Using Browser Interaction Data to Determine Page Reading Behavior

David Hauger[1] and Alexandros Paramythis[1] and Stephan Weibelzahl[2]

[1] Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Altenbergerstrasse 69, A-4040, Linz, Austria
`{hauger,alpar}@fim.uni-linz.ac.at`
[2] School of Computing, National College of Ireland,
Mayor Street, IFSC, Dublin 1, Ireland
`sweibelzahl@ncirl.ie`

Abstract. The main source of information in most adaptive hypermedia systems are server monitored events such as page visits and link selections. One drawback of this approach is that pages are treated as "monolithic" entities, since the system cannot determine what portions may have drawn the user's attention. Departing from this model, the work described here demonstrates that client-side monitoring and interpretation of users' interactive behavior (such as mouse moves, clicks and scrolling) allows for detailed and significantly accurate predictions on what sections of a page have been looked at. More specifically, this paper provides a detailed description of an algorithm developed to predict which paragraphs of text in a hypertext document have been read, and to which extent. It also describes the user study, involving eye-tracking for baseline comparison, that served as the basis for the algorithm.

Keywords: interaction monitoring, modeling algorithm, eye-tracking, empirical study

1 Introduction

Server-side data collection is the most common source of information in adaptive hypermedia systems (AHS). The main drawback of relying solely on request-based information is that requesting a page is not necessarily equivalent to reading everything that is presented on this page. Therefore, more recent systems also utilize time between requests [10] and / or semantic information embedded in the requests to improve on derived assumptions.

Client-side user behavior has long been identified as a potential additional source of information, but due to technical limitations it was difficult to access. Early attempts used custom browsers [15] or browser plugins [5] to enable client-side monitoring. With JavaScript now established as a commonly supported in-browser technology, more recent systems used this to reliably capture mouse and keyboard events on the client side. For instance, mouse movements have been used to identify learning types [14] [1] or as input for a neural network to calculate a “level of activity” for a page [5]. Within “The Curious Browser” [4], Claypool et al. found that the amount of scrolling and the time spent on a page may be used to identify interests, and that the absence of individual scrolling actions or mouse clicks helped to identify the least interesting pages.

More recent work by the authors has examined the premise that increased granularity of information on a user’s client-side activities might help not only in making inferences on a page as a whole, but also in splitting pages and treating the resulting fragments separately [6]. A first user study conducted to this end [8] addressed the question of whether browser events (resulting from user interaction) are generally suited for differentiating the user’s reading behavior in distinct parts of a page. In this study, users were asked to read a single news page including several short articles. Their behavior was recorded with a purposely developed JavaScript monitoring library. Results showed that some events (especially clicks and text selections) are well suited to identifying whether the related text fragment has been read, although the lack of explicit interactions reduces the accuracy of assumptions. For instance, a selection in a paragraph is a strong indicator that the paragraph has been read; however, the more common case of “no selections” provides hardly any information at all. On the other hand, in some cases it is trivial to determine that something has not been read (e.g., the user never scrolled to a part of the page), but increasing times of visibility of text fragments –above the estimated time required for reading– by themselves, only slightly change the probabilities that something has been read. Using the amount of time the mouse pointer has hovered over articles was found to give some additional information on whether some paragraph might have been read.

Following these first encouraging results, we went on to examine whether it is possible to increase the accuracy of predicting what a user has read while at a page, by identifying and interpreting specific patterns in the user’s interactive behavior. A primary objective in this second study has been to perform the monitoring unobtrusively, allowing the user to behave naturally (in contrast to approaches that enforce specific user behavior, such as blurring the screen and highlighting only the area around the mouse pointer, to force the user to “read with the mouse” [16]). Our overall goal was to find out how the observation of users’ normal and unencumbered mouse and keyboard behavior could be related to what users are currently reading. Correlations of mouse and eye positions in situations with many “required” mouse interactions like web browsing [3] and within search interfaces [2] [13] have already been measured. The same is true for repeated visitation patterns [12] [11]. Our own results [7] showed a potential for learning environments as well, and we have been able to prove a number of hypotheses based on interaction patterns that were then used as a basis for an algorithm that associates such patterns with the users’ reading behavior.

