issues for the next generation of hypermedia systems*. In: ACM Journal on Computer Documentation (New York, NY, USA: ACM Press) 25 (3), pp. 71–87. doi:10.1145/507317.507321, reprint.

[14] Paul Benjamin Lowry, Aaron Curtis, and Michelle R. Lowry. 2004. Building a Taxonomy and Nomenclature of Collaborative Writing to Improve Interdisciplinary Research and Practice. In *Journal of Business Communication*, Vol. 41, No. 1.

[15] Bo Leuf and Ward Cunningham. 2001. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, London.

[16] J. Rubart, J.M. Haake, D.A. Titze, and W. Wang. 2001. Organizing Shared Enterprise Workspaces Using Component-Based Cooperative Hypermedia. In *Proceedings of Hypertext'01,* ACM Press.

[17] T. Schümmer, and S. Lukosch. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons. 2007.

[18] L. Scissors, N. S. Shami, T. Ishihara, S. Rohall, and S. Saito. Real-Time Collaborative Editing Behavior in USA and Japanese Distributed Teams. In *Proceedings of the ACM International Conference on Human Factors in Computing Systems - CHI 2011*, pages 1119-1128, Vancouver, BC, Canada, May 2011. ACM Press.

[19] Wikimedia. 2017. List of Wikipedias. https://meta.wikimedia.org/wiki/List_of_Wikipedias. Retrieved June 15, 2017.

[20] Wikipedia Help: Edit Conflict. 2017. https://en.wikipedia.org/wiki/Help:Edit_conflict. Retrieved June 15, 2017.