Supporting the Sense-Making Processes of Web Users by Using a Proxy Server

Teppo Räisänen
Department of Information Processing Science, University of Oulu, Finland
Teppo.Raisanen@oulu.fi

Abstract
This paper presents a study on how we can support knowledge creation - especially a process called comprehension - in a Web 2.0 environment by providing new functionalities to users of existing Web services. The contribution of this paper is twofold. Firstly, a framework for providing new functionalities is presented. Secondly, a prototype Web service is implemented and evaluated. The prototype uses Wikipedia as an example and as a knowledge repository. The emphasis is on the prototype, a service that allows us to 1) insert sticky notes in Wikipedia articles, 2) enhance the translation capabilities of Wikipedia, and 3) highlight texts in Wikipedia. The analysis of the prototype service shows that we can provide new functionalities to Web users with a proxy server and that the implemented tools offers some support for knowledge creation process called comprehension. The translation service proved especially useful.

1. Introduction

As technologies and protocols evolve, users and developers invent new ways (and revise existing ones) to utilize the Internet. The evolution of the technologies is most evidently seen on the World Wide Web, where companies like Microsoft, Google and Yahoo! – among others – create new programming environments, paradigms, API’s, and services in an ever-accelerating pace.

The second generation of these Internet-based services, which emphasize online collaboration and sharing of knowledge between users, are referred to as Web 2.0. According to O’Reilly [20], some of the key concepts and technologies associated with Web 2.0 are 1) Web as platform, 2) architecture of participation, 3) rich user experience, 4) blogging, and 5) Wikis. Following O’Reilly [20] and Räisänen & Oinas-Kukkonen [25], we define Web 2.0 as a set of novel technologies and philosophies that use the Web as a platform to deliver services\(^1\) that emphasize user participation.

With the success of the new Web applications, the knowledge management paradigm is changing its focus from the management of organizational knowledge into management of social knowledge constructed mainly by communities of practice within the Web. One example of how knowledge is created in Web 2.0 environment is the wiki [14]. The most famous example of the power of wiki technology is probably Wikipedia (http://en.wikipedia.org).

In this paper, we will use a knowledge management framework called the 7C model [23] to understand the various knowledge creation processes, and how to support them in the Web 2.0 environment. An environment supporting the 7C model has been presented by Räisänen & Oinas-Kukkonen [25], and it provided technologies and concepts that can be applied when designing support for the processes presented in the 7C model. The environment identified some existing tools that can be used to support the 7C processes (e.g. Wiki’s), but it did not specify the needed functionalities very thoroughly.

The aim of this paper is to design and implement a simple and extendable prototype that allows new tools to be easily integrated into the 7C environment. To achieve this, we will use a design science approach [13]. Design science attempts to “create things that serve human purposes” [17], e.g. to build and evaluate an information system that would help Wikipedia readers to better comprehend the contents of the articles they read. To evaluate the implemented prototype system, we performed interviews for qualitative analysis to gain insights on how to support comprehension.

We argue that this prototype and 7C tools should be implemented as Web services [2]. By doing this, the 7C tools can be used with any other Web services to create environments that offer support where needed. We will also show that when implementing 7C tools as

\(^1\) Web service here refers to any Web-based “software applications identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts” [26].
Web services, they can be integrated with other Web services by using a proxy server. We demonstrated this by building three simple 7C tools: translation, annotation, and highlight. They were integrated with Wikipedia to help readers learn and understand articles better. The system was then tested with six participants to see if it offers any support for the users.

The rest of this paper is organized as follows. Chapter 2 presents background for the study. Chapter 3 presents a solution to provide new functionalities for Web users. Chapter 4 presents a proof-of-concept implementation and a preliminary evaluation of the system. Finally, Chapter 5 discusses the findings and Chapter 6 concludes the paper.

2. Background

We use the 7C knowledge creation framework [23] to understand how knowledge is created (and maintained) in the Web 2.0 communities. The 7C model describes the processes through which knowledge is created, shared and applied. It is based on Nonaka and Takeuchi’s [19] SECI-model, and it assumes that new knowledge can emerge from the interaction between tacit and explicit as well as individual and social knowledge. According to the 7C model [23], the key processes in knowledge creation are comprehension, communication, conceptualization and collaboration (see figure 1).

