Analyzing Client-Side Interactions to Determine Reading Behavior

David Hauger[1] and **Lex Van Velsen**[2]

[1]Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Linz, Austria

[2]Department of Technical and Professional Communication
University of Twente, Enschede, Netherlands

Abstract

Traditional monitoring and user modeling techniques in adaptive hypermedia systems consider pages as atomic units although different sections may refer to different concepts. This has been mainly due to the fact that most user interactions being monitored referred to the request of a new document and there was too little activity information to differentiate between sections of a page. Client-side monitoring can provide additional information on user interactions inside the browser window and may relate them to areas within a document. A user study was carried out to show whether and how this data might be used to identify which parts of a page have been read.

1 Introduction

It has been a widely accepted fact for several years now that "the user can prefer some nodes and links over others and some parts of a page over others" [Brusilovsky, 1996]. Opening a page does not necessarily mean that a user read all its contents. Consequently, adaptive hypermedia systems (AHS) should monitor these nodes separately to tell (a) how much of a page has been read, and (b) what parts of a page have been read or are of particular interest, especially if they concern different topics.

Most AHS try to (partially) meet these demands by monitoring requests to the server, which makes it possible to determine the links a user followed. Nevertheless, concerning text nodes (or links that have not been followed), most AHS treat pages as atomic items. Elaborate algorithms try to add additional information to user models by analyzing requests (e.g. to calculate the estimated "time spent reading" based on the time difference between requests [Farzan and Brusilovsky, 2005]), but there are hardly any attempts to treat different parts of a page separately [Hauger, 2008].

The approach put forward in this paper shows how monitoring user interactions inside the browser could help to overcome these limitations. A user study has been carried out to determine how users interact and how it is possible to determine whether a page has been read.

2 Related Work and State of the Art

Traditional user modeling techniques of AHS log requests of resources on the server and use this information as a basis for modeling. However, most interactions of users do not cause requests to the server (mouse movements, scrolling, etc.) and are therefore not monitored.

Several attempts have been made to use client-side interactions in AHS. Hijikata [Hijikata, 2004] showed that text tracing, link pointing, link clicking and text selection are an indicator for interest. Goecks and Shavlik [Goecks and Shavlik, 2000] defined a "level of activity" based on mouse and scrolling activities monitored via JavaScript. They used it for a neural network inside the browser. Hofmann et al. [Hofmann *et al.*, 2006] sent timestamps of interactions to the server to calculate periods of inactivity.

Claypool et al. [Claypool *et al.*, 2001] developed "The Curious Browser" to log interaction events inside the browser. The results were used to establish a connection between user interaction and the level of interest. Although this solution is effective, it is not ideal because in order to be able to use client-side information in common e-learning situations, additional hardware and software requirements should be avoided and standard technologies should be used for monitoring and transmitting the data. Putzinger [Putzinger, 2007] used mouse and keyboard events on input elements to determine the "locus of attention". This information has been sent to the server to adaptively provide help.

Nevertheless, most systems referred to pages as a whole. Differentiating between sections requires new monitoring techniques. Eye-tracking is one possibility to identify the locus of attention [Conati *et al.*, 2007]. As the applicability of this approach is limited due to additional hardware and software requirements, other solutions using standard technologies need to be found.

Client-side user monitoring as described in [Hauger, 2009] is able to (a) retrieve additional information on user interactions and (b) treat different sections of a page separately. The work described in this paper tries to find out whether and how the information that can be retrieved may be used to determine which parts of a page have been read.

3 Client-Side User Monitoring

In order to overcome the limitations of traditional approaches using server-side logs as the only source of information, the monitoring process itself could be improved by monitoring activities within the browser window [Hauger, 2008]. For this reason a JavaScript library has been developed which monitors these client-side events and maps them to parts of a page [Hauger, 2009].

