# Tracking Students' Internet Browsing in a Machine Exam

Henrik Nygren, Juho Leinonen, Arto Hellas
University of Helsinki
Helsinki, Finland
{henrik.nygren,juho.leinonen,arto.hellas}@helsinki.fi

## ABSTRACT

Traditionally, introductory computer science courses have focused on teaching programming, and have not included teaching information retrieval skills. However, a large part of a programmer's time is spent looking at documentation or browsing the internet for guidance on how to solve the small subtasks that programming often consists of or which library to use for a specific need.

We have developed a browser-plugin that tracks how students use online resources during a machine exam. Such a tool could be used – for example – to detect whether there is a difference between the browsing behavior of high- and low-performing students. To this end, we conduct a case study with the tool where we examine students' browsing in a lab-based programming exam.

In the future, the tool could be used to examine students' browsing and possibly inform decisions on how to teach information retrieval skills to students.

**ACM Reference Format:**
Henrik Nygren, Juho Leinonen, Arto Hellas. 2017. Tracking Students' Internet Browsing in a Machine Exam. In *CSERC '17: The 6th Computer Science Education Research Conference, November 14, 2017, Helsinki, Finland*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3162087.3162103

## 1 INTRODUCTION

Learning any craft such as programming requires plenty of effort. The students need to familiarize themselves with the relevant concepts, the underlying machinery, and to build procedural knowledge that aids them in solving programming problems. During the last decades, the craft of programming has evolved a long way from the time where practitioners referred to printed manuals to determine how a specific function or library works. Currently, printed manuals are almost extinct, and instead, developers use a myriad of online sites and forums to ask and to search for information.

These information retrieval skills that are a core necessity for any programmer are rarely taught or assessed in introductory programming courses. Instead, programming knowledge is typically assessed using exams, be it either pen-and-paper exams, lab exams, or take-home exams. To our knowledge, however, students' ability to look for information whilst working is not assessed, and as such, it is likely that students do not consider that as something that they should practice.

In this article, we present a browser plugin that can be used to collect information on sites that students visit while browsing as well as to collect the queries that students make while referring to e.g. the help of Google or Stack Overflow. Through a case study of a lab-based programming exam with internet access, we show how the tool can be used to determine whether poorly performing students search for information in a different fashion than students who perform better.

In the future, such a tool could also be used as a voluntary part of a suite that could be used to, e.g., inform students about academic misconduct such as plagiarism. Through analysis of students' browsing history while they have been working on course assignments or an the exam, one could, for example, identify sites that provide solutions to course problems. Such a site collection could then be added as a part of the plugin, which could subsequently inform students if they are about to visit an unwanted site.

The tool that we discuss here is similar to the one that was used by Sadowski et al [13], who studied Google developers' use of a specific domain. To our knowledge, however, their tool is not publicly available. Students' browsing behavior has been studied previously in e.g. web usage mining [14], but again, to our knowledge, the differences in student groups have rarely been studied.

This article is organized as follows. In Section 2, we briefly go over related previous research. Then, in Section 3, we present the tool for tracking students' browsing. In Section 4, we examine students' internet usage in a programming exam as a case study of using the tool. We discuss possible uses of the tool in Section 5, and conclude the article in Section 6.

## 2 RELATED WORK

Information retrieval skills on the Internet have been studied in the context of what types of search terms individuals use, and whether the individuals refine their search terms in order to pinpoint relevant information. Anick studied search term refinement using anonymized activity logs of the AltaVista search engine [3]. They collected all changes to the search box and all the visited search results. All of the collected events contained timestamps and identifiers that were used to distinguish users. The search logs were clustered into user sessions based on the sequential search event times, which were then analyzed in order to determine whether the user found what they were looking for, and to see how the search terms were refined.

