

Learning Analytics for Moodle: Facilitating the Adoption of Data Privacy Friendly Learning Analytics in Higher Education

Von der Fakultät für Bauingenieurwesen der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines Doktors
der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Thomas Dondorf

Berichter: Universitätsprofessor Dr.-Ing. Heribert Nacken
 Privatdozent Dr. phil. Malte Persike

Tag der mündlichen Prüfung: 19. April 2022

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Abstract

Learning analytics refers to the analysis of data generated in learning environments with the aim of supporting and improving teaching and learning. Due to the increasing availability of large amounts of educational data and possibilities to process them, there has been a growing interest in the topic in recent years. Long term, learning analytics could become a method to improve teaching methods based on data-driven evidence and to give students the opportunity to reflect on their learning behavior.

In Germany, the adoption of learning analytics is scarce. Strict data privacy regulations and user concerns pose challenges for researchers and developers. Many lecturers do not even have access to simple descriptive usage statistics for their digital course rooms. Few efforts are being made regarding the practical integration into teaching. While the COVID-19 pandemic has reinforced the trend towards digital teaching, suitable tools for collecting and processing data from learning environments are still missing.

This dissertation examines how learning analytics can be integrated into learning environments in practice. This thesis covers Moodle, a course management system widely used at German universities. Based on related work, a data privacy-friendly solution for learning analytics is developed that can be adopted by other universities and is able to cover future use cases.

As result, a software is presented that does not store personalized data. During the development phase of this dissertation, the implemented software gained interest at other universities. At the time of writing of this thesis, it is used at six universities. Several universities were also involved in the evaluation. Two longitudinal case studies based on user surveys conclude that the chosen approach can successfully support lecturers and their teaching. The technical evaluation shows that the integration into Moodle is successful in terms of scalability, extensibility, and transferability. The results of this dissertation can be used by universities as ready-to-use introduction to learning analytics.

Kurzfassung

Learning Analytics umfasst die Analyse von Daten, die in Lernumgebungen anfallen, mit dem Ziel, Lehr- und Lernprozesse zu unterstützen und zu verbessern. Durch die steigende Verfügbarkeit großer Datenmengen im Bildungsbereich und die immer leichter werdenden Möglichkeiten selbige zu verarbeiten, hat es in den letzten Jahren ein zunehmendes Interesse an dem Thema gegeben. Dabei könnte Learning Analytics sich langfristig als Methode etablieren, um basierend auf datengestützter Evidenz, Lehrmethoden zu verbessern und Studierenden die Möglichkeit zu geben ihr Lernverhalten zu reflektieren.

In Deutschland ist der Einsatz von Learning Analytics noch wenig verbreitet. Strenge Datenschutzgesetze sowie Bedenken der Anwender stellen Forscher und Entwickler bei der Implementierung vor Herausforderungen. Vielen Lehrkräften fehlt eine Möglichkeit einfache deskriptive Zugriffsstatistiken in ihren digitalen Kursräumen abzurufen. Nur vereinzelt werden Maßnahmen zur praktischen Integration von Learning Analytics in den Hochschullalltag entwickelt. Während die COVID-19-Pandemie den Trend hin zu digitaler Lehre verstärkt hat, fehlen weiterhin die Werkzeuge zur Erhebung und Analyse von Daten aus Lernumgebungen.

In dieser Arbeit wird daher untersucht, wie Learning Analytics in der Praxis in Lernumgebungen integriert werden kann. Als Anwendungsbeispiel wird hierfür das Lernmanagementsystem Moodle genutzt, welches an deutschen Hochschulen weit verbreitet ist. Aufbauend auf existierenden Arbeiten, wird eine datenschutzfreundliche Lösung für Learning Analytics entwickelt, die von anderen Hochschulen genutzt werden kann und auch zukünftige Anwendungsfälle berücksichtigt.