3.1 Page Fragmentation

Different sections of a page in an AHS might need to be treated separately. As exact mouse positions might be difficult to compare and evaluate, alternative segmentation techniques need to be considered that are robust to changes

in the size and topology of page elements. The library that has been developed supports different approaches to split pages:

- *split by vertical position*: Independently from the actual content a page may be vertically divided into k segments; each one representing $\frac{1}{k}$ of the page. This type of segmentation may be used to calculate how much of a document has been read and it may easily be applied for static and unstructured pages.
- *split by content type*: In oder to identify learning style preferences it is for example possible to monitor images separately to make assumptions on whether users prefer textual or graphical content.
- *split by semantic meta data available*: If there is already semantic meta data available (concepts, keywords, etc.), it is possible to monitor items including such additional information and relate the activity information to this data.
- *split by source*: For "composed" pages with items derived from multiple sources it is possible to automatically link user interactions to the original source of the fragment.
- *split by structural information*: Structural information (if available) like headlines may be used to distinguish between different sections of a page.
- *add custom fragments*: In addition to all mentioned splitting techniques, each HTML element may (even at runtime) be manually defined as a fragment that has to be monitored.

3.2 Monitored Interactions

As JavaScript is used to monitor interactions, the library logs the events already available (including mouse moves, clicks, keyboard activities, scrolling, window resizing and window events like focus and blur) and uses them for further processing (e.g. for mapping positions to fragments). In addition to those predefined JavaScript events, a number of custom events has been created; e.g. to identify text selections (which may be used to identify text tracing) and inactivity (no interactions for a longer period of time), as well as to determine events of a temporal basis. The monitored variables in detail:

- *visible time*: The time a fragment has been visible on the screen. This can be regarded as a requirement for reading. Printing a page, saving it for offline use, etc. may allow to read parts of a text never having been visible within the browser window, but this may be regarded as an exception.
- *mouse over time*: The total time the mouse has been placed above a specified fragment. Some people place their mouse above the text they are currently reading. Therefore this is being monitored to check whether it can really be used as indicator to identify reading.
- *mouse on same y time*: The total time the mouse has been placed within the vertical borders of a fragment. This is similar to the "mouse over time", but ignoring the horizontal position of the mouse. If there is only one ("main") column of text (as in the current experiment), the two variables should be similar. For two or more columns there might be differences, e.g. if a user always places the mouse on the right side of the screen, independent from the horizontal position at which the user is reading. This, however, will be part of future work.
- *number of mouse moves*: Amount of mouse moves taking place above the current fragment. Mouse moves within 500ms have been regarded as a single mouse movement. Passing an item with the mouse in less than half a second has not been counted.
- *number of clicks*: The total amount of clicks performed on the fragment.
- *number of text selections*: Counting how often a user has selected text within a specific fragment.

The main premise of the work described in this paper is that based on these interactions it should be possible to draw additional assumptions on users' reading behavior, interests, etc.

4 User Study

In order to determine how client-side user interactions and reading behavior are related, a user study with 53 volunteers has been carried out. The results of client-side user monitoring should be compared to explicit feedback given by the users. The main goal was the identification of client-side user activities that may be used to identify which parts of a page have been read.

A single page containing a number of news items (20–23) from an Austrian news page (http://oesterreich.orf.at) has been provided. Each item consisted of a thumbnail (width: $\approx 100 - 150px$) on the left side and a headline with $4-6$ lines ($\approx 20-40$ words) of additional text (short summary of an article) next to it. Internally, the page was split automatically in order to monitor each news item separately. As the system should focus on interaction information that cannot be gained through server-side monitoring, links to the extended articles were disabled. The page was updated twice a day to increase the probability users have not read the news before, which should result in higher interest.

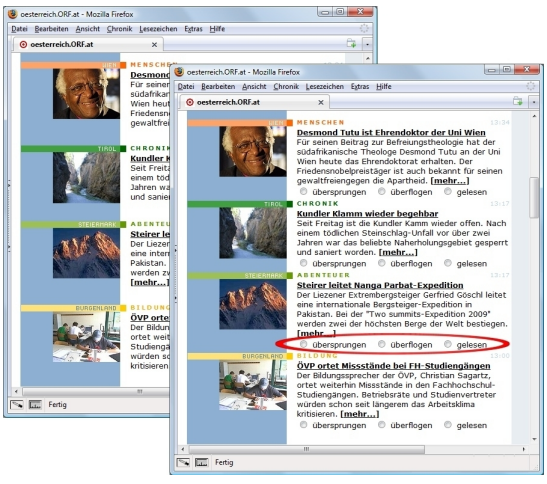

Figure 1: user study: reading and evaluation page

The study itself was entirely anonymous – the participants were not even asked to enter demographic information. Participation was possible via the web. On a first page the experiment was explained and users were instructed to read only whatever interested them, as if they were visiting the news page in a normal context of use.