2 The processes in the SECI-model are socialization, externalization, combination and internalization.

Figure 1. The 7C model.

The 7C model states that the users must be provided a rich environment in which they can interact with existing knowledge and information in order to comprehend something new or innovative [23]. The interaction must go deeper than just browsing or reading existing content [25]. The user must be able to link things together (e.g. tag), and combine and play with the content (e.g. annotate). For example, “an associative link between two knowledge objects would explain to the user that these objects are somehow related or that they have something in common” [25].

Through better interaction with existing knowledge, the users may gain new tacit knowledge. In the 7C knowledge creation spiral, this is a process called comprehension. Oinas-Kukkonen [23] defines it as “a process of surveying and interacting with the external environment, integrating the resulting intelligence with other project knowledge on an ongoing basis in order to identify problems, needs and opportunities.” In psychology, comprehension refers to the understanding of individual stimuli, especially words, sentences or chunks of prose [6]. Oinas-Kukkonen’s definition of comprehension is also equivalent to the process of sense-making, which is defined as “a motivated, continuous effort to understand connections (which can be among people, places, and events) in order to anticipate their trajectories and act effectively” [15].

The rich user experience (see [1]) offered by Web 2.0 technologies offers a good platform for this comprehension. Also the use of hypertext functionalities has been identified to support the comprehension [25]. Examples of possible hypertext functionalities would be annotation, deep linking and tagging. To allow the users to truly interact with the existing knowledge, hypertext must be provided in a richer way than with static Web pages (or even with dynamic Web pages) [25]. Hypertext functionalities can promote options and allow freedom of choice with contextual support. This can provide users with a rich environment for comprehension [23].

For example, Räisänen and Oinas-Kukkonen [25] argue that the usage of “personal and shared tags supports comprehension by providing the kind of associative linking which enables the user to recognize similarities and possibly [...] identify specific needs and opportunities as well as potential problems.”

Another way to support comprehension could be to provide the users an optimum user experience called flow [7]. According to studies, they could have a better chance of learning with such an experience [12]. As such, by improving user’s changes of flow, we can support his comprehension, too. Many times this can mean allowing the user to stay within flow [24], as interruptions can mean the optimum user experience is disturbed, and the user may drop ‘out of flow’.

Providing the users a way to see how others have interacted with the existing knowledge should also provide the user a way of comprehending something new. The user might be able to think outside of his
world view and letting him know how others see the same thing could be an eye-opener.

The user must also be able to share his newly gained knowledge and innovations with others, preferably in real-time and face to face. In the 7C model this process is called communication. Many times new ideas might be lost if there is no one to share them with the moment they happen, or at least while they are still “fresh.” The ability to share ideas face to face, and as they emerge, would also be important; we would have a better chance of understanding the shared ideas because the context of where and when the ideas emerged would still be present. When the sharer and receiver are out-of-context, the ability to remember and understand what is being shared diminishes. Conversely, while in-context, the sharer has a better chance of articulating and sharing the ideas, and the receiver has a better chance of understanding them.

Sharing the ideas in real time and face to face can help in comprehension as the possible feedback from other persons can help the comprehension process. This helps in reducing premature segmentation [21] as people can refine and represent the ideas before they write them down or enter them into an information system. In another words, users can share ideas “when one’s thoughts are [...] mature enough for externalizing a [...] rationale” [21].

People are often in possession of relevant knowledge in given situations, but for one reason or another they choose not to share it. Such reasons may be lack of confidence, fear of rejection, or uncertainty regarding the consequences of sharing the knowledge. In the Web people can have different identities, which could help. John Doe might be afraid that people will laugh to his ideas but Terminator123 (John’s login name to a Web 2.0 portal) can present the same ideas anonymously.

Another reason may be related to the way people equate knowledge with power. For example, employees may want to secure their organizational status by sitting on knowledge, making themselves ‘irreplaceable’ but at the same time detracting from organizational knowledge creation and development. In the Web environment, this should not pose such a big problem as the power of knowledge is not as apparent as it is in organizations.

