Analyzing Programming Education Quality based on Students' Questions

Anne van Harten
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
a.a.vanharten@student.utwente.nl

ABSTRACT

With the increase of digital tools used within programming education, new possibilities open up to analyze the quality and delivery of a course by students. Without data collected by digital tools, decisions are often made by educators that are not fully aware of the problem areas for students. By using data collected from a digital question queuing system, we can identify new points of interest for educators to improve course materials. The textual question data from this system can be processed to extract keywords and trends. Combining this with data of the timing and corresponding exercises results in valuable insights that can be overlooked when going on anecdotal evidence. These points of interest include unclear and harder exercises and programming concepts, and other time savers for both students and teaching assistants.

Keywords

Programming education, Lab queue, Lab assistance, Course material usage, Learning material evaluation

1. INTRODUCTION

Within programming education many different techniques are used to make as many students succeed as possible. One of these techniques that has been around for a very long time are closed lab sessions. In these session students typically raise their hand when they have a question about a certain assignment, which a teaching assistant will respond to. Even with the rapid increase of online lab sessions and Massive Open Online Courses (MOOCs), face to face session are still a valuable and effective option [5]. However, in sessions with large numbers of students, the students typically have to wait longer and the teaching assistants lose overview of who is next.

Since students have to raise their hand to ask a question, both the student and teaching assistant need to keep track of who is being helped. With larger numbers of students, this method results in students waiting eagerly and trying the get the attention from a teaching assistant. This creates a problem for students since they are often unable to continue working when they are waiting for help. Another problem introduced with this method is that the queue of students can no longer be tracked by the teaching assistant and is often no longer fair. To resolve these issues, the idea of electronic queuing has been proposed to move this waiting list to a web tool. This stimulates students to continue on working as they don't need to attract the attention of the teaching assistants [3]. Similar systems have been used effectively to reduce waiting time and overhead for both the student and the teaching assistants as well as providing a fairer process [8]. With the move of lab sessions to a more digital environment, there is a sudden influx of data that is unobtainable with face to face questions. Systems such as electronic queuing open up the possibility to identify possible pitfalls and complicated assignments.

This paper explores the new possibilities of identifying course quality and possible improvements based on data collected by such an electronic queuing system. By using various methods of analysis we will provide new insights into programming education.

31st Twente Student Conference on IT July. 5th, 2019, Enschede, The Netherlands.

2. BACKGROUND

2.1 TA-Help.me

The data used in this research has been collected through a web application called TA-Help.me. This application can be used to facilitate fair electronic queuing of questions during lab sessions. TA-Help.me facilitates this process online such that students no longer have to raise their hand, but add themselves to a queue whenever they have a question.

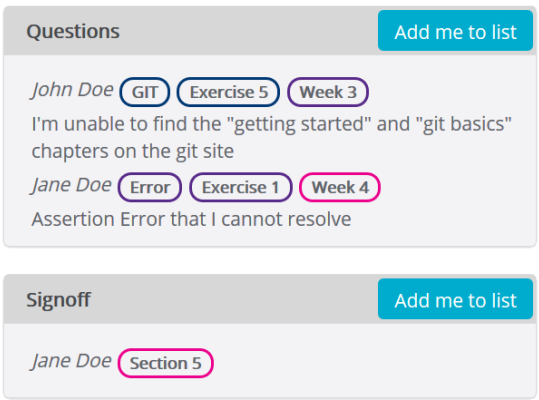

Figure 1. An example of a lab session using TA-Help.me

What differentiates TA-Help.me from a standard electronic queuing system, is that it extends upon it with different lists, categories and textual questions. An example of such

a list is shown in Figure 1. This method allows the teaching assistants to have a clear overview of the problem of the student making them able to prioritize and specialize on certain types of questions. Within the University of Twente, this has been widely used and regarded as preferable by the teaching assistants over plain queuing.

2.2 Dataset

The dataset that will be used in this research contains information about the questions asked during tutorial sessions that used the tool TA-Help.me. In the dataset is anonymous data corresponding to the questions asked by students to the teachings assistants. The type of data and the format of the dataset is shown in Table 1. The dataset consists of a total of 30.776 entries from a total of 4 programming courses. Of these entries, 2864 contain a textual question. Only the largest course (13.994 entries) made use of subcategories. In that particular course there were 2 lists, one with 7 categories, the other with 10 categories, 151 subcategories and in further subcategories respectively 470 and 604. An example of how these categories look in practice can be seen in Figure 1, in the first question the furthest subcategory is GIT, with it's parent being Exercise 5 and one further Week 3.