Im Ergebnis wird eine Software-Lösung vorgestellt, die keine personalisierten Daten speichert. Bereits während der Entwicklungsphase stießen die Ergebnisse auf Interesse an anderen Hochschulen. Zum Zeitpunkt der Erstellung dieser Arbeit wird die Software an sechs Hochschulen eingesetzt. Auch an der Evaluation waren mehrere Hochschulen beteiligt. Zwei begleitende Fallstudien, basierend auf Nutzerbefragungen, kommen zu dem Schluss, dass der gewählte Ansatz eine Unterstützung der Lehrenden und ihrer Lehre erreicht. Die technische Evaluation der Arbeit zeigt, dass die Integration in Moodle hinsichtlich Skalierbarkeit, Erweiterbarkeit und Übertragbarkeit erfolgreich ist. Die Ergebnisse dieser Arbeit können von Hochschulen als einsatzbereiter Einstieg in Learning Analytics genutzt werden.

Acknowledgments

I would like to thank everyone who supported me in the process of creating this dissertation.

First, I would like to express my deep gratitude to my supervisor Prof. Dr.-Ing. Heribert Nacken. I am very thankful for his support, advice, and feedback. His guidance helped me in the entire time of research and writing of this thesis. Many thanks for this opportunity.

Furthermore, I want to thank PD Dr. phil. Malte Persike. His support and feedback helped me to complete this dissertation.

I would also like to thank Prof. Dr. rer. nat. Matthias Müller for supporting this research early on and for his valuable feedback.

In addition, I would like to thank the Moodle operators and developers from the IT Center for deploying and operating the developed software. Thanks to Bernd Decker, Peter Schartner, Tim Schroeder, and Rahel Phasen.

Likewise, I would like to thank the team from the Center for Learning Services. Thanks to Harald Schnurbusch, Sebastian Schilling, and Ralph Delzepich for their feedback and support.

I would like to thank all students, apprentices, and developers who supported my work. Special thanks to Marcel Behrmann for his work in the early stages of development and to Thorben Laprell and Kasimir Timmers for their contributions.

In addition, I want to thank all early adopters from other universities who had the trust in me to run my software and contributed to the evaluation. Special thanks to Annamaria Köster and Jonathan Liebers for their support.

Furthermore, I thank my colleagues. Thanks to Prof. Dr.-Ing. Hani Sewilam for inspiring discussions and feedback. Thanks to Richard Gramlich and Leon Filser for supportive conversations and proofreading. And thanks to the whole team for a great time.

I would also like to sincerely thank my family for their support. Thanks to my parents for raising me and to my sisters for their encouragement.

Last but not least, I thank Diana for her love, encouragement, support, and endless patience.

Contents

1 Introduction	1
1.1 Motivation	1
1.2 Objectives.....	2
1.3 Outline	3
2 Background	5
2.1 Learning Analytics.....	5
2.1.1 Definition	5
2.1.2 Related Research Fields.....	6
2.1.3 Initiatives and Networks	11
2.1.4 Process Models and Concepts	13
2.2 Moodle	18
2.2.1 Adoption and Usage in Higher Education	19
2.2.2 Moodle at RWTH Aachen University	20
2.2.3 Plugins.....	21
2.3 Data Privacy	21
2.3.1 Anonymization and Pseudonymization.....	22
2.3.2 General Data Protection Regulation.....	22
2.3.3 Situation in Germany	23
2.3.4 Ethics	24
3 Related Work	27
3.1 Learning Analytics Architectures.....	27
3.1.1 Apereo Learning Analytics Initiative	27
3.1.2 Jisc Open Learning Analytics	29
3.1.3 Notable Approaches.....	30
3.1.4 Summary	31
3.2 Learning Analytics Tools.....	31
3.2.1 Learning Analytics Projects.....	31
3.2.2 Generic Analytics Tools	34
3.2.3 Learning Record Stores.....	35
3.3 Learning Analytics for Moodle.....	37