When users are sharing their comprehensions and innovations, they need to reach a shared understanding on the issue at hand. In the 7C model this process is called conceptualization. It is a collective process of forming explicit concepts out of the different comprehensions and ideas that the users have shared. The outputs of the conceptualization process are explicit concepts that can be used to perform the actual task at hand. Many times, users have to reach a consensus to form the concepts. Providing tools to support comprehension and communication also helps in conceptualization as users must share their ideas and understand each other before they can reach shared understanding of the task they are performing.

After users have reached a shared understanding through the conceptualization process, they can start applying the produced concepts in group effort called collaboration. It is the process whereby the users produce explicit knowledge. This could be a Wikipedia article, design document, or a set of functions written in Java. Through collaboration, users have a chance of comprehending. Thus the 7C model is a spiral. Through each cycle of the spiral, the group can become better at applying its expertise (this is called collective intelligence in the 7C model).

Since the processes of communication (see e.g. [18]) and collaboration (see e.g. [8]) have received a lot of attention, we will not focus on them. Instead we focus on supporting comprehension. This will also indirectly support conceptualization, but studying that is beyond the scope of this paper.

Section 2.1 will investigate research finding related to 7C model, and annotation.

2.1 Related research

The 7C model has been presented in [23]. It has been used, e.g., to understand online collaboration [28]. Räisänen and Oinas-Kukkonen [25] have defined the system architecture for the 7C knowledge environment. They argued for the use of various Web 2.0 technologies to enable the kind of knowledge creation spiral that the 7C model poses. They found Wikis to be suitable as a basis of the 7C environment. However, they did not identify specific tools that could be used to support various processes. They concluded that the most crucial parts of the 7C model are comprehension and conceptualization. We will follow this line of thinking and concentrate on supporting comprehension of users reading Wikipedia articles.

Hypertext functionality and knowledge rationale [25] have been identified to offer support for comprehension. Thus if we can provide users with a new (and useful) functionalities, or allow him to investigate the rationale behind knowledge objects (e.g. Wikipedia articles) he could have better chances of comprehending the information he is reading or studying.

Comprehension is typically supported only indirectly in Web-based services like Wikipedia. Both sticky notes and text highlighting can be seen as a way of annotation, and the ability to annotate has been considered a "basic tool for collaboration and exchange of ideas" [4]. As such, annotations should support the
7C processes of communication (i.e., exchange of ideas) and collaboration. Later on this paper we will investigate if they also support comprehension. Annotations are also considered an important part of the semantic Web [3], which is a project that tries to facilitate information exchange by bringing structure to the meaningful content of the Web.

Typically, annotations are made [27] 1) for the readers themselves, 2) for the author of the annotated text, or 3) for other readers of the text. They have been found to support concentrated, intensive reading [27], help re-reading, learning, and knowledge sharing [11]. Concentrated, intensive reading combined with improved learning should - in theory - help comprehension, too.

There have been many different implementations to support annotations within the Web -- Anotea [14], Crit [29] and eLAWS [11] to name a few. Most of the existing solutions are from the "Web 1.0" era. Thus using Web 2.0 solutions could provide some new insights into annotations.

3. Providing new functionalities to Web users

This chapter will present our solution for supporting individual comprehension while taking account of other 7C processes as well. We want our solution to be a Web service that can be used with any existing Web services. This way we can combine our tools with any other Web service and create mashups that support – at least partially – the knowledge creation processes of the 7C model.

One way to increase the functionalities of the Web is to use browser plug-ins (or add-ons). They interact with the host application – usually a Web-browser – to provide certain functionalities. An example of a browser plug-in is Flash player, which allows browsers to play Flash animations and movies that are embedded on a Web page. YouTube (www.youtube.com) uses such a plug-in.

However, plug-ins have a few drawbacks. For example, the 7C model poses that the better the connectivity of the users, the better we can support the knowledge creation processes [23]. Those users who do not have the right plug-in installed cannot access the system, which lowers connectivity [25]. Plug-ins are also usually browser specific, so we would need to have different versions of the plug-in for different browsers, which again lowers connectivity. We would also need to have plug-ins for mobile environment to have best support for connectivity. So finding a way to provide the same functionality without necessary installs would be better.

Implementing 7C tools as Web services could do just that. In fact, we argue that all the 7C tools should be implemented as Web services so that these tools can be used with any other Web services to create environments that offer support where needed.