While they were reading, their interactions within the browser were monitored using the library mentioned in section 3. Information on the absolute location of events were mapped to the news items to be able to compare them later on. The preprocessed events were sent to the server and stored in a database, as well as the values for the variables mentioned in section 3.2 (per user and news item). In addition to this, the total time for a page being requested was recorded, which is the only information that could have been retrieved by server-side monitoring as well.

After reading the users were asked to fill a small questionnaire. For each news item they had to select whether they read this item, glanced at it or skipped it. The page for reading and the feedback form are shown in Figure 1.

It has to be stated that the feedback of the users is subjective and there may be differences in what single users regarded as reading, glancing or skipping.

The evaluation of the results should show how reading and client-side interactions within the browser are correlated. The final goal is the establishment of an algorithm that is able to tell with a satisfactory level of certainty whether a fragment has been read or not. Although the scope of the experiment is not sufficient for getting exact values and parameters for an overall algorithm, this user study should show directions towards creating it.

5 Results

A total of 53 participants completed the questions related to the news items. They provided feedback on 20 to 23 items, with an average of 22.32 items. The participants spent, on average, 2 minutes and 9 seconds reading the news page, with a standard deviation of 2 minutes and 36 seconds.

The items related to user feedback (item skipped, glanced or read) were scored in a dummy variable to enable data analysis. Each feedback option was made into a separate variable with a score of either 0 (item not skipped, not glanced or not read) or 1 (item skipped, glanced or read). The responses were based on the participants' subjective assessment of their own behavior, and thus there might be differences in what users regarded as read, skipped or glanced at. For some users, skipping meant not even scrolling down to the bottom, while others showed quite some interactions with items they marked as skipped. Additionally, "reading" for some users meant "reading carefully", while others marked items as read that were visible for four seconds only. Nevertheless, the results should be able to point out how information on client-side interactions could be used to identify reading.

Table 1 displays the minimum, maximum, mean and standard deviation of all the recorded variables. The table shows that the mouse cursor was, on average, just a few seconds above each item or on the same y-level. The time items were visible on screen differed widely, with an average of $25.15s$ and a standard deviation of $21.14s$.

	N	min	max	mean	SD
mouse time above item	1183	0	63	3.42	5.60
mouse time on same y-level as item	1183	0	64	4.37	5.99
time item is visible in browser window	1183	0	120	25.15	21.14
amount of mouse moves above item	1183	0	30	1.20	2.64
number of mouse clicks on item	1183	0	9	0.11	0.63
number of text selections inside item	1183	0	2	0.01	0.09

Table 1: Descriptives of assessed variables

Finally, more than half of the news items (57%) were skipped, about one quarter was glanced at (23%) and 20%, on average, was read by the participants.

The first step in determining which factors influence item skipping, glancing or reading behavior was to assess the correlations among the variables. The results can be found in Table 2. It shows correlations between all the recorded mouse actions and time measurements and item skipping or reading behavior. Participants' glancing behavior is not correlated with any of the assessed variables. In other words, item glancing behavior cannot be predicted with any of the measured variables. All measured variables have a positive correlation with reading and a negative one with skipping items. This shows that they might be used to determine whether something has been read or skipped.

	A	B	C	D	E	F
item skipped	$-.28*$	$-.29*$	$-.20*$	$-.28*$	$-.17*$	$-.06**$
item glanced at	.03	.04	.02	.03	$-.01$	.01
item read	$.31*$	$.31*$	$.23*$	$.26*$	$.22*$	$.07**$

A) mouse time above item
B) mouse time on same y-level as item
C) time item is visible in browser window
D) amount of mouse moves above item
E) number of mouse clicks on item
F) number of text selections inside item
$*$ significant at 0.01 level (2-tailed)
$**$ significant at 0.05 level (2-tailed)

Table 2: Correlations among variables

However, the direct correlation between the assessed variables and reading behavior is not very strong, which is due to the fact that reading an item does not necessarily result in observable interactions. Nevertheless, the variables may be used as unidirectional indicators for reading behavior. One example is the selection of text. If text has been selected, the item has definitely not been skipped. However, as in 99.7% of the presented items no selection of text took place, the lack of text selections does not give any information at all.

