

Learning Analytics for Atelier

Sophie Weidmann
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
s.weidmann@student.utwente.nl

ABSTRACT

Learning Analytics play an increasingly important factor in virtual learning environments. Learning Analytics can be defined as the collection and evaluation of gathered data, used to create structured profiles of learners or the environment. These profiles can be utilized to increase the learning success of the individual, or to improve the learning environment as a whole. It is essential to adapt the Learning Analytics to the respective virtual environment in such a way that it offers the most valuable insights. This research provides insights into the effectiveness of Learning Analytics and will be tested on an existing learning environment called Atelier [6]. The outcome of this research will be an extension for Atelier that implements Learning Analytics. The extension will then be used to evaluate the current effectiveness and use of Atelier. It will also allow for further research into the factors that are most effective in improving the learning environment.

Keywords

Learning Analytics, Programming, Feedback, Atelier

1. INTRODUCTION

Virtual learning environments are becoming increasingly important in educational technology and can serve several purposes. They can be used for students to submit their assignments, receive feedback or communicate with other students or teachers. Learning Analytics can be implemented in these virtual learning environments to collect and interpret data that can provide new insights into the performance and effectiveness of the environment [9]. One of these learning environments is Atelier [6]. Atelier was developed for the University of Twente's Creative Technology (CreaTE) bachelor's program and supports teaching assistants and teachers in teaching core programming concepts to students. One of the main goals of Atelier is to facilitate collaboration and code sharing. Atelier allows students to submit their code, receive feedback and communicate with the teaching team.

The proposed research will explore Learning Analytics in the context of Atelier to enable the teaching team to examine the course and student performance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

35th Twente Student Conference on IT July 2nd, 2021, Enschede, The Netherlands.

Copyright 2018, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

1.1 Research Questions

The proposed research will address the following questions:

RQ1: Which Learning Analytics are considered the most promising in current research?

RQ2: How can Learning Analytics be integrated into the Atelier environment to achieve the greatest benefit for the teaching team?

RQ3: Does the inclusion of Learning Analytics enable new insights into the progress and development of students and the course as a whole?

RQ4: What long-term effect does the inclusion of Learning Analytics have on students and the course?

It is important to note that in order to answer RQ3, it is equally important to look at the impact on student development as well as the well-being of the course as a whole. Sub-questions to be answered include "What mistakes are made more often?", "How effective is the feedback given?" and "How active is the teaching team and the students as a group?".

RQ4 is outside the scope of this research as the impact of integrating Learning Analytics would need to be monitored over a longer period of time. However, the aim of this research is to enable further research in relation to this aspect, so the question is included as a prospect.

1.2 Methodology

This research includes several steps to answer the research questions. First, a literature review of the current state of Learning Analytics is performed. The aim of this literature review is to establish a basic understanding of current concepts and practices, which will then be used as a basis for answering RQ2.

Secondly, design research will be conducted to answer RQ2. The design research will consist of developing an extension for the Atelier environment. The design and implementation will be based on the results found in RQ1 and will be carried out in close collaboration with the stakeholders. Stakeholders in this research include the teaching team that uses Atelier in their courses. In the design research, a prototype is first created to answer RQ3. The prototype will then be further developed with the intermediate results of RQ3.

To answer RQ3, the prototype created in RQ2 will be applied to data collected in old courses where Atelier was used. This will show whether the extension works as expected and provides the right insights. Once the prototype is completed, RQ2 and RQ3 will be worked on simultaneously.

Finally, to answer RQ4, observational research would be necessary. However, answering this question is not possible for this research, as the long-term effects of the ex-

tension would have to be observed over a longer period of time.

2. BACKGROUND

This research focuses on a tool developed for the University of Twente's bachelor programme Creative Technology (CreaTe). CreaTe's programme involves teaching core programming concepts to first-year students using the Processing programming language. In addition to teaching coding in general, special emphasis is placed on teaching object-oriented programming. Object-oriented programming is based on the concept of classes and objects.

Create offers students the freedom to define and realise their own programming projects. For this reason, there is no blueprint solution with which the teaching team can match the solutions of the projects. Rather, the teaching team must review each code snippet themselves to assess whether the student has correctly understood and applied the programming concepts. A course may have more than 100 students, so manually reviewing each code snippet can be tedious. Atelier was developed to address this problem and support the manual correction of code.