This paper reports on the aforementioned second study, along with the hypotheses tested and the results obtained; the prediction algorithm developed on the basis of these results; and the performance of the algorithm. The paper is concluded with a discussion of the algorithm’s strengths and limitations, and an outlook of our ongoing and forthcoming work in this area.

2 Method and Experimental Setup

2.1 Hypotheses

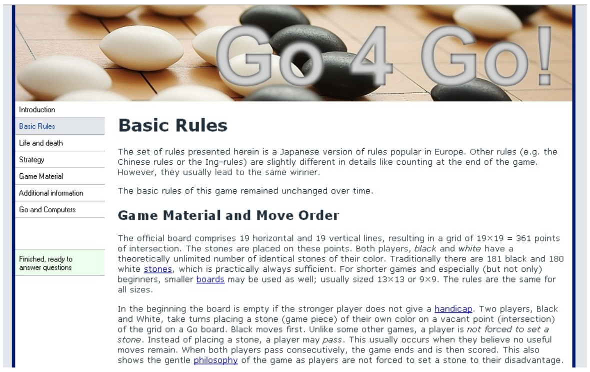
As a first step in developing an algorithm for predicting what users read on a page, we examined a number of hypotheses that attempted to relate specific interaction patterns with reading behavior. These included (imprecise terms used are defined in section 2.3, after discussing the experimental setup):

H1: For pages where users moved their mouse frequently: (a) there is strong correlation between the positions of the mouse pointer and the users' gaze; (b) there is strong correlation between the positions of the mouse pointer and the users' gaze, while the users are moving the mouse; (c) the paragraph under the mouse pointer tends to be the same as the one being read; (d) the paragraph under the mouse pointer tends to be the same as the one being read, while the users are moving the mouse; and, (e) if the frequent movement is vertical, the mouse pointer's position is strongly correlated with the position of the users' gaze.
H2: An indicator for the user's current reading position is: (a) moving the mouse; (b) clicking on text; and, (c) selecting text.
H3: For users using their mouse frequently, the mouse position may be used to identify the relative position within the screen (e.g., top, middle, bottom) they most likely pay attention to (using the mouse position as indicator)
H4: After scrolling up, users are more likely to focus their attention on the items that became visible and were not visible before.
H5: Users scrolling down at small increments, tend to read mostly within a relative area of the screen (top / center / bottom).

2.2 Experiment Setup

To test these hypotheses, we designed a study that allowed us to compare users' reading behavior when encountering different types of text, to their interactive behavior while reading these texts in a browser. Reading behavior was determined through eye-tracking (described in more detail later), whereas interactive behavior was recorded through the purposely developed JavaScript library. The study involved a total of 13 participants (6 male, 7 female) in Ireland. Participants were given five tasks to perform, each based on a different type of text typically encountered online (one main task with seven pages of instructions and information for a board game, and four additional single-page tasks: a multiple choice questionnaire on the board game, a set of search results, a health-related article, and a set of news items). User interaction with the texts, as well as with all other study-related materials and instructions, was through a browser.

The main task involved the users learning about, and answering questions regarding, the game of "Go". The seven different pages comprised text (ca. 7010 words), graphics (11) and pictures (5). A typical page is shown in Fig. 1. Participants were free to navigate between pages, using the navigation bar or hyperlinks



Basic Rules

The set of rules presented herein is a Japanese version of rules popular in Europe. Other rules (e.g. the Chinese rules or the Ing-rules) are slightly different in details like counting at the end of the game. However, they usually lead to the same winner.

The basic rules of this game remained unchanged over time.

Game Material and Move Order

The official board comprises 19 horizontal and 19 vertical lines, resulting in a grid of 19×19 = 361 points of intersection. The stones are placed on these points. Both players, *black* and *white* have a theoretically unlimited number of identical stones of their color. Traditionally there are 181 black and 180 white stones, which is practically always sufficient. For shorter games and especially (but not only) beginners, smaller boards may be used as well; usually sized 13×13 or 9×9. The rules are the same for all sizes.