Table 1. The format and fields of the dataset

Session	The session in which the question was asked, of which the start and end time are known.
Time opened	The time the student submitted the question.
Time closed	The time the question was accepted by the teaching assistant.
Action	What action was taken, either Answered by the teaching assistant or Cancelled by the student.
List	The list in which it was submitted, in this dataset only Questions or Sign-off.
Text	The actual question typed out by the student, this is not in every entry.
Categories	Possible categories corresponding exercise to the question of the student, this is not in every entry. An example of categories can be seen in Figure 1,

3. RESEARCH QUESTIONS

In this paper the following research questions will be addressed:

RQ1 What are possible points of interest for course design that can be identified from the dataset?

RQ2 How can the dataset be used to identify points of interest for course design?

- **RQ2.1** How can **textual question data** be used to identify points of interest for course design?
- **RQ2.2** How can **categories** be used to identify points of interest for course design?
- **RQ2.3** How can **the timing of questions** be used to identify points of interest for course design?

4. RELATED WORK

There are various previous researches done on quantifying the quality of education and finding methods of improving it. Brown and Altadmri [4] found that educators often make incorrect claims about which errors students frequently commit, and are often unaware of the frequency of some errors. This makes the need for a good insight from another perspective very valuable to analyze course material quality.

In a literature review by Ithantola et al. [6], they show that there is a substantial amount of research done on using student submissions, snapshots and keystrokes to analyze educational quality. Most papers (63 studies, 83%) used descriptive statistics methods to quantify their results. Classification of results was also present in multiple studies, with both automated and interpretive classification. Since electronic queuing is a new technology, no studies have yet been done to check the effectiveness of using such a dataset for an analysis. This paper can fill this gap and explore the possibility of introducing a new form of data to improve course material quality.

Leppänen et al. uses their dataset of page movement on a website with course materials to identify points of interest [7]. They create heat maps to determine exercises or parts of the page that students commonly return to. Comparing this to our dataset, we can identify points on interest by analyzing exercises that students get frequently or repeatedly stuck on.

In order to determine the similarity of textual data, a keyword-based method for measuring similarity was analyzed by Bi et al [1]. By using this form of natural language processing, similarity can be analyzed and grouped to identify possible points of interest.

5. METHODOLOGY

5.1 Identifying points of interest

In order to identify points of interest from the dataset, first the potential and desirable points have to be determined (**RQ1**). Since no previous research has been done on a dataset similar to this, a clear idea has to be formed of what can be done with the data, and what is desirable. In order to get a clear view of this, a closer look has to be made at the dataset. Apart from this, often sought for statistics by educators are good potential points of interest to improve course design.

One of the educators of the courses used in this research pointed out that their primary interest was to be able to have a clear view of what did not go well in order to improve it for a next semester. While this is very broad, it is helpful to look at things such as room capacity during tutorial sessions and subjects that might be difficult or unclear to students.

5.2 Dataset cleaning

The raw data extracted from TA-Help.me as described in subsection 2.2 contained many invalid entries. This occurred primarily due to users testing out the system. To properly analyze the data these have to be removed. Most rooms used for testing had the word 'test' in it's name, or were made at times outside of standard lecture times. These entries and rooms have been taken out of the dataset to ensure more accurate results.

Another issue with the raw dataset is that some records still contained some data which could be used to identify a specific student. This was primarily due to textual data containing personal information. Since this study does not intend to look into specific students, but the course as a whole, these occurrences have been removed to preserve the anonymity of the students.

5.3 Analyzing Textual Data

In order to answer **RQ2.1** and see how textual question data can be used, there are two necessary parts. The first part is to pre-process the data in order for it to be analyzed. The second step is analyzing this data and determining how well it contributes to identifying points of interest.

5.3.1 Pre-processing

The textual data contains mostly short sentences typed out by students. After inspection the sentences seem to contain many grammatical errors and are often incomplete and unclear. For example: "Help" or "board" were asked very often. In order to gather valuable data from these questions, a lot of focus has to go into the removal of noise. In order to achieve this, three steps have to be performed: tokenization, Part of Speech (POS) tagging and normalization.