Sadowski et al. investigated how developers search for code in code repositories [13]. A group of Google developers installed a browser extension that presented them with a survey whenever they accessed an internal code search site. Additionally, the extension collected all the requests to the search site. The requests were


*CSERC '17, November 14, 2017, Helsinki, Finland*

ACM ISBN 978-1-4503-6338-9/17/11…$15.00
https://doi.org/10.1145/3162087.3162103

then divided into sessions, and each session was associated with the survey results. This data was used to determine the reasons for using the search tool, what the typical search queries and sessions were composed of, and if there were distinct patterns in the searching behavior.

Browsing behavior has also been collected in parental supervision programs. The patent by Bates et al. [4] discusses the creation and use of a user profile that contains information on whether specific sites are allowed to the user, and the user's history of visited web addresses.

The use of data from such systems have been studied, for example, within the domain of web usage mining [14]. Srivastava et al. discuss options for collecting data, ranging from server-side to client-side data collection options. They point out that client-side data collection requires user cooperation, and that incentives may help with having more participants use the collection tools. Finally, they discuss a set of data mining options for mining web log data, including opportunities for user characterization and privacy issues [14].

Within the computing education domain, while there are several tools that can be used to monitor the programming process [7], the collection and use of students' browsing behavior has typically been limited to single systems. For example, Kurhila et al. [10] collected students' browsing behavior within an online educational system in order to show other students which online resources individuals within the system were accessing. Similarly, Leppänen et al. [11] focused on the students' use of a particular course material, and tracked the visible viewport of the students in order to determine areas of interest within the material.

The use of course materials have been used to also predict students' course performance. For example, Romero et al. [12] used a set of machine learning classifiers on student browsing data from a Moodle system. In the study, the usage of Moodle was found to be indicative of course success, albeit the authors did note that the accuracies were not too high.

To our knowledge, analyzing students' browsing behavior in a laboratory examination has not been studied so far. This can partially be related to the difficulty in gathering data from students, as client-side data collection can be difficulty to justify due to e.g. privacy reasons [14]. When considering open book exams, they may be a valid option for assessing procedural knowledge – several authors have pointed out the benefits of lab exams in programming courses [5, 8, 9], and having access to online resources during the examination may be a meaningful addition. We do acknowledge that open book examinations may lead to worse recall when compared to traditional closed book exams [1]. In the context of programming, the benefits of being able to refer to a manual and to search for information are hard to question.

## 3 THE TOOL

The tool[1] that we propose collects data from students' browsing as they visit different pages. Contrary to many of the existing approaches for collecting data, the tool is not limited to collecting data from a single domain or a single site, but can be used to collect the whole browsing history. Similarly, if the students are expected to install the tool themselves, the administrative overhead is small – there is no need to e.g. configure a firewall or to set up separate exam distributions.

The tool functions as a a browser extension that can be installed using a few clicks, and the address to which the browsing data is sent can be configured. The data that is collected by the plugin includes visited web sites – in practice, we use the *tabs* API of Google Chrome to collect events associated with the browser tabs. That is, every time a browser tab navigates to a new address, we record the address, the site's title, whether the tab was active at the time, loading status, identifier of the tab's window, and the tab's identifier. In addition to logging the addresses of the visited sites, the data also contains additional information including a timestamp for each event. Search terms and page topics were also included as they often appear in the page title.

In principle, one could also store the request content including the request body, but due to the possibility of users attempting to authenticate to specific services, the tool does not currently store the body for privacy reasons.

The recorded data is sent to a configurable server that stores all the data that it receives. In our case, the tool did not require authentication, but the students were identified through the use of a student-specific course page and their IP address. We are considering the opportunity for adding a random UUID string to the tool that could be used to distinguish between users in the future.

## 4 MONITORING INTERNET USAGE IN A PROGRAMMING EXAM

To demonstrate the use of the plugin, we show how it has been used in studying how students browse the Internet in an introductory programming exam.

### 4.1 Context and data

We applied the plugin in an introductory programming exam offered by the University of Helsinki during Spring 2017. The exam was given as a lab exam in which each student had access to a computer and the relevant development environments, and in which each student could access the internet.

Communication or the use of chatrooms was prohibited, and the students were not allowed to use any services that required authentication outside the exam page and the assessment system that the course used. The choice of browsers was limited to Google Chrome that had the plugin preinstalled.