Similarly, all assessed variables have been analyzed to find implications to be derived from the observed data. Table 3 shows how often information on client-side behavior could be retrieved and how measuring interaction times or the occurrence of interactions were related to users' responses on whether an item has been read.

	mouse over	mouse: same y	visible time	mouse moves	mouse clicks	text: select
value > 0	57.3%	74.1%	94.2%	35.6%	5.4%	0.3%

Total percentage of items where client-side data could be retrieved

	mouse over	mouse: same y	visible time	mouse moves	mouse clicks	text: select
read if 0	12.9%	12.7%	1.4%	14.0%	17.1%	19.8%
read if > 0	25.2%	22.5%	21.1%	30.6%	70.3%	75.0%
glanced if 0	21.8%	17.3%	2.9%	24.0%	23.2%	22.9%
glanced if > 0	23.7%	24.9%	24.1%	20.9%	17.2%	25.0%
skipped if 0	65.3%	69.9%	95.7%	61.9%	59.7%	57.3%
skipped if > 0	51.0%	52.7%	54.8%	48.5%	12.5%	0.0%

Total percentage of items having been read / glanced at / skipped depending on whether client-side monitoring returned a value > 0

Table 3: Occurrence of interactions

Generally it may be said that the observation of client-side interactions at least doubles the probability that an item has been read. 80% of the items with no monitored interactions or an interaction time $< 0.5s$ (rounded to 0) have not been read and most of them have been skipped.

However, half of the items where interaction times have been measured or mouse moves have been monitored have been skipped as well. Therefore, the second part of the current section consists of a closer analysis of the assessed variables and should show how higher activity values correspond to a higher probability that something has been read.

Analyzing Mouse Over Time 52.8% of all items that have been read had a mouse over time of more than 3 seconds. Items with a total mouse over time of more than 8 seconds (12.9% of all cases) have a 0.50 probability of having been read and a 0.77 probability that the item has not been skipped.

As shown in Figure 2 a higher mouse over time goes along with a higher probability that an item has been read.

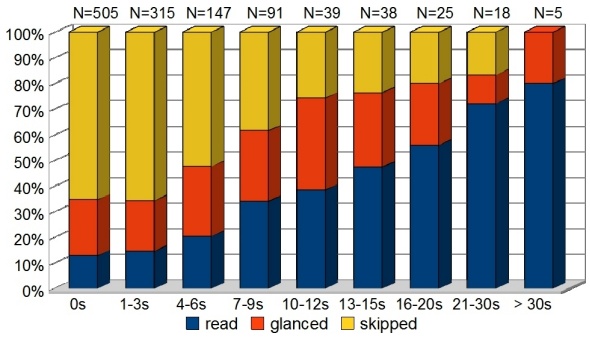

Figure 2: correlation of mouse over time and reading

Analyzing Vertical Mouse Position The time the mouse cursor has been placed at a vertical position within the borders of the news item is similar to the mouse over time and therefore the results as well (see Figure 3).

Compared to the mouse over time, the probability of the mouse never having been on the same y position as an item is lower (of course; as hovering an item implies that the mouse is also on the same vertical position). Comparing $mouse\ over\ < 1s$ and $same\ y\ < 2s$ both cover more or less the same test cases. Generally, the small differences between mouse over time and the vertical mouse position lead to the assumption that users that placed their mouse cursor inside the page tended to place it above the news items. However, this effect might have been different if more items had been placed next to each other on the same vertical position. Further work needs to be done to tell whether the y-position of the mouse or exact hovering is more significant in a different context.

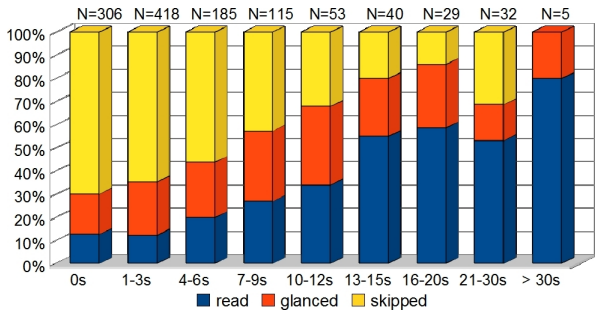

Figure 3: correlation of vertical mouse position and reading

Analyzing Visibility Time 81.1% of the items that have been visible for less than 5 seconds (13.4% of all cases) have been marked as skipped. The probability that an item has not been read (i.e. skipped or glanced at) if it has been visible for less than 5 seconds is 0.93. Only 1% of all items have been marked as read and were visible for less than 5 seconds (no surprise as items have to be visible to be read).