2.1 Atelier

Atelier is a virtual learning environment developed for the bachelor's degree programme Creative Technology at the University of Twente and first used in 2020. Atelier is available as an open source project and is hosted on GitHub¹. The teaching team can create virtual courses on Atelier that mirror the real courses on campus. Students enrolled in the virtual courses can upload their projects written in the Processing programming language. The teaching team can then provide feedback in the form of comments on the code and communicate with students about "code smells" or other problems. The comments on Atelier can be made visible to the student or remain invisible.

2.1.1 Processing

Processing² is a programming language that was introduced in 2001 and finds its use in teaching non-programmers the core-concepts of computer programming in a visual arts context. Processing is based on the programming language Java, but introduced several simplifications, such as that variables cannot be declared as private or public. The CreaTe teaching team nevertheless uses the language to teach students object-oriented programming concepts.

2.1.2 Zita

To further simplify the inspection of individual code snippets, an extension for Atelier called Zita was developed. Zita is based on PMD³, a tool for the automatic analysis of code written in Java. Zita is currently analysing the code for 28 error types (See Appendix C for a complete list). 8 of these 28 error types have been customised and are not based on PMD's predefined errors. 4 of the 8 custom error types are only relevant for code written in Processing and would not be applicable to Java code. When Zita is activated, the code uploaded by the student is analysed and Zita creates comments pointing out the errors made. These comments are initially invisible to the student. The teaching team can decide whether to make them visible or not.

¹<https://github.com/creativeprogrammingatelier/atelier>

²<https://processing.org/>

³<https://pmd.github.io/>

2.2 Problem statement

Although Atelier is able to provide automated feedback on students' code, which facilitates the process of assessing student projects, it cannot display statistics or performance metrics related to students and the course. This makes it difficult to understand to what extent this virtual learning environment is beneficial for the students and the course. The aim of Atelier is to improve students' programming skills and ease communication between students and the teaching team. The two main objectives can be formulated as follows:

Programming skills By using Atelier, the teaching team anticipates that the students will improve their programming skills based on the feedback they receive. This means that the re-occurrence of the same errors over time should be minimised. In addition, the teaching team is particularly interested in teaching the students the concept of object-oriented programming. The two error types `UseUtilityClass` and `StatelessClass` are indicators of object-oriented code. Hence, they should occur the least or not at all.

Communication and Feedback The teaching team expects that the introduction of Atelier will facilitate and improve the feedback cycle and communication between the students and the teaching team. An indication of this is the number of comments that are made, by whom, the length and whether they are automated or not. If only the teaching team makes comments, it is clear that students are not using Atelier as intended.

2.3 Learning Analytics

A first approach to address these two objectives is to examine the current state of Atelier. This can be achieved by integrating Learning Analytics. Learning Analytics describe the analysis and representation of student behaviour. This enables an assessment of the progress of the whole course and gives the teaching team the opportunity to understand the impact of their teaching and thus improve the learning journey of the students. [3].

3. RELATED WORK

In order to answer the RQ1 a literature review was performed. To find relevant literature to this research field Google Scholar, Scopus and IEEE were used. Several scientific articles could be found by using search terms such as "Learning Analytics" and "Programming".

A lot of research in learning analytics related to virtual learning environments can be found. Much of this research explores what Learning Analytics are and researches its areas of application [2, 5, 7, 1, 3].

3.1 Objective Measurement

One area of application is explained by Phillips et al. [9], who researched the use of Learning Analytics in order to provide key indicators of students' behaviour in technology-enhanced environments. The outcome of this research is a learning-analytic tool, that observes students' behaviour through gathered data. This tool is an objective approach to measure students' learning behaviour since it does not rely on educators' subjective opinion.

3.2 Educational Practices

Ihantola et al. [8] performed research on educational data mining and Learning Analytics. This research discusses

the current state of the art in collecting and sharing programming data and presents three case studies that are using programming data for Learning Analytics. Ihantola et al. conclude that a challenge of the field of Learning Analytics is generalization. They found that very few studies build their analysis methods on a specific theory, model or educational practices. Instead the concrete incorporation of Learning Analytics depends on the learning environment and the values of the stakeholders.

3.3 Collaborative Learning

Another area of application is elaborated by Dascal et al. [4], where the analysis of collaborative learning is discussed. By examining collaborative learning, it is possible to see how active users with each other. For this purpose, a cohesion network analysis is carried out, which enables the identification of the learners' interaction patterns. Data on the content of the discourse and the interaction of the participants are collected and analysed. The result is a sociogram that reflects the interaction between the participants. The ReaderBench framework⁴ was used for the study, which can provide the automated assessment. This framework was also considered for this research, but is currently not functional.