The duration of the exam was three hours, and the students were given both essay and programming questions. Each internet address that students accessed was stored to a database that the exam supervisors monitored. As there were a total of 114 students who attended the exam, multiple exam sessions were facilitated.

On average, each student visited 22 links during the exam, and accessed approximately 4 different domains (see Tables 1 and 2). That is, majority of the visited links were related to a specific domain. There were, however, students who did not use online resources at all, as well as students, who used online resources extensively – one student visited 110 links during the three hour period.

[1] Contact the first author for additional details and a possible copy.

**Table 1: Amount of visited links**

| | mean | median | sd | min | max |
|---|---|---|---|---|---|
| All students | 21.8 | 21.5 | 22.3 | 0 | 110 |
| Top students | 21.5 | 14 | 18.0 | 0 | 64 |
| Bottom students | 35.0 | 27 | 29.8 | 0 | 110 |

## 4.2 Research questions and analysis

Our case study focuses on whether the students who performed well in the exam differ in their use of online resources from those who performed poorly in the exam. The research questions for our study are the following:

(1) What kinds of online resources do students use in an introductory programming exam?
(2) Does the behavior of students who perform well in the exam differ from those who perform poorly?

To answer Research Question 1, we used browsing data from the students' exam session. We analyzed the exam data using a quantitative approach, where we first analyzed the number of visited links and domains, and then categorized the visits into sites.

To answer the Research Question 2, we split the students into poorly performing and well performing students by dividing the student population into three parts based on their exam performance in an exam with a maximum of 15 points. The resulting tertiles were based on the students' scores. The lowest tertile had students who scored from 0 to 7 points, students in the middle tertile scored 8-9 points, and the highest tertile had students who scored 10-15 points. The sizes of the tertiles were approximately the same: the lowest tertile had 33 students in it, the middle tertile had 52 students, and the highest tertile had 29 students. For the rest of the paper, we define the highest-scoring tertile as high-performing students and the lowest-scoring tertile as low-performing students.

To analyze the sites the students visit during the programming part of the exam, we filtered out links the students visited during the essay part of the full examination. Additionally, as the students were allowed to visit the essay writing page during the programming part of the exam, we filtered out the essay writing page as it is not related to the programming assignments. We then categorized the remaining links to seven categories (see Table 3): Google-searches, the course material, Stack Overflow content, the students' own old solutions, general course information, Java documentation, and all other domains. The most used resources the students used while they were programming were Google, the course material, Stack Overflow (a QA website dedicated to programming related questions), and the students' own previous solutions.

The differences between the tertiles were analyzed using the Wilcoxon signed-rank test.

## 4.3 Results

The results of our analysis are summarized in Tables 1—4. Table 1 shows that the mean visited links for all students is very close to the mean visited links for top students, while the mean visited links for the poorly performing students is substantially larger. This in practice is a result of a few low-scoring students visiting a great amount of links, skewing the results for the whole group of low-performing students – the median amount of links offers a better view of the "typical" performance.

**Table 2: Amount of visited domains**

| | mean | median | sd | min | max |
|---|---|---|---|---|---|
| All students | 4.3 | 4 | 1.9 | 0 | 9 |
| Top students | 3.9 | 4 | 1.9 | 0 | 8 |
| Bottom students | 5.0 | 5 | 2.1 | 0 | 9 |

**Table 3: Visited links by category**

**(a) All students**

| | mean | median | sd | min | max |
|---|---|---|---|---|---|
| Google | 7.5 | 5 | 8.5 | 0 | 45 |
| Course material | 6.0 | 4 | 6.6 | 0 | 30 |
| Stack Overflow | 3.7 | 1 | 5.0 | 0 | 29 |
| Students' old solutions | 4.2 | 0 | 11.6 | 0 | 84 |
| Course information | 2.4 | 2 | 2.7 | 0 | 15 |
| Java documentation | 0.6 | 0 | 1.4 | 0 | 8 |
| Other | 1.3 | 0 | 2.5 | 0 | 20 |