Other than this the visibility time does not provide any relevant information. As shown in Figure 4 the probability that an item has been read increases only slightly with a higher visibility time. This increase is definitely not sufficient for drawing further conclusions.

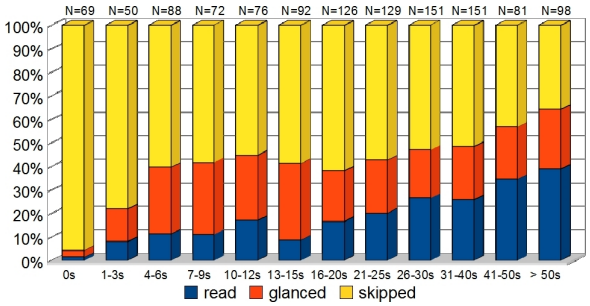

Figure 4: correlation of visibility time and reading

Nevertheless, taking into account the screen size and consequently the number of items displayed at the same time, it might have been possible to derive a weighted metric combining visible time with screen size that might have been more informative the the time by itself. Moreover, the relative position of the item within the screen might give additional information if it can be found that for instance users tend to read text that is displayed in the center of the screen. These two aspects will be considered in future experiments.

Analyzing Mouse Moves 91.1% of the skipped items had only 2 or less registered mouse moves and 98.5% of all skipped items had 5 or less registered mouse moves. Moreover, 54.8% of the items that have been read had at least one registered mouse move. No registration of mouse moves is a good indicator for having been skipped and a high amount leads to the assumption something has been read. Only 0.8% of all monitored news items have been marked as skipped despite having more than 5 mouse moves.

Detailed information can be found in Figure 5.

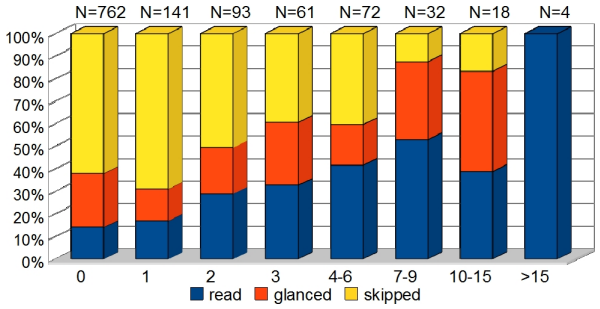

Figure 5: correlation of the number of mouse moves within a news item and reading

Analyzing Click Events Only in 5.4% of the cases clicks have been registered. However, 70.3% of them have been marked as read. Only 0.7% of the test cases showed clicks despite having been marked as skipped. This shows that although clicks do not occur frequently, they are a strong indicator that something has been read.

Analyzing Select Events As already mentioned, text selections occur even less frequently than click events (only 0.3% of all test cases). Nevertheless, text selections are the strongest indicator for reading and none of the items where text selections took place has been marked as skipped.

6 Towards an Algorithm

The results of this user study show that information on client-side user interaction is definitely suited for determining which parts of a page have been read or skipped. However, the observed variables provide different types of information. In some cases (especially interaction times) the lack of information is an indicator for skipping, in others (especially interactions) there is little probability that something can be observed, but if interactions have been monitored they serve as an indicator for reading. The visibility time works very well for identifying skipped items, but high visibility times do not really increase the probability that an item has been read (although, as discussed in the previous section, this effect might be reduced by considering the size of the browser window and the relative position of the items within). Clicks and text selections help to identify read items, but do not work for identifying skipped items.

Based on this information it may be said that when looking for an algorithm returning a probability that an item has been read, linear algorithms are definitely not the best choice. Linear models can still be informative though in terms of the viability of using specific factors and indices into the algorithm. To explore this premise we started our analysis using the following composite metric (which was only intended to give a quick impression on whether the variables might be suited to analyze reading behavior): $(1 + mouse\ \ over\ \ time) * (visible\ \ time) * (1 + mouse\ \ moves) * (1 + clicks)$. The value for the mouse on the same y position is part of the mouse over time and text selections hardly ever occurred, so these two variables have been left out. If the visible time is 0 the item can be regarded as skipped, but for all other variables even a value of 0 could mean it has been read – depending on the other variables. Therefore those variables have been added with 1. Using this simple algorithm 68% of all read items had a value above 108 and 68% of the skipped items had a value below 108. These values are of course specific to the experimental data at hand, and would in all likelihood differ significantly in other cases. However, the results do indicate that these factors do indeed have discriminatory capacity, and, possibly in an appropriately weighted form, can indeed be used as the basis for an algorithmic approach.