In the beginning the board is empty if the stronger player does not give a handicap. Two players, Black and White, take turns placing a stone (game piece) of their own color on a vacant point (intersection) of the grid on a Go board. Black moves first. Unlike some other games, a player is *not forced to set a stone*. Instead of placing a stone, a player may *pass*. This usually occurs when they believe no useful moves remain. When both players pass consecutively, the game ends and is then scored. This also shows the gentle philosophy of the game as players are not forced to set a stone to their disadvantage.

Fig. 1. Excerpt of page on basic game rules for the game of “Go”

in the text. In order to motivate participants to read this text carefully, they were told in advance that they would have to sit a quiz on the content afterwards. Web pages were presented through the Internet Explorer browser (in “kiosk” mode with only a minimal set of browser controls visible in a toolbar at the top of the page). Descriptions of tasks and instructions were also included in web pages. All material was presented through a TFT screen, running at a resolution of 1280x1024 (1280x996 effective, excluding browser navigation bar). Gaze position was determined with an SMI RED4 remote eye-tracker. Gaze data, as well as data about web pages presented, was collected through the so called Experiment Center Suite software.

2.3 Evaluation of the Hypotheses

In the briefly described first step of our analysis [7], we tested the hypotheses to identify interaction patterns suitable for developing an algorithm. In total 112 page requests were recorded, with a page being visited for 2 to 1096 seconds with a mean of 122 ($\sigma = 116s$). On average, each user spent 17.54 minutes on the information on the game of Go. Before proceeding to discuss the results obtained, we need to more precisely define some of the terms used in the hypotheses.

To start with, several of the hypotheses refer to “mouse moves”. For this study we defined a “mouse move” to be any set of changes in the mouse pointer’s position, preceded and followed by at least one second of idle time. This definition was derived empirically and subsequently verified on the basis of the collected data, coupled with direct observation of recorded video of the users’ sessions. Moves outside the viewing area (e.g., users dragging the scrollbar) were filtered out. “Frequency” of mouse moves on a per page basis was defined to be the ratio of time during which the mouse pointer moved, vs. the total time spent on the page (including idle time); e.g., a frequency of 25% indicates that the pointer moved for a quarter of the time a page was viewed. With these definitions at hand, we can now proceed to discuss the findings.

H1: *Are mouse pointer position and gaze position correlated?* Analyzing across all users and including idle times, we found weak correlations both hori-

Table 1. Correlations of pixel positions of mouse cursor and eye gaze, depending on the frequency of mouse usage

	Frequency of mouse moves	**Correlation**		**Regression model**		
		$r_{eye_{vs}mouse}$	**N**	**constant**	**weight**	**sig.**
vertical	baseline	.250	89739	345.133	.228	.000*
	all; weighted by frequency	.528	89739	233.106	.494	.000*
	$frequency > 25\%$	.608	39134	211.082	.567	.000*
	$frequency > 50\%$	.658	21906	173.328	.613	.000*
	$frequency > 75\%$	.746	16360	165.461	.666	.000*
horizontal	baseline	.101	89739	577.499	.75	.000*
	all; weighted by frequency	.284	89739	461.427	.248	.000*
	$frequency > 25\%$	.393	39134	385.193	.386	.000*
	$frequency > 50\%$	.493	21906	241.912	.604	.000*
	$frequency > 75\%$	.560	16360	188.254	.727	.000*

zontally and vertically (see baseline in Table 1). However, these correlations are too weak to make reliable predictions on what has been read. We thus explored how predictions may be improved based on the frequency of movements.

H1.a: *Is it possible to improve prediction of gaze by considering the frequency of mouse usage on a page?* We found that, the higher the percentage of mouse movements, the lower the distance between mouse and gaze positions – see Table 1. Including mouse frequency as a weight (see rows "all" in Table 1 in comparison to baseline) raises the correlation significantly. When events are filtered by the level of frequency of mouse movements (e.g., greater than 25%), the correlation increases even further. As one might expect, the more restrictive the filter, the higher the correlation. In accordance with the baseline, the correlations in the vertical direction are higher than in the horizontal. In summary, predictions of the gaze position will be more accurate for users who use their mouse frequently on a page, than for those using the mouse less often.