**(b) Top students**

| | mean | median | sd | min | max |
|---|---|---|---|---|---|
| Google | 6.2 | 4 | 8.1 | 0 | 33 |
| Course material | 4.8 | 3 | 7.1 | 0 | 30 |
| Stack Overflow | 3.5 | 1 | 4.9 | 0 | 18 |
| Students' old solutions | 3.2 | 0 | 7.0 | 0 | 25 |
| Course information | 2.3 | 1 | 3.0 | 0 | 13 |
| Java documentation | 0.6 | 0 | 1.5 | 0 | 8 |
| Other | 0.8 | 0 | 1.3 | 0 | 4 |

**(c) Bottom students**

| | mean | median | sd | min | max |
|---|---|---|---|---|---|
| Google | 9.0 | 5 | 9.2 | 0 | 45 |
| Course material | 7.4 | 6 | 6.5 | 0 | 29 |
| Stack Overflow | 4.2 | 1 | 5.9 | 0 | 29 |
| Students' old solutions | 8.6 | 0 | 18.5 | 0 | 84 |
| Course information | 2.8 | 2 | 3.3 | 0 | 15 |
| Java documentation | 0.7 | 0 | 1.2 | 0 | 6 |
| Other | 2.2 | 1 | 3.9 | 0 | 20 |

When considering the amount of visited domains in table 2, a similar effect is observed. There is no significant difference between the number of domains visited by the different student subpopulations.

When looking at Table 3, a few interesting results can be noticed. For example, on average, the high performing students visit their old solutions less when compared to the low-performing students. High-performing students also tend to use Google and the course material somewhat less than the students in the bottom tertile.

**Table 4: Correlation with points. Asterisk indicates p < 0.05.**

**(a) Overall**

| | Correlation | p |
|---|---|---|
| Visited links | -0.24 | 0.01* |
| Visited domains | -0.18 | 0.06 |

**(b) By category**

| | Correlation | p |
|---|---|---|
| Google | -0.18 | 0.05 |
| Course material | -0.24 | 0.01* |
| Students' old solutions | -0.16 | 0.09 |
| Course information | -0.08 | 0.39 |
| Java documentation | -0.07 | 0.43 |
| Other | -0.15 | 0.11 |

Interestingly, there is not as big of a difference between the usage of Stack Overflow between the top and bottom students. This means that while the absolute amount of Stack Overflow usage is the same between the groups, the top students use Stack Overflow relatively more. When performing a statistical analysis for the difference of browsing behavior between the groups (Table 3), the differences are not statistically significant – potentially a product of highly skewed data.

To address whether the students' overall browsing activity is related to their exam performance, we conducted a regression analysis between the number of visited links and domains, and students exam points. The results are summarized in Table 4. Overall, there is a small negative correlation ($r = -0.24$) between the number of visited links and the exam performance – that is, visiting more links correlates negatively with exam scores. Similarly, same negative effect can be seen with Course material. In other words, students who rely more on course material during the exam perform marginally worse in the actual exam.

# 5 DISCUSSION

## 5.1 The Tool

In practice, the tool that we propose here is a lightweight addition to the existing systems that are used to collect students' data. As the tool can be installed as a browser plugin, it has rather little overhead from the administrative perspective as normal users are typically allowed to install plugins.

In our case, we used the plugin to monitor students' browsing behavior in a laboratory exam with open internet access. The use of the tool is not, however, limited to such situations, and in practice individuals could use it to monitor their own browsing behavior. Our intent is to provide instructors a possibility to support students in developing their information retrieval skills, and the tool is a first step towards that as it provides an opportunity to study the information retrieval process.

Moreover, as students information retrieval process is not limited to a single domain or a specific site, the tool provides an opportunity to study the paths that are taken to find information. Bundled with a questionnaire to ask if the students found what they searched for, better feedback on succesful browsing could be provided.

The tool could also be used to monitor the use of social networking sites such as Facebook, which could be used to provide information on off-task behaviors while studying. Such information could be then provided back to the student with suggestions on how to regulate their learning.