This allows us to use the service from different platforms, both stationary and mobile. This would increase connectivity, which is crucial as “users must have access to the system whether they are working at home or in the office” [25] or on the move. Having 7C tools as Web services should, in theory, allow us to use them via any device that has a suitable browser.

As we want the prototype to be extendable, we will use JavaScript to offer the new functionalities. There are many existing JavaScript archives on the Web; using them will help us because the existing scripts are usually tested (i.e. they work and they are bug free) and compatible with various browsers. In this way, we can simply choose existing script and a service we want to use it with, then include the script into our system, usually with only minor changes. Using JavaScript means that we will compromise little with connectivity (as browsers must support JavaScript in order to use the system) but we will still have much better connectivity than e.g. plug-ins would offer. Figure 2 shows the high-level description of how the prototype system works. Similar solutions have been presented before, see e.g. [11].

![Figure 2. High-level description of the framework.](image-url)

In short, the user opens a connection to the 7C Server (arrow 1 in figure 2) and requires it to display the original Web service with enhanced functionalities (or 7C tools). The 7C Server will then send an HTTP request to the original Web service (arrow 2). The original Web service will then return the HTML code (e.g. index.html) to the 7C Server (arrow 3). The 7C server will then include the new functionalities in a form of JavaScript to the original HTML code (arrow 4). Finally, it will return it to the user and his browser will render the original service enhanced with new functionalities. The 7C Server itself works as a proxy between the client and the content (this case Wikipedia). It is a combination of browser and server functionalities: the user must be able to connect to it.
Csikszentmihalyi [7] describes flow as "the holistic sensation that people feel when they act with total involvement." In human-computer interaction [5], flow has been shown to increase learning and creativity. Pearce and Howard [24] state that "flow activity is one in which the mind becomes effortlessly focused and engaged, rather than falling prey to distractions." They follow Draper [9], who states that users can "flick in and out" of flow from moment to moment. This view sees flow more as a process than as a state, and any distraction that happens causes the user to flick out of flow. This is especially important if the user was just about to comprehend something. A longer distraction can mean that the user loses the 'opportunity moment’ to learn something.

Both learning and creativity help the knowledge creation spiral, and they allow new knowledge to emerge. In fact, without learning it is difficult to see individuals gaining any new knowledge. In the 7C model, learning supports comprehension. Through learning, the user gains new insights which help him to comprehend "problems, needs and opportunities [...] and [...] embodying explicit knowledge in tacit knowledge" [23].

By giving users better chances for learning and creativity (i.e. keeping them in flow), we can also support comprehension. In this study, we aim to help users to stay in flow by improving existing Web translation services.

Web translation usually works in two ways. The first way is that the users can copy the link of a selected page and use an existing service to translate the whole page into the target language. The drawback of this is that the outcome of the translation is many times not fluent. The readers of the translation can understand the meaning of the text but something may be lost in the process. For example, phrase identification is a problem that arises when groups of words have a special meaning when they co-occur that is different from the individual meanings of the words [30].

In the context of the Web, when a user has found an interesting Web page, the need to go to a separate translation service may flick him out of flow. The user has to move from the page he is reading into another page where he must insert the URL of the first page and wait for the service to translate it. Then he has to find the sentence he was reading and continue from there.

In the second method of translation, the user has to copy and paste (or write) the translated word into the translation service, which then gives him the translation. Again, the user has to leave from the page he was reading (i.e. he could be distracted) to find out the translation. So our aim is to allow the user to translate single words while staying at the page he is reading, so that distraction would be as small as possible.

Another advantage that we get by translating only those words that the user does not understand is that we also help him learn new language. By translating the whole page he does not see the original text and cannot associate the translated words to the original words.

Of course, this type of solution requires that the user is somewhat familiar with the language he is reading. If the user does not understand the language of the page he is reading, we must use those translation services that translate whole page.

3.2. Richer interaction - annotation and highlighting

Another way of supporting comprehension is to allow deeper interaction with existing knowledge stored in the Web. Following Oinas-Kukkonen’s [23] definition (see also [25]) comprehension can be supported by allowing deeper interactions with the external environment, and better tools for surveying it. The translation could be seen as surveying, whereas annotating and highlighting are interaction with the external knowledge (in this case Wikipedia articles).