H1.b: *Are H1.a predictions better while the mouse is in motion?* To analyze this hypothesis, we identified those events where the mouse was actually in motion. In comparison to the previous model, correlations increase yet again – see Table 2. In line with the results above, correlations also increase with more restrictive frequency filters. This suggests that prediction of the gaze position will be more accurate while the mouse is in motion.

H1.c: *Do gaze and mouse point at the same paragraph on the screen?* In general, the element pointed at with the mouse coincides with the paragraph

Table 2. Correlations of pixel positions of mouse cursor and eye gaze while the mouse is being moved, depending on the frequency of mouse usage

	Frequency of mouse moves	**Correlation**		**Regression model**		
		$r_{eye_{vs}mouse}$	**N**	**constant**	**weight**	**sig.**
vertical	baseline	.250	89739	345.133	.228	.000*
	all; weighted by frequency	.752	59857	137.826	.723	.000*
	$frequency > 25\%$	.746	36202	142.850	.725	.000*
	$frequency > 50\%$	.751	21401	139.159	.701	.000*
	$frequency > 75\%$	.777	21270	153.598	.696	.000*
horizontal	baseline	.101	89739	577.499	.75	.000*
	all; weighted by frequency	.521	59857	252.650	.596	.000*
	$frequency > 25\%$	.513	36202	265.610	.579	.000*
	$frequency > 50\%$	.551	21401	202.246	.682	.000*
	$frequency > 75\%$	.580	21270	170.052	.764	.000*

Table 3. Frequency of element hovered by mouse matches element currently being looked at based on frequency of mouse moves – overall and while mouse being moved

Frequency level of mouse moves	overall			filter: while mouse moved		
	Frequency: match	Standard Deviation	N	Frequency: match	Standard Deviation	N
0%-25%	21,00%	.404	24331	59,00%	.493	12555
25%-50%	51,00%	.500	9461	60,00%	.490	8512
50%-75%	70,00%	.458	3478	82,00%	.387	3314
75%-100%	72,00%	.451	13921	72,00%	.449	13844
Total	26,00%	.441	51191	65,00%	.477	38225

looked at in 26% of the cases. When limiting the analysis to cases where people use the mouse a lot, this rises up to 72% (see Table 3). Again, more restrictive frequency filters increase the likelihood that the paragraphs are the same.

H1.d: *Are H1.c predictions better while the mouse is in motion?* In line with H1.b results, predicting which paragraph has been looked at is easier when the mouse is in motion. In particular, for users that do not use the mouse a lot (frequency level $0\% - 25\%$), prediction increases strongly (compare columns "overall" and "filter: while mouse moved" in Table 3).

H1.e: *If vertical predictions are better, should we select vertical moves rather than just frequent moves in any direction?* While in all cases the predictions were better than the baseline and followed the same trends as the previous results (e.g., in motion better than not in motion), frequency of vertical movements did not improve prediction over the levels observed for general frequency of mouse movements (e.g., $r = .397$ for vertical moves vs. $r = .528$ for general moves).

H2.a-c: *When the mouse is actively used, users are likely to look at the region the mouse is positioned.* The mean distance of mouse and eye position reduces to less than 50% when users are clicking, selecting text, or when the mouse is moving (see Tables 4 and 5). Again, the horizontal correlation is lower than the vertical. This is in particular true for text selection activities, where users seem to read left to right, but keep the mouse at one end of the selected text. However, this improvement of prediction comes at the expense of very limited coverage. In short, when mouse actions occur, predictions will be good, but clicks, selections and movements occur only for a fraction of the total observation time.

H3, H4, H5: While we could not establish statistically significant support for these hypotheses, this may partly be due to the type of task we set. For instance, we observed only a limited number of scrolling-up events (H4) and very few instances of small increment scrolling (H5). The analysis for relative areas on the screen (e.g., top, middle, bottom) seems to be invalidated by the fact

Table 4. Mean distances in pixels between mouse cursor and eye gaze for selected types of interactions

		N	mean distance	Std. Error	F	Sig
click	no	86838	383.9	.746	796.5	.000*
	yes	2901	163.4	7.77		
select	no	89706	382.0	.746	26.31	.000*
	yes	33	136.3	47.8		
in move	no	29882	404.7	.768	7063.5	.000*
	yes	59857	222.1	2.033		