3.3.1 Core Functionality	38
3.3.2 Report Plugins.....	40
3.3.3 Generic Plugins.....	41
3.3.4 Commercial Software.....	44
3.3.5 Implementations not integrated into Moodle	47
3.3.6 Implementations without published Source Code.....	48
3.3.7 Summary	51
3.4 Learning Analytics Evaluation.....	53
3.4.1 Evaluation through Expert Analysis	53
3.4.2 Evaluation through User Participation	54
3.4.3 Practice	57
4 Preliminary Work	59
4.1 Prototype 1: Data Sources and Processing	59
4.1.1 Moodle Logging and Events	60
4.1.2 Data Model.....	62
4.1.3 Implementation and Visualizations	64
4.1.4 Results.....	67
4.1.5 Evaluation	72
4.2 Prototype 2: Integration into Moodle	73
4.2.1 Data Model.....	73
4.2.2 Implementation and Visualization.....	73
4.2.3 Evaluation	77
4.3 Conclusion.....	78
5 Analysis	79
5.1 Analysis of existing Learning Analytics Tools.....	79
5.2 Lessons Learned from Prototypes	80
5.2.1 Learning Analytics inside the Learning Management System	80
5.2.2 Extensibility.....	81
5.2.3 No Personal Data Collection by Default.....	81
5.3 Learning Analytics as Service.....	81
5.3.1 Process	82
5.3.2 Institutional Policies.....	82
5.3.3 User Support.....	82
5.3.4 Infrastructure Requirements	83
5.4 Software Requirements	83
5.4.1 Functional Requirements.....	83
5.4.2 Non-functional Requirements.....	85
6 Implementation	89
6.1 Architecture	89

6.2	Data Logging.....	90
6.2.1	Data Model.....	92
6.2.2	Aggregation of Data	94
6.2.3	Import of Data.....	95
6.3	User Interface.....	95
6.3.1	Visualizations	97
6.3.2	Usability.....	100
6.3.3	Data Privacy and Ethics.....	101
6.3.4	Integration into Moodle.....	102
6.4	Operation	102
6.4.1	Administration.....	103
6.4.2	Documentation and Releases.....	105
6.5	Implemented Visualizations	106
6.5.1	Case Study 1: Summer Semester 2020.....	107
6.5.2	Case Study 2: Winter Semester 2020/21	111
6.6	Summary	115
7	Evaluation	117
7.1	Methodology	117
7.2	Data Analysis.....	118
7.2.1	RWTH Aachen University	119
7.2.2	Comparison with other Universities	123
7.3	Technical Evaluation.....	125
7.3.1	Scalability	125
7.3.2	Extensibility	129
7.3.3	Transferability.....	131
7.4	Case Study 1: Summer Semester 2020	132
7.4.1	Setting	133
7.4.2	Survey Results and Discussion	136
7.4.3	Findings.....	143
7.5	Case Study 2: Winter Semester 2020/21	144
7.5.1	Setting	144
7.5.2	Survey Results and Discussion	146
7.5.3	Findings.....	150
7.6	Summary	152
8	Conclusion	155
8.1	Summary	155
8.2	Adoption.....	156
8.3	Future Research and Development Directions	157
	Bibliography	159

List of Figures

Figure 1: The Learning Analytics Cycle (Clow, 2012).....	14
Figure 2: Collective application model (Dron & Anderson, 2009)	14
Figure 3: Learning Analytics Continuous improvement cycle (adapted from Elias, 2011).....	15
Figure 4: Learning Analytics process (Chatti et al., 2012).....	15
Figure 5: Learning Analytics process model (Verbert et al., 2013)	16
Figure 6: Learning Analytics model (Siemens, 2013)	17
Figure 7: Assignment of privacy principles to the reference model for learning analytics (adapted from Cho et al., 2016).....	17
Figure 8: Number of registered Moodle pages since 2003.....	18
Figure 9: Integration of Moodle into a typical university infrastructure (Hunt, 2012).....	19
Figure 10: Number of Moodle pages at German universities since 2010 (based on Steitz, 2020 and the corresponding Internet Archive page).....	20
Figure 11: Apereo Learning Analytics Strategic Vision for an Open Learning Analytics Platform (Apereo, 2016).....	28
Figure 12: Jisc Learning Analytics Open Architecture (Jisc, 2018a)	29
Figure 13: OU Analyse dashboard, part of the course overview page (Kuzilek et al., 2015)	33
Figure 14: Prediction of students at risk (Monllaó & Dalton, 2017)	38
Figure 15: Activity report of a Moodle course	40
Figure 16: IntelliBoard course dashboard (Kochnev, 2021).....	45
Figure 17: Activity over time visualized in LeMo	47
Figure 18: Course Overview page of SCELE (Santoso et al., 2018).....	49
Figure 19: Assessment of usability based on the SUS score (Bangor et al., 2009).....	57
Figure 20: Moodle Legacy Logstore	60
Figure 21: Moodle Standard Logstore	61
Figure 22: Contextualized Attention Metadata model (Schmitz et al., 2011)	62
Figure 23: Data model of the first prototype.....	64
Figure 24: User interface of the first prototype	65
Figure 25: Interface to select names and grades.....	66