4. LEARNING ANALYTICS

To answer RQ2 and Rq3, an extension was created that integrates Learning Analytics into the Atelier environment and provides insights into the behaviour of students and courses. The extension was created in close collaboration with the stakeholders, who communicated their requirements as well as the metrics in which they have the greatest interest.

4.1 Data Set

The dataset used for this research was extracted from the Ateliers database and comprises four modules, starting with Module 4 from 2020 and ending with Module 4 of 2021, which is ongoing at the time of writing. A table of the four courses can be found in Appendix B. The two M4 courses from 2020 and 2021 called Algorithms in Creative Technology are the same course and comparing them can give a good indication of whether differences in metrics are due to the course itself or the time that has passed since Atelier was introduced.

4.2 Data Analysis

In order to investigate the state of the two main objectives stated in **2.2 Problem statement**, two different aspects needed to be analysed that required different data from the data sets.

Interaction The data that is used to determine the interaction of the users with Atelier and with each other includes data about users, submissions, commentThreads and comments, their length, visibility and automation.

Error Types The information needed to extract the error types includes data on users and comments.

4.2.1 Submissions

To see the general interaction of students with the platform, the upload frequency of the submissions had to be examined. This metric is based on the submission table of the database. The implementation calculates the total

number of submissions made and the number of submissions per user and per file. These numbers can be filtered daily, weekly and monthly as well as per weekday.

4.2.2 Comments

The comments show how engaged the teaching team and students are in communicating with each other. This metric is based on the comment and comment thread tables extracted from the database. The comments are further classified into following three categories.

Automated vs. Non-automated Comments The extension calculates the total number of comments, and compares the number of automated and non-automated comments.

Zita Comments This computation shows all comments that were automatically generated by Zita. It also computes the number of Zita comments that were made visible.

Length of Comments For this computation, the non-automated comments are extracted and divided into short and long comments. This distinction was made because short comments are often just a mention of another person. Students may work together on tasks and then often only mention their group partner in a comment. Longer comments indicate that the user is putting more effort into the feedback. Short comments have less than 23 characters. This number was found by evaluating all comments by length and finding the threshold at which comments had almost no partner mentions. Furthermore, this calculation differs between student and teacher team comments.

The results of the computations can be filtered daily, weekly and monthly as well as per weekday.

4.2.3 Error Types

The metric of error types is based on the Zita extension, which generates automated comments with feedback. The extension extracts all course comments from the database and uses pattern matching to categorise the error types. The extension provides the ability to filter the distribution of errors based on user submissions, project submissions or all file submissions. In addition, the extension calculates the absolute number and percentage in relation to the total number as well as the distribution on a weekly basis.

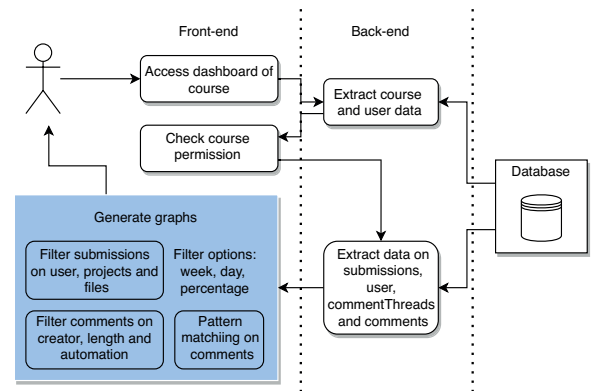


Figure 1. Key implementation parts

4.3 Implementation

The extension was implemented in the Atelier project. The extension is written exclusively in the front-end of

⁴<http://readerbench.com/demo/community>

Atelier. The results of the computations are displayed in several diagrams that are integrated into a dashboard. The dashboard is only accessible to the teaching team of each course.

Figure 1 shows the key implementation parts of the extension. The extension first checks which course the user is in and whether they are part of the teaching team. It then extracts data from the courses and users, including submissions and comments. The comments are then categorised based on automation, visibility, length and author. All automated comments are further analysed to identify error types. When a filter is selected, the data is further processed based on that filter.

4.4 Limitations of Implementation

There are a few limitations with this implementation. Currently, the database does not contain an error type field for the automated comments. As an alternative, pattern matching must be used to categorise the comments into error types. This technique is computationally expensive. The implementation does not currently explore explicit interaction between students and the teaching team. Comment threads could be analysed to investigate who is communicating with whom.

Mentions of other group partners could be filtered out of the database so that metrics for short and long comments are not affected.