At the same time, we acknowledge that collecting and storing student data is a sensitive subject. If the collection of such browsing data can not be justified, then it should not be conducted. In our context, the students were expected to use the tool in an exam condition – having students use the plugin outside exam conditions would have to be voluntary.

## 5.2 Differences in browsing behavior

Looking at the results in Tables 1—4, we observe a trend that the top students are consistently closer in their behavior to the group of all students, whereas the bottom students are further from the average in their behavior. As the groups were about the same size, this indicates that there are likely a small group of individuals with distinctively different browsing behavior in the bottom group. That conclusion is also supported by the standard deviations being consistently larger within the bottom group when comparing to all students or the top students. At the same time, these differences were not statistically significant.

Stack Overflow was used relatively more by the top students compared to the bottom students. This might be due to it being an "expert" resource, and thus it is likely that proficient programmers are aware of its existence whereas novice programmers might not have used it as much previously. Additionally, poorly performing students visit their old solutions more when compared to the top students. One hypothesis is that more competent programmers are more used to searching for programming related information and thus go to Stack Overflow whereas novices go to their own solutions.

While previous work has found that closed-book exams might have better long-term recall [2], we believe that professional programmers are not likely to use books or notes during their work and instead opt to use the Internet to retrieve the information they need to complete their task. Thus, for a programming exam, it likely makes sense to extend the open-book examination to include Internet resources as well. A question arises information retrieval abilities affect the exam performance in this kind of a setting. For example, the information retrieval process in a traditional open-book exam is very different from the same process in an Internet open-book exam. For instance, one can not use search engines with physical course books. Additionally, there are vastly more resources available on the Internet compared to a book, and thus it is likely harder to identify relevant materials. Furthermore, in the context of programming, it is possible to find complete solutions to subtasks within an assignment whereas this is unlikely in many other fields.

Boniface et al. [6] found a statistically significant difference that those who used material more during open book exams did worse on the exam. Our results provide supporting evidence: we also observed a statistically significant negative correlation between the

students exam points and their number of visited links and the use of course material during the exam.

At the same time, the correlations are rather small, and the tests have not been corrected for multiple comparisons. Overall, we hypothesize that it is easier to find what one is looking for in the Internet when compared to a physical textbook. Thus, an effect that is statistically significant with physical books might not be as strong with Internet open-book exams. Similarly, it is possible that when the option to use online resources is given, some extensively use it instead of relying on their own intuition and recall.

### 5.3 Limitations

Both the tool and the case study have some limitations, which we will address here.

First, we acknowledge that having to install a tool to each browser can be cumbersome. At the same time, the installation can be organized through a centralized repository, or the installation can be facilitated as a starting task in an exam – the address to the exam site could be transmitted to the students through the plugin. In some institutions, where there is no access to laboratory or university-specific network settings, such a plugin is the sole opportunity for monitoring the browsing behavior.

Second, it is possible that students deactivate the plugin. In our study, we disabled the possibility to uninstall the plugin, but one could as well send signals to the database if the plugin is disabled.

Third, some may consider it a problem that the plugin functions only within Google Chrome and its derivatives. We acknowledge that this is an issue – in principle the plugin should also work in Firefox (and Chromium), but e.g. Mozilla does not allow installing any plugins to Firefox that are not approved to their online store.

Fourth, regarding the results, as the data for the case study comes from an exam that was held at the end of a 14-week programming course, the students had to do a lot of work to even participate in the exam. Thus, it is possible that the performance differences in the exam were – in the end – rather small could stem from the fact that all the participants in the exam were rather similar and thus there is little difference between their information retrieval skills.

Fifth, as we explicitly stated that we were monitoring internet browsing, it is possible that there was a chilling effect, i.e. that the browsing behavior of students was different than what it would have been had we not monitored their internet usage.

Sixth, the case study is only a simple example of what the tool is capable of. Further studies should be conducted that examine the sequence of website visits and also consider the duration that students spend on different websites. This would allow for a better comparison with previous work on open-book exams, where the time spent reading the material has been considered [6].