Figure 26: Correlation between exam grade of the course and course activity	66
Figure 27: Visualized correlation of number of learning days, number of events, and exam grade (course 1)	68
Figure 28: Visualized correlation of number of learning days, number of events, and exam grade (course 2)	69
Figure 29: Visualization of average learning day, events, and exam grade.....	70
Figure 30: Visualization of student record data, learning days, and exam grade	71
Figure 31: Visualization of course accesses as Moodle block (course content pixelated).....	74
Figure 32: Maximized visualization.....	74
Figure 33: Visualization of student activity in the course room	75
Figure 34: Available visualizations	77
Figure 35: Overall architecture and integration into Moodle (as published in Dondorf et al., 2021)	90
Figure 36: Start page of the user interface.....	96
Figure 37: Micro Charts.....	96
Figure 38: Output of the demo plugin	99
Figure 39: Preview box in the course dashboard based on <i>preview.php</i>	99
Figure 40: Help page	100
Figure 41: Moodle user tour on the dashboard.....	101
Figure 42: Necessary threshold for the indicator is not reached.....	101
Figure 43: Learning analytics log in the list of Moodle logstores.....	103
Figure 44: Option to enable the plugin in the course settings page.....	104
Figure 45: Comparison of Moodle designs	107
Figure 46: Learning analytics dashboard page in a Moodle course.....	108
Figure 47: Dashboard (first case study)	109
Figure 48: Page <i>Participants</i> (first case study).....	109
Figure 49: Page <i>Browser and Operating System</i> (first case study).....	110
Figure 50: Page <i>Activities</i> (first case study)	111
Figure 51: Page <i>Dashboard</i> (second case study)	112
Figure 52: Page <i>Participants</i> (second case study)	113
Figure 53: Page <i>Heatmap</i> (second case study)	113
Figure 54: Page <i>Quizzes and Assignments</i> (second case study).....	114
Figure 55: Page <i>Activities</i> (second case study).....	115
Figure 56: Number of events in Moodle by weekday	119
Figure 57: Number of events in Moodle by hour	119
Figure 58: Heatmap of Moodle activity of students	120
Figure 59: Weekly distribution of teaching activities by the number of registered students and the weekly access pattern of students	121

Figure 60: Weekly aggregated number of events in Moodle over two semesters	122
Figure 61: Comparison of RWTH Aachen University and Hochschule Ruhr West regarding events per hour relative to their absolute daily peak.....	123
Figure 62: Comparison of RWTH Aachen University and GUTech regarding events per hour relative to their absolute peak.....	124
Figure 63: Comparison of activity of students by weekday between RWTH Aachen University and GUTech relative to their absolute peak	125
Figure 64: Integration of a custom subplugin for video views in the bottom left.....	130
Figure 65: Plugin page for videos developed by the IT Center	130

List of Tables

Table 1: Comparison of different types of analytics	7
Table 2: Comparison of Learning Analytics and Educational Data Mining (adapted from Siemens and Baker, 2012).....	9
Table 3: Comparison of Learning Analytics and Academic Analytics (Siemens & Long, 2011)	10
Table 4: Examples for Personalization, Pseudonymization, and Anonymization.....	22
Table 5: Comparison of analytics tools for Moodle.....	52
Table 6: Questionnaires measuring usability and quality of software	56
Table 7: Correlation coefficients and p-values (course 1).....	68
Table 8: Correlation coefficients and p-values (course 2).....	69
Table 9: Correlation coefficients and p-values	70
Table 10: Course sizes, correlation coefficients between exam grades and average learning times, and corresponding p-values	71
Table 11: Correlation coefficients and p-values	72
Table 12: Moodle standard logging database table	93
Table 13: Aggregation table of subplugin <i>lalog_browser_os</i>	94
Table 14: File statistics of the Moodle plugin <i>local_learning_analytics</i>	116
Table 15: File statistics of the Moodle plugin <i>logstore_lanalytics</i>	116
Table 16: Most common events in Moodle	122
Table 17: Database data storage	126
Table 18: Runtimes of database queries for selected courses.....	129
Table 19: Use of learning management systems at German universities	131
Table 20: Participants of the first case study.....	134
Table 21: Faculties of participating lecturers	136
Table 22: Course sizes of participants.....	137
Table 23: Activation of learning analytics service.....	137
Table 24: System Usability Scale results.....	138
Table 25: Usage frequency.....	139
Table 26: Time of use (multiple selections possible).....	139
Table 27: Rating of usefulness for each feature.....	140