Table 5. Regression models for user interactions

	event	Correlation		Regression model		
	filter	$r_{eye_{vs}mouse}$	**N**	**constant**	**weight**	**sig.**
vertical	baseline	.250	89739	345.133	.228	.000*
	click	.873	2901	83.245	.820	.000*
	select	.986	33	64.388	.826	.000*
	in move	.672	59857	161.966	.659	.000*
horizontal	baseline	.101	89739	577.499	.075	.000*
	click	.808	2901	98.057	.774	.000*
	select	.494	33	334.191	.579	.004*
	in move	.436	59857	330.684	.435	.000*

that almost everybody gazed at the middle part of the screen for the majority of time (H3) (see Fig. 2); this finding (i.e. users tend to scroll down just for a few lines while they are reading to keep the currently read item at the center of the screen), however, is in itself also quite useful in establishing a prediction algorithm as we see later.

3 From Hypotheses to Algorithm

3.1 General Structure of the Algorithm

Based on the findings outlined in the previous section, an algorithm was developed to calculate the extent to which paragraphs (or more generally: text fragments) of a page have been read. The main premise of the algorithm is the "splitting" of the time spent reading between the items visible at that time. Therefore each page view is split into "scroll windows", i.e. the time window where the visible items and their relative position on the screen remain constant (identified as the time spans between load, scroll or resize events).

For each such scroll window, the algorithm first calculates the "estimated time spent reading" (T_E). This is based on the measured "available" duration of the scroll window (T_A), but also takes into consideration interaction data that may provide additional information. For instance, if users usually exhibit considerable mouse activity, and then suddenly stop interacting, it is possible that

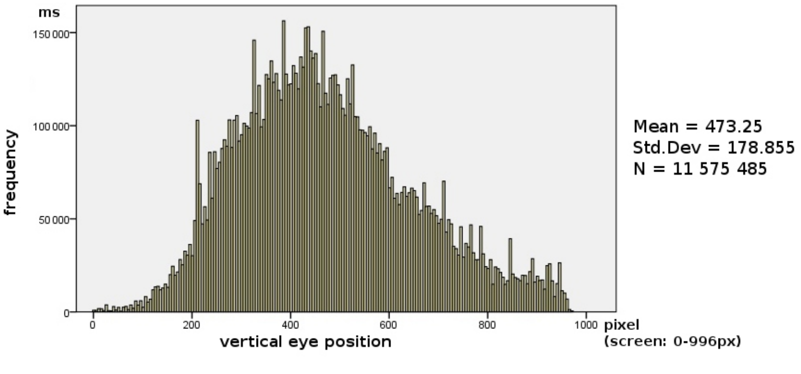


Fig. 2. Histogram of vertical eye position within the screen

they have not been continuously reading. The motivation behind the introduction of T_E is the derivation of a time measure that potentially more accurately represents the *real* time that users spent reading during a scroll window.

To get the time spent reading for each visible fragment ($\overrightarrow{TSR}$), T_E is split among the visible page fragments by multiplying it with a vector defining the percentage of time that should be assigned to each fragment. This vector is a weighted average of a number of normalized distributions of time ($\overrightarrow{TD_N}$) created by different modifier functions (hereforth referred to as "modifiers"), each focusing on a different aspect, for instance, the number of words in the different paragraphs, the number of interactions, the relative position of the paragraph within the screen, etc (see section 3.2). Each modifier receives as input the interaction data of the scroll window, and provides the following output values:

- w_{INT}: The internal weight of the modifier, which provides an indication of the modifier's relative significance for a given scroll window. For instance, a modifier based on text selections would return a w_{INT} of zero if no selections were made during a scroll window, as it can not provide any predictions.
- $\overrightarrow{TD_N}$: The modifier's normalized distribution of time over the text fragments (partially or entirely) visible during the scroll window. The result is a vector of weights for each such fragment.
- $T_{\%}$: The modifier's estimated percentage of the total available time (T_A) the user spent reading in a scroll window.
- w_{TIME}: A weight to be used in association with $T_{\%}$. Similar to the internal weight for the time distribution this value is the internal weight for the estimation on the percentage of time a user spent reading.