Our solution for this is to allow users to post "sticky notes" and to highlight texts in Wikipedia articles. The sticky notes are like post-it notes that stay in the place they are placed. The user can edit the content of the note. Highlighting allows the user to use the mouse to mark text, e.g. to highlight sections that he sees as important.

The type of a sticky note can be a comment, a tag, a question, an answer or an argument. Comments, tags and questions refer to specific Wikipedia pages, whereas answers refer to question-notes and arguments to answer-notes. Comments can be anything from saying ‘I like this’ to ‘I disagree.’ Tags are in essence short comments (usually with only one word), e.g. ‘Design’ or ‘San Francisco’ that categorize the article in some way. Questions are notes that ask something
about the article: ‘Why is this sentence written this way?’ Answers are responses to the questions-notes and arguments-notes relate to answers. Arguments can be against or for the answer it is linked to. Asking questions, answering them and arguing against and for the answers provides valuable knowledge on why certain parts of the article are written the way they are. In fact, typing the nodes as question (Q), answer (A) or argument (R) could help capture some of the knowledge rationale behind the article [25]. This method is called QAR [22], and it is more commonly used to capture design rationale. We believe it can also capture knowledge.

The users can, for example, ask questions about the parts of the article they do not fully understand. As other users answer the question, more rationale about the article is stored as well. Räisänen and Oinas-Kukkonen [25] propose that each produced knowledge object should be stored with the rationale why it is built the way it is, i.e. Knowledge Rationale.

Annotations can also support conceptualization. As conceptualization is a social process, the decisions on how to build an object should be based on reasoning and collective wisdom of the group. An example of such an object could be a Wikipedia article. Storing the rationale of the articles in a structured way allows us to browse the rationale for example. Currently, each Wikipedia article has a discussion page where people can argue about the contents of the article page and these discussions contain much of the rationale behind the articles. Using a more structured way of organizing the discussions would provide a clearer view on why the document is built the way it is.

The highlighting tool is simple. Whenever a user presses the Alt-key and highlights some text with a mouse, the highlighted text will stay highlighted. This allows him to mark important chapters or sentences (as he would with paper and a magic marker) that he finds interesting. Highlighting together with sticky notes allows users to show to which part of the text the note actually refers, e.g. note of “This sentence does not make sense” next to a highlighted sentence allows users to relate the note and the sentence.

Table 1 summarizes the tools. The translation tool helps the user stay in flow. Annotation and highlight tools enable deeper interaction with existing knowledge. By integrating these tools in Wikipedia, we argue that we can support individual users’ comprehension. By allowing users to type the nodes, we can help comprehension even more. The power of annotations and highlights would further increase if we allowed them to be shared with other users. If users could read what others have annotated in the same article we are reading, we could have a better chance of understanding something new. However, it is beyond the scope of this paper to study the affects of shared annotation and highlights.

<table>
<thead>
<tr>
<th>7C tool</th>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation</td>
<td>Translate single words that the user double-clicks</td>
<td>Allow user to better stay in flow by not requiring him to use external translation services</td>
</tr>
<tr>
<td>Annotation</td>
<td>Insert “sticky notes” to Web page. User can edit the notes</td>
<td>Allow user to attach annotations (e.g. comments, questions, answers or arguments) to Web page.</td>
</tr>
<tr>
<td>Highlight</td>
<td>User can highlight texts in the page by selecting text with mouse</td>
<td>Allow user to highlight those text sections he considers important / difficult etc.</td>
</tr>
</tbody>
</table>

In addition, the tools described in Table 1 are by no means an exclusive set of tools supporting comprehension. Utilizing rich links or various types of metadata [4], for example, could also provide some support. For our proof-of-concept implementation, the tools of Table 1 seem feasible and when combined they should offer some support for comprehension and possibly communication, conceptualization and collaboration as well.

4. Proof-of-concept implementation and preliminary evaluation

We implemented the features described in Table 1 on top of Wikipedia. We have argued in previous sections that each of them offers some support for comprehension. The first feature is translation of individual words. The users can double-click on any non-link words and the service will translate that word from English to selected language (only to those languages supported by Google Translator). The service uses AJAX, so the translation requests are made without the need to refresh the whole page. We reused existing Javascript functions found on the Internet and implemented a small script to connect the double-click event to Google’s translation service.