Table 28: Responses regarding personalized analytics	140
Table 29: Overall rating	141
Table 30: Results of student evaluation	142
Table 31: Participants of the second case study	145
Table 32: Faculties of participating lecturers at RWTH Aachen University and Hochschule Ruhr West in the second case study	146
Table 33: Course sizes of participants	147
Table 34: System Usability Scale results	148
Table 35: Time of use (multiple selection possible)	149
Table 36: Rating of usefulness for each feature	149
Table 37: Overall rating	150
Table 38: Comparison of scores of features and their data sources	151
Table 39: Overview of case studies and participating universities	153

1 Introduction

Learning Analytics aims to use the increasingly large amounts of data like interaction data generated in online learning environments to understand and optimize learning and teaching processes. The multidisciplinary field of learning analytics closes the gap between learning and data sciences and incorporates a wide range of techniques to gather, store, and report data. In the past decade, learning analytics has gained a lot of attention. The 2011 Horizon Report (Johnson et al., 2011) stressed the potential impact of learning analytics expecting a widespread adoption in the following years. The most recent 2021 Horizon Report for teaching and learning named learning analytics as one of six key technologies (Pelletier et al., 2021). In Germany, learning analytics is only used cautiously. Many higher education institutions struggle to provide even basic descriptive course statistics due to data privacy concerns (Ifenthaler & Yau, 2019).

In the following sections, the main motivation behind this dissertation is discussed. Afterwards, the key research questions and methodologies used to answer these questions are presented. Finally, the overall structure of this thesis is outlined.

1.1 Motivation

In the last decades, the trend towards digital teaching, video-supported lectures or similar scenarios, often simply referred to as *blended learning* has made digital teaching environment and learning management systems an integral part of daily teaching activities. This trend was intensified by the COVID-19 pandemic in 2020 moving nearly all education online (Daniel, 2020) and leading to a rapid growth of interest in tools supporting digital teaching and learning (Kerres, 2020).

Even before the growing interest due to COVID-19, learning analytics has been perceived as a valuable addition to blended learning teaching scenarios (Persike, 2020). But while the digitalization of teaching and learning is increasing, the adoption of learning analytics in the higher education sector in Germany is scarce (Paul & Sharon, 2014; Williamson, 2017). Recent research trends in the field of learning analytics focus on machine learning (Spikol et al., 2018) or performance prediction of students (Moreno-Marcos et al., 2020). At the same time, higher education institutions in Germany struggle to provide basic descriptive statistics to lecturers and students due to data protection (Ifenthaler & Yau, 2019) and privacy concerns (Drachsler & Greller, 2016). The adoption of learning analytics is stagnating and has in some cases even been reversed following concerns of stakeholders regarding privacy and ethics over the handling of personal data (Drachsler & Greller, 2016). The 2020 Horizon Report notes *Analytics and Privacy*

Questions as global technological trend and calls for institutions to be more proactive in protecting students data (Brown et al., 2020). This dissertation recognizes the issue of data privacy and protection as a central component and makes it a key feature of the developed software implementation.

The field of learning analytics is becoming more and more of interest at RWTH Aachen University. In a prospect for teaching 2021, the vice-rector announced learning analytics and educational data mining as tools to increase student success and reduce dropouts (Krieg, 2020). Previously, the shift from degree-focused approaches towards individual support of students had been announced (Leicht-Scholten & Nacken, 2010) and the importance of the digitalization of teaching and learning was emphasized (Nacken, 2015).