Further to the above, each modifier has an "external weight" (w_{EXT}), which denotes the relative significance of a modifier over others. A modifier based on text selections for instance provides stronger indicators of reading behavior than one based on fragment visibility.

Based on the above, T_E is defined as follows:

$$T_E = T_A \cdot \frac{\sum_{i=1}^{N_M} w_{EXT_i} \cdot T_{\%_i} \cdot w_{TIME_i}}{\sum_{i=1}^{N_M} w_{EXT_i} \cdot w_{TIME_i}}$$

where N_M is the total number of modifiers applied. The final algorithm can then be described as follows:

$$\overrightarrow{TSR} = T_E \cdot \frac{\sum_{i=1}^{N_M} w_{EXT_i} \cdot w_{INT_i} \cdot \overrightarrow{TD_{N_i}}}{\sum_{i=1}^{N_M} w_{EXT_i} \cdot w_{INT_i}}$$

where $\overrightarrow{TSR}$ is the column vector containing the calculated time spent reading for each visible text fragment.

The external weight of the modifiers is the only part of the algorithm that is not directly derived from user interaction. Our first experiments had already shown which interactions should get stronger weights (e.g., text selections). Combining these results with the more recent findings (specifically with the identified strength of the correlation for confirmed hypotheses), allowed us to arrive at a set of weights that were used to derive the results described in section 4. Note that we do not consider these weights to be final or absolute. We expect that adjustments may be needed to cater for specific characteristics of the reading context. Nevertheless, there are two points that merit attention: (a) the derived weights appear to have only little sensitivity over the type of text being read; and, (b) even in a "worst case" scenario with all weights set to 1 (equivalent to no knowledge of the expressiveness of different interaction patterns) the algorithm still classified 73.3% of the paragraphs correctly (92.9% with a maximum error of 1 level); please refer to Section 4 for a discussion of these percentages.

3.2 The Weight Modifiers

Currently there are six implemented modifiers focusing on different aspects of the interaction data. Due to lack of space we provide here only a brief outline of each modifier, along with its base hypotheses and external weight:

M_{Select}: This modifier is based on text tracing, i.e., selecting portions of text while reading [9], which is a strong indicator of current reading. In all our experiments it was both the strongest indicator, but also the least frequent type of interaction. (H2.c, $w_{EXT} = 150$)

M_{Click}: Based on mouse clicks, which, like text selections, are a strong indicator of current reading. If users click on fragments / paragraphs, this modifier splits the available time among them. (H2.b, $w_{EXT} = 70$)

M_{Move}: Based on the users' tendency to move their mouse while reading. This modifier sets weights according to the time the mouse cursor has been moved above a fragment. The more users tend to move their mouse, the stronger the weight of this modifier. (H1.a-d and particular H1.c-d, $w_{EXT} = 45$)

$M_{MousePositions}$: Even if the mouse is not moved the position of the cursor may be used to identify the area of interest. This modifier considers the placement of the mouse over a fragment, as well as its placement in at a position that falls within the vertical constraints of the fragment (e.g., in the white-space area next to the text). (H1.e, $w_{EXT} = 45$)

$M_{ScreenAreas}$: Even if there are only few interactions we may make further assumptions on what has been read. Most people prefer to read in the center of the screen, so if the page is long enough that a user could scroll up or down (the first and last paragraphs of a page definitely have to be read while on top/bottom of the screen), this modifier puts its weight on the centered 80% of the page. A more fine-grained distribution over different parts of the screen or additional knowledge on the user's preferred reading area might improve a future version of this modifier. (adjusted H3 as per Fig. 2, $w_{EXT} = 5$)

$M_{Visibility}$: The simplest modifier, this one just splits the time among all visible paragraphs based on the number of words they contain. ($w_{EXT} = 1$)

Table 6. Classification distance of paragraphs in the Go course

Dist.	**# Par.**	**%**	**Cumulative %**
0	746	78.7%	78.7%
1	143	15.1%	93.8%
2	47	5.0%	98.7%
3	12	1.3%	100.0%
Total	948	100.0%	

Table 7. Classification distance of paragraphs in Questions page

Dist.	**# Par.**	**%**	**Cumulative %**
0	41	85.4%	85.4%
1	3	6.3%	91.7%
2	2	4.2%	95.8%
3	2	4.2%	100.0%
Total	48	100.0%	