Currently, only the entire course history is analysed. The individual course participant could also be analysed so that the teaching team can target the needs of individual course participants.

5. RESULTS

The following results mainly focus on the three modules "We Create Identity" (M1) with 133 students, "Algorithms in Creative Technology" (M4 2020) with 78 students, and "Algorithms in Creative Technology" (M4 2021) with 142 students. "Smart Environments" was largely omitted because the course only provides data from two weeks. Also, the results for M1 and M4 2021 are only from the first 5 weeks. M1 only provides data from 5 weeks and at the time of writing only data from the first 5 weeks of M4 2021 is available as this course is still running.

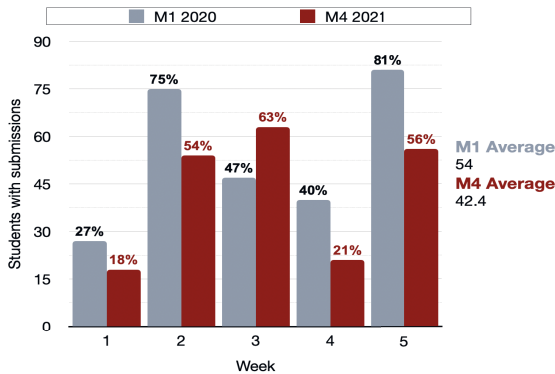


Figure 2. Number of Submission in Relation to Number of Users

5.1 Activity level and Interaction

The first point that is interesting to investigate and gives a good indication of the effectiveness of Atelier is the activity level of the students and the teaching team. The level of activity is measured by the number of submissions, the

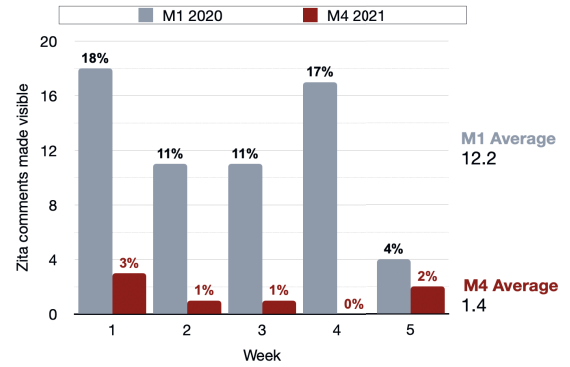


Figure 3. Number of Visible Zita Comments in Relation to Total Zita Comments in M1 & M4 2021

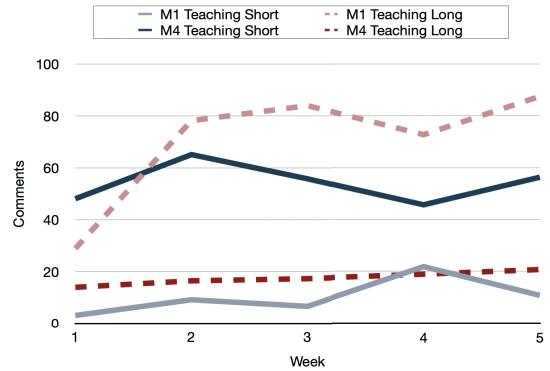


Figure 4. Number of Long and Short Comments of Teaching Team in M1 & M4 2021

number and type of comments that were made visible and the length of the comments.

5.1.1 Submissions

Figure 2 shows the percentage of submissions based on the total number of students enrolled. Submissions are split between weeks and grouped by student, so even if a student has made multiple submissions in a week, those submissions are only counted as one. In this way it is possible to see how many students were active in each week. As can be seen in Figure 2, 100% of the submissions are never reached. Students can work in groups of two, with only one partner uploading the solution to Atelier and mentioning their group partner in a comment. Assuming that students worked in pairs each week, the number of submissions should average 50%. In M1 it was not yet mandatory to use Atelier, however in M4 2021 it was mandatory for students to submit on a weekly basis. Figure 2 shows that M1 had more students with submissions than M4 2021. The students with submissions averages for M1 to 54%, and for M4 2021 to 42.4%.

5.1.2 Use of Zita

This metric shows how actively the Zita extension is used by the teaching team to give students a reflection of their errors. Figure 3 shows the percentage of visible Zita comments compared to the total number of comments generated by Zita. On average 12.2% of Zita generated comments were made visible in M1, 1.4% in M4 2021 and 14.875% in M4 2020.