For the second feature (annotations) and third feature (highlight), we found existing scripts that we only slightly modified. We decided to let the users determine the type of annotation (e.g. note, question, answer or argument). The users can also link the notes to each other, e.g. certain answers can be linked to questions, and arguments can be linked to answers (producing Q-A-R as discussed earlier).
We used Java to implement the 7C server-side. The server connects to Wikipedia and includes the wanted functionalities (double-click translation, annotations and highlight) to the page. The server also handles database connections storing each translation and annotation.

Figure 3. Screenshot of the system, including 2 notes, highlight text ("Pori is known, among other things"), and a translation of “foothold” from English to Spanish.

Figure 3 is a screen shot of the system displaying all three new functionalities on top of Wikipedia page about Pori (a small town in Finland).

4.1. Preliminary evaluation

To study the proof-of-concept implementation, we interviewed six users between 27 and 35 years old. Of the participants, two were Japanese, two Finnish, one Chinese and one American. Four were male and two female. They were all experienced with using the Web as they used it daily. They all worked in IT (they were all visiting scholars at a well-known American university) and used Wikipedia (mostly for search) approximately 3-5 times a week. They all considered Wikipedia as excellent source of information.

They were asked to use the system for 15 minutes while the authors observed, after which we conducted interviews lasting around another 15 minutes. We then performed positivistic analysis of the interviews.

We tested the translation service using those participants who did not speak English as a first language. In Wikipedia there are 2.2 million articles written in English (as of 1st of March, 2008). Germany has the second most articles (716,000). As there are three times more English content than any other language in Wikipedia, providing non-native English readers better tools for reading should help users to understand these articles better.

The first result of the study was that the prototype works. It is quite simple to include new JavaScript code in the 7C server and enhance the functionalities of Wikipedia, or any other Web service that uses normal W3C standards. Problems can arise when we try to include new functionalities in a service that is mostly scripting based (e.g. Facebook, www.facebook.com).

4.1.1. Translation. The first impression of the translation was that “It is good and it works.” One of the main benefits seemed to be that it is quick. There is no need to copy and paste: “This is very useful, especially in my situation. I know the grammar but I don't have the vocabulary and I am using a dictionary every time when I read article[s]. I copy and paste, [and it is] hard and it takes [a] long time, for me this is easy to read." One participant also noted that it is “also quite natural that [when] you click a word you don’t understand” the system translates the word.

One participant also indicated that the double-click translation might also support flow: "if you read the article with full concentration, you need to translate the word quickly. If you have to use [a] separate dictionary, you can lose your trail of thoughts." This could indicate that the translation tool causes less distraction, and would not reduce their changes of flow.

There were a few possible problems: “sometimes word translated literally is not what it means in the context [i.e. in the sentence where it is used], but usually you can guess what it means.” Allowing the user to stay in the Wikipedia article while translating the word might indeed help him to understand the article. He can see the sentence and the translation on the same page. He can then translate the whole sentence and continue reading the article.

Sometimes the words might be too difficult so that even a translation does not work. This happened when one of the participants tried the system with medical Wikipedia articles. Maybe the system should show the definition of the word as well or offer a link to some service providing definitions.

4.1.2. Annotations. Participants were used to annotating papers that they read. They also felt having the ability annotate Wikipedia would be useful.

Writing annotations could help readers: “It probably helps when I write it because I have to think what I write.” In a way, the annotations could be used to store new ideas and the articles the annotations are attached could provide the context. This ability to link user annotations to the article probably supports

3 From http://en.wikipedia.org/
comprehension or at least remembering what the annotations were for.

Perhaps more useful than writing would be the ability to read annotations that the user himself or someone else had written. Having someone else’s annotations “would speed up reading the article.” However, if the annotations were from someone whom the reader would not know, it might cause troubles: “Reading someone else’s annotations could feel annoying if they were not relevant to me.” So sharing annotations with colleagues could be more useful than sharing annotations with strangers.