4 Results

In order to evaluate our algorithm we measured the reading speed of each user (rate of words per minute). We used that rate, along with the number of words in each paragraph, to estimate the time the user would require for reading it ($T_{p_{req}}$). We then used that in conjunction with the time the user spent on the paragraph, as per the algorithm's predictions ($T_{p_{pred}}$), to define four "levels" of reading for paragraphs:

- level 0 (paragraph skipped): $T_{p_{pred}} < 0.3 \cdot T_{p_{req}}$
- level 1 (paragraph glanced at): $0.3 \cdot T_{p_{req}} \leq T_{p_{pred}} < 0.7 \cdot T_{p_{req}}$
- level 2 (paragraph read): $0.7 \cdot T_{p_{req}} \leq T_{p_{pred}} < 1.3 \cdot T_{p_{req}}$
- level 3 (paragraph read thoroughly): $1.3 \cdot T_{p_{req}} \leq T_{p_{pred}}$

The user's fixations have been used to calculate the baseline reading level our algorithm should be compared against. Table 6 shows the absolute distances between the calculated reading level and the baseline from the eye tracking data. In 78.7% of all cases the algorithm was able to classify the paragraph correctly. However, not only the exact matches, but also the difference between the baseline category and the level selected is important. In 93.8% of all cases this distance is only 0 or 1.

Table 8 shows in more detail how paragraphs of each level have been categorized by the algorithm. The highest precision was reached for paragraphs that have been skipped or read thoroughly. However, even for the intermediate levels the algorithm classified most paragraphs correctly.

The focus of our experiment was to test the algorithm in the context of reading learning materials. Nevertheless, it is worth noting that the algorithm performs comparably well in the other contexts tested. For example, on pages where users answered questions (a task that inherently requires more interaction), the algorithm performed even better than in the case of the Go course (see Table 7). However, we concentrate on the learning scenario where it is more difficult to get valid information due to reduced requirements for interaction.

5 Conclusions and Ongoing Work

This paper has demonstrated that it is possible to predict, with satisfactory precision, the users' reading behavior on the basis of client-side interaction. In

Table 8. Classification of paragraphs split by the actual reading level (L0-3) – context: Go course

L0	**# Par.**	**%**
L0	**596**	**89.1%**
L1	46	6.9%
L2	15	2.2%
L3	12	1.8%
Total	669	100.0%

L1	**# Par.**	**%**
L0	23	26.7%
L1	**34**	**39.5%**
L2	18	20.9%
L3	11	12.8%
Total	86	100.0%

L2	**# Par.**	**%**
L0	7	8.0%
L1	15	17.2%
L2	**43**	**49.4%**
L3	22	25.3%
Total	87	100.0%

L3	**# Par.**	**%**
L0	0	.0%
L1	14	13.2%
L2	19	17.9%
L3	**73**	**68.9%**
Total	106	100.0%

our experiments, users visited all pages of provided hypertext material. A traditional AHS might, thus, assume everything has been read. In contrast, using the proposed approach, we were able to determine that 70% of the paragraphs were not read, and users focused on certain paragraphs instead of reading entire pages. Our experiment has shown that the algorithm, using mouse and keyboard events, can correctly identify a paragraph's "reading level" in 78.7% of all cases (and in 93.8% of the cases calculate the correct level ±1).

The algorithm, in its current form, has weaknesses that need to be addressed. To start with, it is geared towards pages that contain one main column of text. While this may be typical for learning content, enhancements are required before the algorithm can satisfactorily handle multi-column page content. A related question is how well the algorithm might perform in mobile settings, with different screen factors (and, therefore, different amounts of text visible at a time) and potentially different interaction patterns (brought forth by the screen factor, or by alternative input techniques available). Another area that requires further work is the establishment of the effects of external modifier weights in different reading contexts (e.g., with less text visible at a time, the visible part of a page may be a stronger indicator on what is currently being read).

Among the strengths of this algorithm is its extensibility. For example, additional input devices may be easily integrated through client-side "drivers" and the introduction of corresponding modifiers (e.g. a webcam, eye tracking, etc.). The same is true for interaction patterns that may be established as evidence for reading behavior in the future.