5.1.3 Non-automated Comments

This metric investigates the number of manual comments

and their length. Figure 4 shows the number of comments categorised by short and long comments of the teaching team for M1 and M4 2020. Figure A.8 for M1 and Figure A.9 for M4 2020 show the total numbers of comments categorised by short and long comments from the teaching team and the students. In Figure A.8 it can be seen that the teaching team wrote significantly longer comments than shorter comments in M1. Furthermore, student comments were mainly limited to the first week of the course. Figure A.9 shows that more short comments than long comments were written in M4 2020. The distribution of student comments is almost evenly distributed over the weeks for M4 2020.

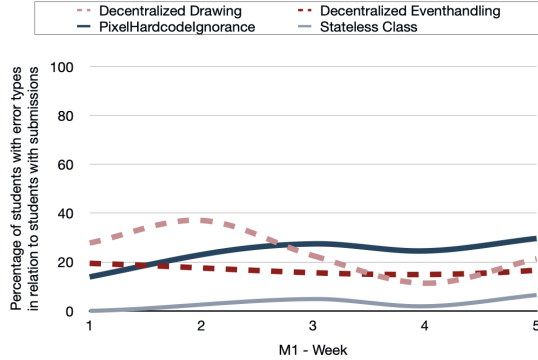


Figure 5. Occurrence of Error Types in Relation to Students with Submissions in M1

5.2 Re-occurrence of Errors

Another metric that indicates how effective Atelier is, is the frequency of re-occurring error types. Figure 5 is based on M1 and shows the percentage of students with error types in relation to students with submissions on a weekly basis for four error types. Three of the four error types are exclusive for Processing and not based on PMD. The fourth Processing-exclusive error "OutOfScopeStateChange" is left out of the evaluation because its definition has changed across courses and the number of occurrences is therefore unstable. "StatelessClass" was taken into account because this error type gives an indication of whether object-oriented concepts are included. "UseUtilityClass" would provide the same indication, but this error never appeared in the evaluated datasets.

Until week 3, Decentralised Drawing is the most frequent error type, which then changes to PixelHardcoreIgnorance. The stateless class is the least frequent error type. As can be seen in Figure 5, all error types have a decrease in occurrence in week 3 and 4. However, the occurrence for all error types increases again after week 4. Comparing the number of error types to the curve of the students with submissions, shows that the curve of reduced error types correlates with the curve of the number of submissions grouped by students.

Figure A.6 shows the number of students that made an error on a weekly basis for M4 2020. The number of errors correlates again with the number of submissions. It is also notable that in week 7, every error type still appears. In week 7 the Stateless Class had also the highest count. Figure A.7 shows the same metric for the first 5 weeks of M4 2021. It can be seen that almost none Decentralised Drawing and PixelHardcoreIgnorance errors were made in M4 2021. Furthermore, the PixelHardcoreIgnorance error of M4 2020 and the Decentralised EventHandling error of M4 2021 have the highest frequencies and show a similar

pattern in weeks 2 and 3. In general, Figure A.7 shows that altogether less errors were made in M4 2021 compared to M1 2020. The total numbers can be found in Figure A.10 (M1), Figure A.11 (M4 2020) and Figure A.12 (M4 2021).

5.3 Comparison of Courses

M1 and M4 2021 show some differences in performance. The number of Zita comments made visible was higher in M1. To see if this difference is caused by the time or by the course difference, M4 2021 of Figure 3 can be compared to Figure A.13. Figure A.13 shows the number of visibly made Zita comments for M4 2020. As can be seen the average number for M4 2020 is at 14.875% and the average number for M4 2021 is at 1.4%.

Figure 4 shows the comments of M1 and M4 2021 for the teaching team. As can be seen, the number of short and long comments between M1 and M4 has reversed almost completely.

M2 only ran for 2 weeks, yet some findings can be made which show that longer comments (84) were made more often than short ones (4) by the teaching team. A total of 90 comments were made, of which only 2 were from students, so no student interaction took place in this course.

5.4 Extension of Atelier

The result of the implementation presented in this paper is an extension that can be integrated into Atelier and will display metrics for all old and future courses. The extension will display the data in the form of a dashboard in the course view of Atelier. Permission to access the dashboard is currently only given to members of the teaching team.

6. DISCUSSION

One of the key findings from this research is that Learning Analytics need to be adapted to the learning environment. As Ihantola et al. [8] stated, there are no guidelines on how Learning Analytics should be implemented. The Learning Analytics for Atelier were implemented in close collaboration with the stakeholders of the project and provide some key indicators of student and course performance.