Extracting tokens from a sentence is a simple process. This can be done by splitting on whitespaces, this creates the opportunity to process the words of a sentence individually. An example of how tokenizing works is as follows:

'How do I shorten JML so that checkstyle is okay with it'

$$\downarrow$$

$$[How, do, I, shorten, JML, so, that, checkstyle, is, okay, with, it] \quad (1)$$

After tokenizing, the next step is POS tagging. In this step every token gets assigned a lexical category such as noun, verb, adjective or adverb. The Natural Language Toolkit (NLTK) is able to do this effectively [2]. After this step, the structure of the sentence becomes more clear and we can focus on the essential parts of the sentence. When inputting the tokens we previously gathered into the NLTK we get the following:

$$[How, do, I, shorten, JML, so, that, checkstyle, is, okay, with, it]$$

$$\downarrow$$

$$[(How, WRB), (do, VBP), (I, PRP), (shorten, VB), (JML, NNP), (so, IN), (that, DT), (checkstyle, NN), (is, VBZ), (okay, JJ), (with, IN), (it, PRP)] \quad (2)$$

The last step needed to process the textual data is normalization. With this step it is possible to remove redundant words and extract the most likely core of the question. Firstly, the NLTK is used to remove all English stopwords from the list of tokens. This makes sure that words such as 'How' and 'is' are taken out. After this, the tokens are stripped of special characters and made lowercase to generalize all words. Finally, the tokens are lemmatized using the previously gathered lexical categories and the NLTK. By lemmatizing the inflected forms of a word, they will be grouped together such that they can be analyzed together. This means that words such as 'walk' and 'walking' are assumed to be the same, and grouped under 'walk'. The example of this full process of normalizing is as follows:

$$[(How, WRB), (do, VBP), (I, PRP), (shorten, VB), (JML, NNP), (so, IN), (that, DT), (checkstyle, NN), (is, VBZ), (okay, JJ), (with, IN), (it, PRP)]$$

$$\downarrow$$

$$[shorten, jml, checkstyle, okay] \quad (3)$$

5.3.2 Grouping tokens

In the previous steps a sentence has been turned into a set of important tokens, while removing all the redundant ones. With these tokens it is then possible to look into trends between questions by grouping them together. In the dataset this is done on a course wide basis. This means that all questions of a specific course will be broken into the set of important tokens and then grouped together by occurrence. After processing it is then possible to determine the topics that reoccur most often between different students and could thus be a good indicator of a harder or more complex subject.

When grouping together these tokens by occurrence a lot of information is lost in the process. As seen in the examples of the previous section, just 'jml' might not be enough, and it would be better to see if more students ask questions specifically with 'checkstyle' or 'shorten' as well. In order to achieve this it is possible to group the tokens not only by occurrence, but also indicate the tokens that they were most often grouped in an initial sentence with. This makes sure that this particular information is not lost and it is clearly visible if there is a specific token often combined with another one. In the examples of the previous section, in combination with a hypothetical set of normalized tokens such as:

$$[jml, checkstyle, error]$$

would result in a ranking of occurrence of tokens combined with their common pairs, as can be seen in Table 2 below. In this example, the number behind a token pair indicates the amount of times these tokens occurred in the same sentence.

Table 2. Example occurrence of tokens with pairs

Amount	Token	Common pairs
2	jml	checkstyle (2), shorten (1), okay (1), error (1)
2	checkstyle	jml (2), shorten (1), okay (1), error (1)
1	shorten	jml (1), checkstyle (1), okay (1)
1	okay	shorten (1), jml (1), checkstyle (1)
1	error	jml (1), checkstyle (1)

With larger sets of data, the upper rows can potentially give an indication of popular types of questions, which in combination with the pairs can be analyzed to determine the relevance of such a token.

5.4 Analyzing Categories

To determine how well categories can be used to determine points of interest (**RQ2.2**) the data has to be combined with other statistics in the dataset. Within this particular dataset the courses always contain a root category called Questions, and in some courses another root category called Sign-off. Within these categories it is possible to have sub-categories, which can go into an infinite depth, but in practice go no deeper than 5 levels in total. In order to use this data it is possible to group the entries from students by category. By doing so, it is possible to analyze how popular these self defined categories are. If this is combined with categories for exercises or concept, the result would be a clear view of the amount of questions in such a category. Apart from this, the data is able to determine how many unique questions have been asked by students. This can be an indicator if students often

have the same type of question in a category or if most are unique. This can then be compared by an average amount for all the categories to determine outliers.