Further to the above, and specifically in the domain of learning, we intend to test the effects of having access to predictions of reading behavior on learner models and their use in adaptive educational hypermedia systems. Our next experiment will use the presented algorithm to make predictions on which questions relating to course content a learner is likely to be able to answer, based on what that learner has (been predicted to have) read from that content.

Finally, as soon as the algorithm has matured and been shown to be of general applicability, we intend to make the implementation (along with the accompanying JavaScript library for monitoring) publicly available.

Acknowledgments

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

References

1. Cha, H.J., Kim, Y.S., Park, S.H., Yoon, T.B., Jung, Y.M., Lee, J.H.: Learning Styles Diagnosis based on User Interface Behaviors for the Customization of Learning Interfaces in an Intelligent Tutoring System. In: Intelligent Tutoring Systems. vol. 4053, pp. 513–524 (2006)
2. Chee, D.S.L., Khoo, C.S.: Users' Mouse/Cursor Movements in Two Web-Based Library Catalog Interfaces. In: Khalid, H., Helander, M., Yeo, A. (eds.) Work with Computing Systems, pp. 640–645. Damai Sciences, Kuala Lumpur (2004)
3. Chen, M.C., Anderson, J.R., Sohn, M.H.: What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In: CHI '01: CHI '01 ext. abstracts on Human factors in computing systems, pp. 281–282. ACM (2001)
4. Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: Intelligent User Interfaces. pp. 33–40 (2001)
5. Goecks, J., Shavlik, J.W.: Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. In: Int. Conference on Intelligent User Interfaces - Proc. of the 5th Int. Conference on Intelligent User Interfaces. pp. 129–132 (2000)
6. Hauger, D.: 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. Trento, Italy (2009)
7. Hauger, D., Paramythis, A., Weibelzahl, S.: Your Browser is Watching You: Dynamically Deducing User Gaze from Interaction Data. In: UMAP 2010 - Adjunct Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization. pp. 10–12. Big Island of Hawaii, HI, USA (2010)
8. Hauger, D., Van Velsen, L.: Analyzing Client-Side Interactions to Determine Reading Behavior. In: ABIS 09 - Adaptivity and User Modeling in Interactive Systems. pp. 11–16. Darmstadt, Germany (2009)
9. Hijikata, Y.: Implicit user profiling for on demand relevance feedback. In: IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces. pp. 198–205. ACM, New York, NY, USA (2004)
10. Hofmann, K., Reed, C., Holz, H.: Unobtrusive Data Collection for Web-Based Social Navigation. In: Workshop on the Social Navigation and Community based Adaptation Technologies (2006)
11. Liu, C.C., Chung, C.W.: Detecting Mouse Movement with Repeated Visit Patterns for Retrieving Noticed Knowledge Components on Web Pages. IEICE - Transactions on Information and Systems E90-D, 1687–1696 (October 2007)
12. Mueller, F., Lockerd, A.: Cheese: tracking mouse movement activity on websites, a tool for user modeling. In: CHI '01: CHI '01 extended abstracts on Human factors in computing systems, pp. 279–280. ACM, New York, NY, USA (2001)
13. Rodden, K., Fu, X., Aula, A., Spiro, I.: Eye-mouse coordination patterns on web search results pages. In: CHI '08: CHI '08 extended abstracts on Human factors in computing systems. pp. 2997–3002. ACM, New York, NY, USA (2008)
14. Spada, D., Sánchez-Montañés, M.A., Paredes, P., Carro, R.M.: Towards Inferring Sequential-Global Dimension of Learning Styles from Mouse Movement Patterns. In: AH. pp. 337–340 (2008)
15. Stephanidis, C., Paramythis, A., Karagiannidis, C., Savidis, A.: Supporting Interface Adaptation: The AVANTI Web-Browser. In: Stephanidis, C., Carbonell, N. (eds.) Proc. of the 3rd ERCIM Workshop "User Interfaces for All" (1997)
16. Ullrich, C., Melis, E.: The Poor Man's Eyetracker Tool of ActiveMath. In: Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (eLearn-2002). vol. 4, pp. 2313–2316. Canada (2002)