It was found that the modules used with Atelier performed differently in the number of submissions, length of comments and visible Zita comments. In general, M1 had a higher level of activity and more interaction than M4 2021. In M4 2021 it was compulsory for students to use Atelier, however the average number of students with submissions in M4 2021 is 11.8% lower than the average of M1 and below the threshold of 50%. This indicates that not all students submitted their projects on time and missed some deadlines.

Furthermore, the number of visible made Zita comments sank from 12.2% (M1) to 1.4% (M4 2021). This suggests that the teaching team scarcely took the time to go through the proposed comments in order to make them visible. A hypothesis could be that the teaching team made more Zita comments visible in M1, because Zita was just newly introduced. An indicator that supports this thesis is, that the 2020 edition of the M4 course had an average of 14.875% visible made comments, 13.475% more than the 2021 edition of the same course. The M4 2021 course has still three weeks left, at the time of writing, but comparing the first five weeks of Figure 3 and Figure A.13 show a significant difference in numbers. This finding suggests that the difference in comments made visible does not correlate with the courses themselves, but with the amount of time since Zita was introduced. However,

further observations would have to be made over a longer period of time to support this thesis.

In addition, it was found that the length of the teacher team's comments changed in M1 and M4 2021. This suggests that the teaching team took less time to formulate longer feedback, although longer feedback can provide more detail to students and is therefore more helpful. However, one reason for this could be that M4 2021 includes a tag system and M1 does not. Tags are short keywords introduced by a "#". All tags currently fall under the category of short comments.

A last point of discussion are the re-occurrence of error types. A clear reduction of the error types cannot be seen in Figure 5, Figure A.6 or Figure A.12. If the number of students who made a type of error is lower, the number of students who made a submission is also lower, suggesting that the two numbers are correlated. It could be speculated whether this is because the Zita comments indicating these errors are not made visible. However, the average number of visible Zita comments is 2.675% higher in M4 2020 than in M1, but Figure A.11 shows that at the end of the course, all error types are also still appearing in M4 2020, and that the Stateless Class error has the highest count here. Since the Stateless Class error indicates whether a student has understood the concept of object-oriented programming, it seems that students have not fully grasped the concept by the end of the course. However, it is also important to note that students learn more about the concepts as the course progresses, so some errors may not appear at the beginning and are more noticeable at the end. This would explain why no Stateless Class error was made in the first week of M1 and M4 2020.

7. CONCLUSION

This research explored what Learning Analytics are and what it takes to incorporate them. The result is an extension for the Atelier virtual learning environment.

Learning Analytics provide an objective measure of student course behaviour. Incorporating Learning Analytics into an extension of Atelier provides new insights into the effectiveness and user behaviour of Atelier. It was found that the comments generated by Zita are not used as much as they should be and that the teaching team reduced their overall activity from M1 to M4 2021. Furthermore, no significant reduction in the re-occurrence of the same errors was observed.

7.1 Future Work

The future benefit of this extension includes that the teaching team can see in real time how their courses are performing. They can see how effective their teaching is. If they see that certain mistakes are still being made in a week, they can adjust their teaching to educate the students more about these issues. They can also see how active the students and the teaching team are. If they see that almost no Zita comments are made visible, or that mostly short comments are written, they can discuss the cause with the other members of the teaching team and adjust for the coming weeks.

To analyse collaborative learning for Atelier, the Readerbench framework could be used to analyse who is communicating with whom based on the comment threads. Currently the Readerbench framework is not ready for use, but it could be integrated in the future.

Finally, to answer RQ4, a long-term analysis of the courses and the use of Learning Analytics should be conducted. This analysis should answer whether the implementation

can improve the efficiency of teaching by quickly adapting to various scenarios.