5.5 Analyzing Timing

With every question that is asked, the time it was accepted by a teaching assistant is recorded. This data opens up the opportunity to do multiple types of analysis on the session and a course as a whole and determine how it can be used to identify points of interest (**RQ2.3**). One of these possibilities is to determine the number of questions per session, but also per day or per hour. This process only requires counts on the session, day of the week and hour a question was asked. Combined with the amount of sessions that were open during those hours can allow for a quick overview of busier and quieter sessions.

Another possibility with this data is to analyze the time it takes for a teaching assistant to answer a certain question. Since only the start time is recorded of the teaching assistant answering a question, some assumptions have to be made to extract this data. The main assumption is that once a teaching assistant is done with answering a question, they will go on to the next waiting student. As such, if there is a queue, the time the teaching assistant spents on a question can be assumed to be

$$answertime = timeclosed^2 - timeclosed^1 \qquad (4)$$

However, this does not result in an accurate time in some cases. To improve the accuracy some special cases have to be ignored. One of these cases is when a teaching assistant does not answer two consecutive questions in the same session, as the time difference would then be very large. Another case is when a teaching assistant closes multiple entries of the same person after each other, as this most often means they resolved everything consecutively. The final case is when a teaching assistant takes a break, making the predicted answer time inaccurate. In order to resolve this last case, it is necessary not to look at the individual answer times, but to look only at the average. When grouping these times by another factor such as categories, the average answer time becomes more accurate. In order to improve the accuracy even more, we exclude all the grouped results with less than 5 unique answer times and remove the top and bottom 10% of the other grouped results to remove outliers and get a more accurate average.

6. RESULTS

6.1 Points of Interest

Within the dataset there are various factors that can be used. Most significantly this includes keywords, categories and timing in combination with the amount these occur. From these factors, multiple possible points of interest can be identified. In this research, four points of interest will be attempted to be extracted from the data that has been collected. These include 'Rooms and questions per day and hour', 'Most popular categories', 'Commonly mentioned keywords' and 'Time spent answering'. Within the subsections the importance and results will be discussed. Other points of interest that had the possibility to be extracting from the dataset could include topics such as student progress and automated evaluation of a single session. In order to limit the scope of this research, it will only discuss the first four mentioned. In the results in the following sections only the data from the largest course is displayed and noteworthy details from the other courses will be separately mentioned.

6.2 Rooms and questions

To determine the amount of rooms and questions per day and per hour, the time opened has been used to determine where a question was placed. In Table 3 an overview can be seen of the rooms and questions of the largest course per day of the week. What can be seen in these results is that there are not significantly more questions on a specific day of the week.

Table 3. Rooms and questions per weekday

Day	Rooms	Questions	Questions per Room
Monday	82	3259	39.7
Tuesday	79	2476	31.3
Wednesday	98	3353	34.2
Thursday	43	1733	40.3
Friday	85	3100	36.5

To display the average amount of questions per room per hour, Monday is split up in hours in Figure 2. In the graph, clear troughs are visible for the start and end of the day and lunch breaks. What is not clear from this graph is that since sessions typically start at 8:45, Monday mornings are actually busier than an hour later. Since 6.6 questions in roughly 15 minutes, is more than 14.5 in an hour. This contradicts the idea that students tend to attend sessions less on Monday morning.