Finally, the amount of students in the case study was not very large, and thus it is possible that some effects that would have been statistically significant with a higher $n$ were not detected. That is, due to the small sample size, the population validity is low.

## 6 CONCLUSIONS

In this work, we have presented a tool for tracking the internet browsing of students in an educational context. The tool is easy to install as it is provided as a browser plugin, and requires no access to e.g. university or lab firewall settings.

As an example of what kind of research is possible with the data provided by the tool, we conducted a case study of examining students' browsing behavior in an exam of an introductory programming course. The result of the case study was that low- and high-performing students' browsing behaviors were quite similar, although it seems that low-performing students visit more links during the exam on average even though this difference is not statistically significant. Our results also suggest that using course materials and visiting links in general is negatively linked with exam performance.

We are interested in replicating the study with a larger student population to see whether the effect persists. Similarly, we are looking for other factors than exam performance that may distinguish the students who used the online resources scarcely from those who used them extensively.

In the future, we are interested in using the tool as a way of studying students' browsing behavior during the course. We believe that such tool could be used to estimate students' information retrieval skills, to provide feedback aimed at improving searches, and to potentially identify students' misconceptions and problematic areas in the course materials.

## REFERENCES


[1] Pooja K Agarwal, Jeffrey D Karpicke, Sean HK Kang, Henry L Roediger, and Kathleen B McDermott. 2008. Examining the testing effect with open-and closed-book tests. *Applied cognitive psychology* 22, 7 (2008), 861–876.

[2] Pooja K Agarwal and Henry L Roediger III. 2011. Expectancy of an open-book test decreases performance on a delayed closed-book test. *Memory* 19, 8 (2011), 836–852.

[3] Peter Anick. 2003. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 88–95.

[4] C.L. Bates, B.J. Cragun, and P.R. Day. 2004. Method and computer program product for implementing parental supervision for internet browsing. (June 1 2004). https://www.google.com/patents/US6745367 US Patent 6,745,367.

[5] Jens Bennedsen and Michael E Caspersen. 2007. Assessing process and product: a practical lab exam for an introductory programming course. *Innovation in Teaching and Learning in Information and Computer Sciences* 6, 4 (2007), 183–202.

[6] David Boniface. 1985. Candidates' use of notes and textbooks during an open-book examination. *Educational Research* 27, 3 (1985), 201–209.

[7] Petri Ihantola et al. 2015. Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies. In *Proceedings of the 2015 ITiCSE on Working Group Reports (ITICSE-WGR '15)*. ACM, New York, NY, USA. https://doi.org/10.1145/2858796.2858798

[8] Arto Hellas, Juho Leinonen, and Petri Ihantola. 2017. Plagiarism in Take-home Exams: Help-seeking, Collaboration, and Systematic Cheating. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 238–243. https://doi.org/10.1145/3059009.3059065

[9] Norman Jacobson. 2000. Using on-computer exams to ensure beginning students' programming competency. *ACM SIGCSE Bulletin* 32, 4 (2000), 53–56.

[10] Jaakko Kurhila, Miikka Miettinen, Petri Nokelainen, and Henry Tirri. 2002. EDUCO-A collaborative learning environment based on social navigation. In *AH*. Springer, 242–252.

[11] Leo Leppänen, Juho Leinonen, Petri Ihantola, and Arto Hellas. 2017. Using and Collecting Fine-grained Usage Data to Improve Online Learning Materials. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track (ICSE-SEET '17)*. IEEE Press, Piscataway, NJ, USA, 4–12. https://doi.org/10.1109/ICSE-SEET.2017.12

[12] Cristobal Romero, Pedro G Espejo, Amelia Zafra, Jose Raul Romero, and Sebastian Ventura. 2013. Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education* 21, 1 (2013).

[13] Caitlin Sadowski, Kathryn T Stolee, and Sebastian Elbaum. 2015. How developers search for code: a case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 191–201.

[14] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. 2000. Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter* 1, 2 (2000), 12–23.