8. REFERENCES

- [1] M. Berland, T. Martin, T. Benton, C. P. Smith, and D. Davis. Using learning analytics to understand the learning pathways of novice programmers. *Journal of the Learning Sciences*, 22(4):564–599, 2013.
- [2] D. Boulanger, J. Seanosky, V. Kumar, Kinshuk, K. Panneerselvam, and T. S. Somasundaram. Smart learning analytics. In: *Chen G., Kumar V., Kinshuk, Huang R., Kong S. (eds) Emerging Issues in Smart Learning. Lecture Notes in Educational Technology.*, pages 289–296, 2015.
- [3] D. Clow. An overview of learning analytics. *Teaching in Higher Education*, 18(6):683–695, 2013.
- [4] M. Dascalu, M.-D. Sirbu, G. Gutu-Robu, S. Ruseti, S. A. Crossley, and S. Trausan-Matu. Cohesion-centered analysis of sociograms for online communities and courses using readerbench. In V. Pammer-Schindler, M. Pérez-Sanagustín, H. Drachsler, R. Elferink, and M. Scheffel, editors, *Lifelong Technology-Enhanced Learning*, pages 622–626, Cham, 2018. Springer International Publishing.
- [5] E. Er, C. Villa-Torrano, Y. Dimitriadis, M. Gasevic, D.bEmail Bote-Lorenzo, J. Asensio-Pérez, E. Gómez-Sánchez, and A. Martínez Monés. Theory-based learning analytics to explore student engagement patterns in a peer review activity. *11th International Conference on Learning Analytics and Knowledge: The Impact we Make: The Contributions of Learning Analytics to Learning, LAK 2021; Virtual, Online; United States*, pages 196–206, April 2021.
- [6] A. Fehnker and A. Mader. Atelier for creative programming. May 2020. 12th International Conference on Computer Supported Education, CSEDU 2020, CSEDU ; Conference date: 02-05-2020 Through 04-05-2020.
- [7] I. Hilliger, C. Miranda, G. Schuit, F. Duarte, M. Anselmo, and D. Parra. Evaluating a learning analytics dashboard to visualize student self-reports of time-on-task: A case study in a latin american university. *11th International Conference on Learning Analytics and Knowledge: The Impact we Make: The Contributions of Learning Analytics to Learning, LAK 2021; Virtual, Online; United States*, pages 592–598, April 2021.
- [8] 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. page 41–63, 2015.
- [9] R. Phillips, D. Maor, G. Preston, and W. M. Cumming-Potvin. Exploring learning analytics as indicators of study behaviour. *Proceedings of World Conference on Multimedia, Hypermedia and Telecommunications 2012, Chesapeake V.A.*, pages 592–598, January 2012.

APPENDIX

A.

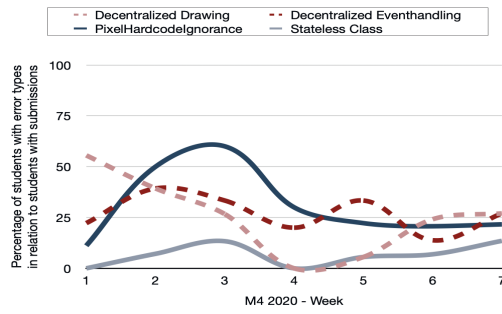


Figure A.6. Occurrence of Error Types in Relation to Students with Submissions in M4 2020

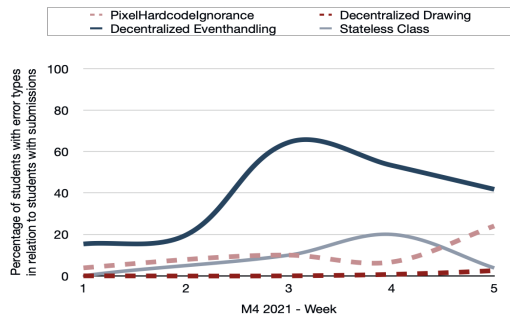


Figure A.7. Occurrence of Error Types in Relation to Students with Submissions in M4 2021