Having established at least some of the factors that an algorithm could incorporate, we turned our attention to the nature of the algorithm that could be used to identify page segments that had been read. The primary design requirements were:

- *real-time*: The algorithm should be fast enough to provide just-in-time information for several users while continuously monitoring user interactions.
- *predictive*: The algorithm should be able to handle continuously updating information without relying on an analysis after a user left the page.
- *white box*: The algorithm should consist of semantically understandable parts in order to be able to extend the algorithm and add factors later (or set different factors for different contexts).

Based on these requirements we decided to direct our attention to rule-based approaches, Bayesian networks and decision trees, as well as hybrid approaches comprising the above and potentially complementary ones as well.

In order to find a more appropriate algorithm the open source data mining software Weka [Witten and Frank, 2005] has been used applying different machine learning algorithms for classification. This software may be used to automatically generate models for classification algorithms by using a subset of the raw data. The other part of the raw data is used to evaluate the deriving models in order to determine how well data sets can be classified. The exact way in which the data set is split affects the performance of the algorithms. Therefore, a 100-fold cross validation has been used, i.e. the data set is randomly split 100 times and the result refers to the average value for all test cases.

The data from the user study was used to get an algorithm for predicting whether an item has been read fully or not. For the purposes of the analysis presented herein, "glanced at" and "skipped" have been combined to a single group. For each of these two classes the number of correctly classified items has been calculated as well as the precision (the probability that the item has really been read / not read, if the algorithm classified it this way).

Most algorithms had an overall precision of $\approx\ 80\%$. They showed good results especially for identifying items that have not been read. More than 95% of the "not read" items have been classified correctly (with an overall precision above 80%). However, the algorithms were less successful in identifying items that have been read. The total precision for items classified as "read" was $\approx 60\% - 70\%$ and only $15\% - 30\%$ of the read items have been correctly classified as read.

As an example three different classifiers will be discussed in details. They have all been tested using a 100-fold cross validation in Weka. The results are listed in Table 4.

One simple approach for classification is a rule based algorithm:

```
read =
    (mouse_moves >= 2 && same_y >= 13) ||
    (visible_time >= 17 && clicks >= 1)
```

The highest precision for read items was reached by the Bayesian Network shown in Figure 6. On the other hand it only classified 16% of the read items correctly.

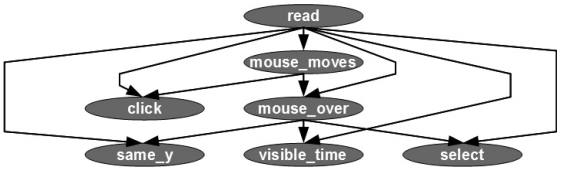

Figure 6: model for setting up a Bayesian Network

The highest average precision and the highest percentage of correctly classified read items was reached by the decision tree shown in Figure 7.

The above results clearly indicate that more work needs to be expended in devising a generic algorithm, as well as in understanding how different interaction- and context- characteristics influence the significance of the identified factors (and of how to incorporate these varying levels of significance in the algorithm itself). Nevertheless, the results seem promising in terms of being able to use client-side interactions to make assumptions on reading behavior.

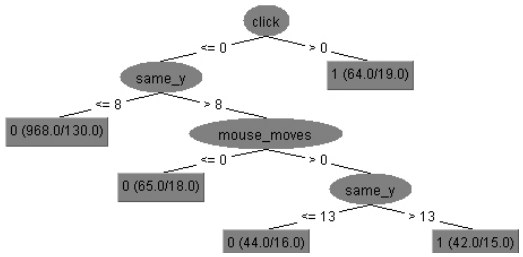

Figure 7: decision tree

	rule-based	Bayesian Net	tree-based
correctly classified read	25.8%	16.1%	30.5%
correctly classified not read	96.0%	98.4%	95.8%
correctly classified (average)	82.0%	82.0%	82.8%
precision read	61.6%	71.7%	64.3%
precision not read	83.9%	82.5%	84.7%
precision (average)	79.4%	80.3%	80.6%

Table 4: performance of different classifiers in Weka

7 Ongoing Work and Future Perspective

As a next step an experiment comparing the current work with eye-tracking will be performed. This should show how mouse positions are related to the locus of attention and whether client-side monitoring could provide parts of the information available through eye-tracking. Moreover, it should show whether users have preferences concerning the relative position of what they are currently reading (i.e. whether they focus more on elements that are displayed at the center of the screen). If this is found to be relevant, the library will be extended to monitor the relative position of page fragments on the screen and get more fine-grained information on the visibility time.