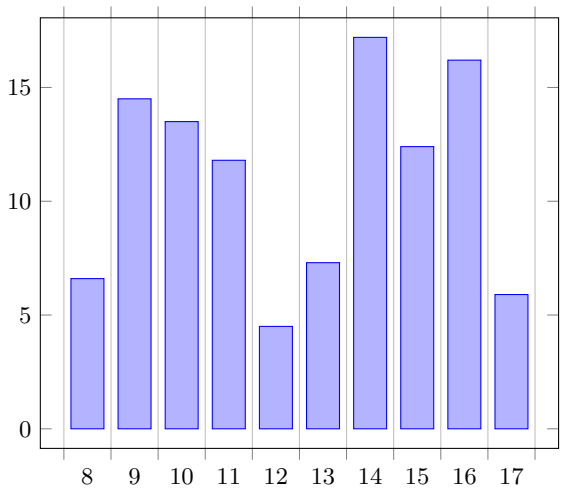

Figure 2. Average # of Questions per room per hour on Mondays

6.3 Most popular categories

In Table 4 the 10 most popular categories are displayed ordered by the amount of entries. The category column depicts the entire category tree in which the student placed its question as discussed in subsection 5.4. In this example, in the last entry the student has a question about Sorting, which is applicable to Exercise 4.08. For entries in the Signoff category, no question was needed, so these columns are empty. A clear outlier is the top entry, question about the project. This is expected due to the project spanning over multiple weeks. What is interesting on this result though, is that the amount of unique questions is almost half of the total entries, meaning students often have the same type of question. Upon further inspection the most asked question (10% of all), was about a specific part of the project, meaning this might be difficult or unclear to students.

Table 4. Most popular categories

Category	Entries	Unique questions
Questions -> Project	1171	690
Signoff -> Week 7	469	-
Signoff -> Week 6	314	-
Signoff -> Week 5	268	-
Signoff -> Week 4	190	-
Questions -> Other	99	78
Signoff -> Week 3	35	-
Questions -> Exercise 7.17 -> Producer-Consumer pattern	27	10
Questions -> Exercise 7.19 -> Other	23	17
Questions -> Exercise 4.08 -> Sorting	20	10

6.4 Commonly mention keywords

The keywords that were extracted as described in subsection 5.3 are ranked by occurrence and the top 10 is displayed in Table 5. Many of these keywords do not contain valuable information such as 'error' and 'help'. However, not much further into the table multiple topics return and seem to be asked many times. What can be seen is that there are many questions about 'server', often combined with 'client' or 'connect'. This could mean that setting up the client to connect to the server is a difficult or unclear task. Apart from that, keywords such as 'jml', 'board' and 'tile' occur very often meaning that explaining these subjects in more detail could save a lot of time for both students and teaching assistants. When looking at results further than the top 10 even more points of interest can be found on topics such as git and class diagrams.

Table 5. Commonly mentioned keywords

Amount	Keyword	Common pairs
135	error	test (6), get (5), still (5), find (4), junit (4)
129	work	doesnt (44), expect (20), git (12), dont (9), test (8)
107	help	plz (13), send (10), please (10), need (8), pls (8)
105	question	408 (10), unclear (6), general (6), dont (6), message (6)
104	server	client (45), connect (12), game (9), question (5), run (5)
94	jml	jmljml (288), need (5), kind (5), proper (4), cant (4)
92	board	tile (3), field (2), create (2), string (2), make (2)
72	tile	start (4), player (4), place (4), flip (3), board (3)
71	doesnt	work (44), git (11), test (5), run (5), expect (4)
68	client	server (45), connect (8), make (4), one (3), interaction (3)

6.5 Time spent answering

The last way of extracting points of interest is by analyzing the time spent to answer certain questions. The answer time that has been extracted in subsection 5.5 is used in combination with the categories to determine the time spent answering questions in a certain category. The top 10 results by total time are displayed in Table 6. In this table, the estimated total time is calculated by multiplying the time per entry by the amount of entries. This ensures outliers are not present in the total time.

The first point of interest that can clearly be seen from this table is the total time needed to sign off exercises. There has been much recent discussion as to how necessary it is to let students sign off every exercise. As can be clearly seen, it takes teaching assistants almost two days in total just to sign off one week of exercises (Week 7). The total signing off time spans many hours that could have also been used to help students with questions.

Another interesting point is the few exercises that take so much time to answer. In this example we see that many of the categories that have a long time to answer are also those in which the most questions have been asked. Looking further than 10 results it is also possible to see more questions on the same topic such as rank 11, where the Producer-Consumer pattern takes a total time of just over 2 hours. Combined with number 9, the total time spent explaining the Producer-Consumer pattern in more detail is over 5 hours. This indicates there might be a need to explain this in more detail in a lecture or through another form.