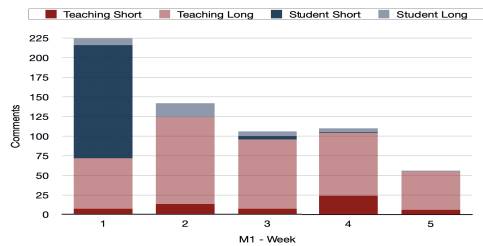


Figure A.8. Number of Long and Short Comments of Teaching Team in M1

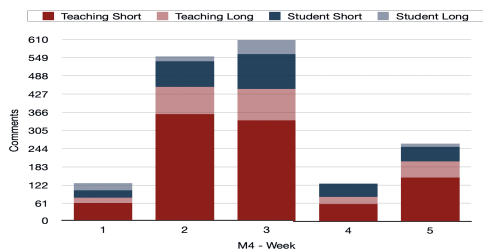


Figure A.9. Number of Long and Short Comments of Teaching Team in M4 2021

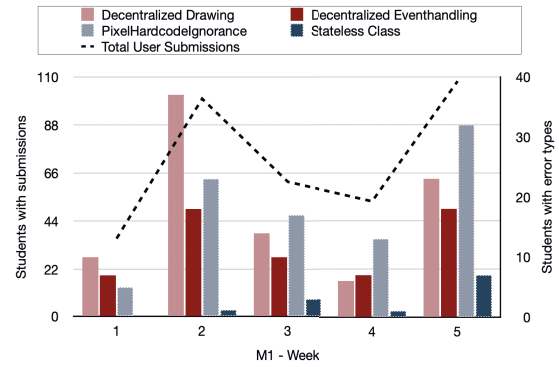


Figure A.10. Occurrence of Error Types in M1

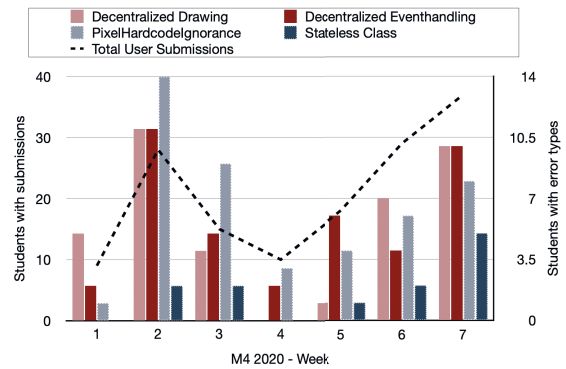


Figure A.11. Occurrence of Error Types in M4 2020

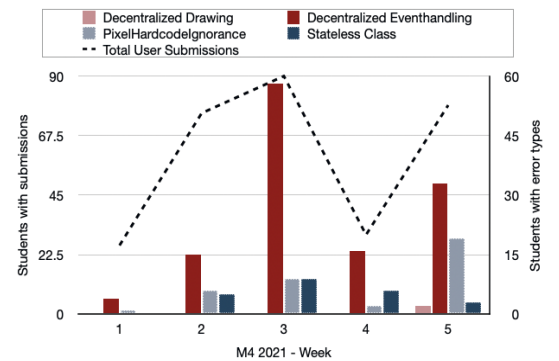


Figure A.12. Occurrence of Error Types in M4 2021

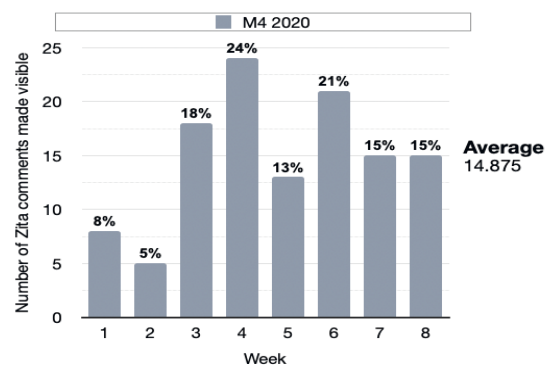


Figure A.13. Number of Visible Zita Comments in Relation to Total Zita Comments in M4 2020

B. ATELIER COURSES

Table 1. Dataset of modules in Atelier

Name	Module	Year	No. of students	Time span
Algorithms in Creative Technology	M4	2021	142	6 Weeks
Algorithms in Creative Technology	M4	2020	78	8 Weeks
Smart Environments	M2	2020	65	2 Weeks
We Create Identity	M1	2020	133	5 Weeks

C. ERROR TYPES

Table 2. Dataset of modules in Atelier

Error type	Custom
AddEmptyString	No
AssignmentInOperand	No
AtLeastOneConstructor	No
AvoidDeeplyNestedIfStmts	No
AvoidFieldNameMatchingMethodName	No
AvoidFieldNameMatchingTypeName	No
AvoidReassigningParameters	No
ClassNamingConventions	No
ControlStatementBraces	No
CyclomaticComplexity	No
DecentralizedDrawing	Yes
DecentralizedEventHandling	Yes
EmptyIfStmt	No
EmptyStatementNotInLoop	No
FieldNamingConventions	No
FormalParameterNamingConventions	No
GodClass	Yes
IdempotentOperations	No
LocalVariableNamingConventions	No
LongMethod	Yes
LongParameterList	Yes
LongVariable	No
MethodNamingConventions	No
OutOfScopeStateChange	Yes
PixelHardcodeIgnorance	Yes
ShortMethodName	No
ShortVariable	No
SimplifyBooleanExpressions	No
SingularField	No
StatelessClass	Yes
TooManyFields	No
UncommentedEmptyConstructor	No
UncommentedEmptyMethodBody	No
UnconditionalIfStatement	No
UnusedFormalParameter	No
UnusedLocalVariable	No
UseUtilityClass	No