A few participants said that “the ability to share annotations would yield the biggest benefits.” The prototype of the system did not allow users to share their annotations with each other. The ability to put links or pictures into the annotations was seen as great, especially when “the link [...] is relevant to the article.”

4.1.3. Highlight. Highlights helped people to re-read articles: “When I go back, I only read the highlighted parts.” This speeds up remembering articles read previously. It probably also allows users to recall things. Users also indicated that many benefits of the annotations also apply to highlights. For example, the ability to share highlights would be great.

Seeing what others have highlighted would allow participants to understand each other but it “might also distract you,” especially if other users’ highlights would differ greatly from one’s own. This could indirectly help communication and conceptualization if users knew the things they had highlighted differently, e.g. why somebody sees some part of the text important and someone else sees another part.

5. Discussion

The proof-of-concept implementation of the system worked well. It provided the new functionalities to Wikipedia readers. The participants also found them useful. The translation service especially was seen as a great addition to Wikipedia. All of the participants said that annotation and highlight services could help them. It was not always so clear if or how they actually improved comprehension. The participants did feel that they could help but as they did not use the system in real situations, they could not be sure how.

This should be studied by using the system more thoroughly. In addition, the ability to share highlight and annotations should be included in the future versions of the system. It could be that the true support for comprehension comes when we can see what others are doing and see how our own thinking differs from that.

The translation service was seen as the most useful. The reason for this is probably the fact that only one of the participants spoke English as a first language.

The proof-of-concept implementation of the system did not allow users to share their annotations with each other. The ability to put links or pictures into the annotations was seen as great, especially when “the link [...] is relevant to the article.”

Download time is increased (connection goes through a proxy server).

With fast connections, download time is not a big issue. However, if we use the system with a mobile device, the increase in download time could be crucial as download times are already longer with mobile devices. To tackle the first drawback, the annotations and highlights must be attached to certain versions of the article. This way when someone, for example, deletes a sentence we have highlighted, we can still find the highlight in the previous versions of the article. Still, keeping annotations and highlight up to date can be a challenge. For example, do users have to transfer the annotations manually from older versions of the article to the new one?

In addition to these drawbacks using a proxy server might also disturb knowledge creation. Those users who did not use Wikipedia through the proxy would not see the annotations made by other users. However, if the proxy-users would comprehend something they could probably better contribute to Wikipedia articles. However, if some new tool were especially beneficial it could always be implemented to Wikipedia, too.

6. Conclusion

This paper presented a study about how we can support sense-making processes of Web users by using a proxy server. The contribution of this paper is
twofold. Firstly the paper presented a way to use a proxy server to provide new functionalities to Web users. The new functionalities are included into existing Web services in a form of JavaScript code. The proxy server will first download the original HTML code and include the JavaScript into it before sending the file to the browser. This way the user does not need separate browser plug-ins to support the new functionalities. The service works with any device supporting JavaScript without any installs.

Secondly, as a proof-of-concept implementation, we implemented a prototype of the server and three simple tools to support the 7C process called comprehension. To the authors knowledge this is the first study that tries to provide functionalities supporting the 7C model. The tools could be used while reading Wikipedia articles. Analysis of the tools revealed that users did indeed find them helpful in learning (translation, annotation and highlight) and staying in flow (translation). Based on our preliminary evaluation we can not make more thorough analysis. However, the translation tool was seen as excellent addition to all non-native English speakers.

We conclude that they offer some support for comprehension. However, more studies are needed to investigate the system in real-life situations. Also, providing new functionalities to existing Web services (whether by using a plug-ins or proxy servers) seems to offer interesting possibilities in supporting sense-making processes of individual users.

As a future work, the complete set of functionalities that support the whole 7C knowledge creation spiral should be identified and implemented into the 7C Server. Especially the ability to share annotations with other users would be helpful in supporting comprehension. Using QAR we could also support conceptualization. The solution presented in this paper should also be tested more rigorously, e.g. by using an experiment where the prototype system would be compared to normal use of Wikipedia.

Acknowledgements

The author would like to thank the reviewers, members of the Persuasive Technology Lab at Stanford and professor Harri Oinas-Kukkonen for their feedback and comments.

References


Create the Dynamics of Innovation, Oxford University Press.