Furthermore, the library will be extended to get more fine-grained information on user behavior. This includes monitoring the scrolling speed and the size of the browser window. As the number of items visible in parallel depends on the window size (e.g.: big screen vs. mobile device), this may help to better use the visible time (fewer items on a screen increase the probability for a single one being read). Moreover, client-side monitoring should be used in different contexts; the way of reading a news page may be different from reading text in an e-learning course.

As a strong correlation between "mouse over" and "mouse on same y" has been found, it has to be tested whether this is also true if several items are placed at the same vertical position or whether it is possible to ignore information on the horizontal position of the mouse.

Another important factor for future research is the length of the text within a single item. This length is important to estimate the time required for reading. For the current experiment only elements of almost the same structure and length have been used to reduce the complexity of the test. The average reading speed as well as the estimated personal reading speed in relation to the length of the text comprised by a monitored page fragment are additional factors that we believe may need to be considered as factors and incorporated into the algorithm. Based on the estimated required reading time the visibility times and interaction times could possibly provide additional information.

The main work however lies in the further development of an algorithm (or a number of algorithms that work for different contexts). The results of this work should be integrated into a version of AHA [De Bra and Ruiter, 2001] running in the open source learning platform Sakai [Sakai, 2009] to provide the findings of ongoing research for a larger audience and help to improve existing AHS.

Acknowledgments

Part of the work reported in this paper was funded by the "Adaptive Support for Collaborative E-Learning" (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

[Brusilovsky, 1996] Peter Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[Claypool *et al.*, 2001] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Intelligent User Interfaces*, pages 33–40, 2001.

[Conati *et al.*, 2007] Cristina Conati, Christina Merten, Saleema Amershi, and Kasia Muldner. Using Eye-tracking Data for High-Level User Modeling in Adaptive Interfaces. In *Proc. of The AAAI Conference on Artificial Intelligence (AAAI 07)*, pages 1614–1617, 2007.

[De Bra and Ruiter, 2001] P. De Bra and J.-P. Ruiter. AHA! Adaptive Hypermedia for All. In *Proceedings of the WebNet Conference*, pages 262–268, October 2001.

[Farzan and Brusilovsky, 2005] R. Farzan and P. Brusilovsky. Social Navigation Support in E-Learning: What are the Real Footprints? In *Third Workshop on Intelligent Techniques for Web Personalization (ITWP '05). At 19th Int. Joint Conf. on Artificial Intelligence*, 2005.

[Goecks and Shavlik, 2000] Jeremy Goecks and Jude W. Shavlik. Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. In *International Conference on Intelligent User Interfaces - Proceedings of the 5th international conference on Intelligent user interfaces*, pages 129–132, 2000.

[Hauger, 2008] David Hauger. Fine-grained user models by means of asynchronous web technologies. In *ABIS 2008 - Adaptivity and User Modeling in Interactive Systems*, pages 17–19, Würzburg, Germany, 2008.

[Hauger, 2009] David Hauger. Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems. In *UMAP 09: Adaptation and Personalization for Web 2.0*, 2009.

[Hijikata, 2004] Yoshinori Hijikata. Implicit user profiling for on demand relevance feedback. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 198–205, 2004.

[Hofmann *et al.*, 2006] K. Hofmann, C. Reed, and H. Holz. Unobtrusive Data Collection for Web-Based Social Navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.

[Putzinger, 2007] A. Putzinger. Towards Asynchronous Adaptive Hypermedia: An Unobtrusive Generic Help System. In A. Hinneburg, editor, *LWA 2007: Lernen - Wissen - Adaption*, pages 383–388, 2007.

[Sakai, 2009] Sakai. collaboration and learning for educators, by educators, free and open source. http://sakaiproject.org, 2009.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data mining: practical machine learning tools and techniques*. San Francisco, 2nd edition, 2005.