Table 6. Time spent answering

Category	Time per entry	Amount	Est. total time
Questions -> Project	0:06:56	975	4 days, 16:40:00
Signoff -> Week 7	0:07:33	357	1 day, 20:55:21
Signoff -> Week 6	0:09:06	237	1 day, 11:56:42
Signoff -> Week 5	0:09:53	197	1 day, 8:27:01
Signoff -> Week 4	0:09:26	151	23:44:26
Questions -> Other	0:07:09	73	8:41:57
Signoff -> Week 3	0:12:32	22	4:35:44
Questions -> Exercise 4.08 -> Sorting	0:11:52	17	3:21:44
Questions -> Exercise 7.17 -> Producer-Consumer pattern	0:06:29	25	2:42:05
Questions -> Exercise 6.06 -> Error	0:09:19	15	2:19:45

7. DISCUSSION

When processing the data multiple limitations have occurred. The first being the impact of ill-formed and non-descriptive questions by students. This results not only in keywords being picked up such as 'help' in Table 5, but the data also becomes less accurate further down the list. When looking beyond the top 10 commonly mentioned keywords, Dutch words often occur as the NLTK toolkit sees these all as keywords. This is intended for words it does not recognize but can be a real limitations when students tend to use another language as well.

Another limitation is the assumption made on the length of answering a certain question. On a large number of questions this will most likely still give an accurate result, but when looking at categories in which less is asked, this data is no longer accurate. One way of resolving this problem is having a teaching assistant or student mark when the question has been answered properly.

8. CONCLUSION

From the data gathered from the tool TA-Help.me multiple points of interest can be identified. In this research, four main directions have been identified to extract points of interest from the dataset (**RQ1**), namely 'Rooms and questions per day and hour', 'Most popular categories', 'Commonly mentioned keywords' and 'Time spent answering'. To answer **RQ2**, if the dataset can be used to identify points of interest, it is necessary to look at the usefulness of the results and the usage of the different fields of the dataset used to achieve this.

By using the timing of questions it is possible to get a clear overview of how busy certain days and certain hours of the day were (**RQ2.3**). This can in term be useful to optimize the timing of these particular sessions. With the categories it is possible to retrieve the most popular categories which, if it has any outliers, can be a clear indication that a certain concept or exercise is too difficult or unclear (**RQ2.2**). Combining these categories with timing can be used to not only see if questions are frequently asked in a specific category, but also if answering these questions takes a long time. Lastly by analyzing the questions typed out by students it is possible to identify keywords within these questions. By grouping those keywords on occurrence it is possible to find problems and concept that occur over the time span of the entire course and might occur much too often (**RQ2.1**). With this overview an educator is then able to identify what might need to be explained in more details in lectures or coursework.

Taking a step back and looking at the results, each of the three main features of the dataset (question data, categories & timing) are used to identify points of interest. Within this research it is seen that the categories and timing combined as seen in Table 6 can give a more powerful overview than when used separately. Overall the data in the dataset can be very effectively used to identify points of interest for course design.

9. FUTURE WORK

Looking into the effect of ill-formed and nondescriptive questions on the results is a useful direction for future work. It might be possible these have little effect, but it is also certainly possible that when forcing students to ask better thought out questions, the analysis of the course would also improve.

Another useful direction would be applying different types of keyword extraction on the data, or applying the process displayed in this research to other similar datasets.

10. REFERENCES

[1] Y. Bi, K. Deng, and J. Cheng. A keyword-based method for measuring sentence similarity. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 379–380, New York, NY, USA, 2017. ACM.

[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[3] D. Bouchard. Lab activity question queue software. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '16, pages 351–351, New York, NY, USA, 2016. ACM.

[4] N. C. Brown and A. Altadmri. Investigating novice programming mistakes: Educator beliefs vs. student data. In *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ICER '14, pages 43–50, New York, NY, USA, 2014. ACM.

[5] C. Gao and B. Goda. Face-to-face, or online, that is the question. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, SIGITE '18, pages 60–60, New York, NY, USA, 2018. ACM.

[6] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, M. A. Rubio, J. Sheard, B. Skupas, J. Spacco, C. Szabo, and D. Toll. 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, pages 41–63, New York, NY, USA, 2015. ACM.

[7] L. Leppänen, J. Leinonen, P. Ihantola, and A. Hellas. 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, pages 4–12, Piscataway, NJ, USA, 2017. IEEE Press.

[8] S. Sadjadee. Reducing students' waiting time for assistance in programming laboratory sessions by using electronic queueing. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018, pages 358–359, New York, NY, USA, 2018. ACM.