One central component of learning analytics is the learning management system (LMS). Not only is the interaction data of students generated and potentially stored in it. It can also be used to present results of possible learning analytics reports. Since 2019, the Moodle LMS is used university-wide at RWTH Aachen University with about 3000 courses that are being created each semester. It is also one of the most used LMS worldwide (Medved, 2015) and by far, the most used LMS in higher education in Germany. Today, more than 170 German universities use the Moodle learning management system (Steitz, 2020). Therefore, not only the theoretical results of this dissertation can be applied at other universities. In practice, the developed software implementation can be used to facilitate the adoption of data privacy-friendly learning analytics in higher education.

1.2 Objectives

The objective of this dissertation is to bring learning analytics to higher education institutions and simplify the adoption of learning analytics. This dissertation focusses on the learning management system Moodle, but the presented findings and conclusions are applicable to other LMS as well.

In the following, the main questions raised in this dissertation are discussed.

How can Learning Analytics be integrated into a Learning Management System?

In this thesis, I discuss how learning analytics can be integrated into a learning management system. As RWTH Aachen University uses Moodle as central LMS, this thesis covers Moodle as case study. There are many individual tools available to support learning analytics in Moodle, but no single software solution exists to provide learning analytics to lecturers. I discuss technical requirements for the software and technical challenges like the scalability to large courses or even universities. The goal is to operate the software with more than 45,000 students. However, not only the technical side of the implementation is covered. The research question also covers how a learning analytics application can be implemented in a usable and approachable way for users. Finally, the question shall be answered how the system can support professors and teaching assistants in their day-to-day teaching activities. For this, I elaborate a suitable evaluation strategy that assesses the created software in terms of usefulness, usability, and impact on learning and teaching.

How can a Learning Analytics system support changing requirements, data sources, and use cases?

Similar projects have been implemented as part of research projects or ongoing research. Most of these projects do not exist anymore or are outdated today. In this dissertation, the goal is to create a system that does not only support current use cases and requirements but is also extendable and open to other data sources to support future use cases. Other researchers, system operators, and developers shall be able to create own reports and visualizations and integrate them into the created learning analytics system. The created framework shall make it as easy as possible to adapt to new use cases and changing requirements.

How can the barriers to use Learning Analytics be minimized to achieve a higher adoption of Learning Analytics in the German higher education sector?

The adoption of learning analytics in Germany is scarce. One main reason are the data privacy regulations and concerns. This dissertation covers them and discusses what issues related to them can be solved on a technical level. In addition, the dissertation explains how other universities can adopt the created software and how they can provide learning analytics to their lecturers.

To answer the research questions, a software application is implemented in this dissertation. The preliminary work, software requirements, as well as the implementation are presented and discussed. Afterwards the implementation is evaluated regarding its use and benefits. The logged data as well as feedback from IT operators is used to conduct the technical evaluation. Two longitudinal case studies are conducted to evaluate the software regarding its benefits for lecturers, usage, and usability. In contrast to most related research, all logged data was stored anonymously to answer the last research question.

1.3 Outline

This thesis is composed of eight chapters.

Chapter 2 gives an introduction into fundamental concepts and background knowledge of this thesis. The chapter compares the field of learning analytics to related research fields and presents learning analytics initiatives and networks. Afterwards it introduces Moodle, the central learning management system at RWTH Aachen University. In addition, the legal and ethical foundations regarding data privacy and protection for this thesis are explained.

Chapter 3 presents related work regarding the implementation and evaluation of learning analytics software. It presents architectures, tools, and individual software solutions for learning analytics in general but also specific to Moodle. Afterwards, related work is presented containing evaluation techniques used to evaluate the results of this dissertation.

Chapter 4 describes the implementation of two software prototypes developed as part of previous learning analytics initiatives. Conclusions were drawn during the process that have influenced the analysis and the final implementation.

Chapter 5 describes the conceptual approach by formulating design and software requirements based on the analysis of the related work and the previous software prototypes.

Chapter 6 explains the practical implementation of the software. The resulting software was split into two plugins for Moodle, which will be presented and explained. In addition, the operation of the software is discussed and the user interface is shown.

Chapter 7 presents the evaluation of the created software. First, the data that was collected over the course of two semesters is presented to put the following chapter into context. The actual evaluation is split into multiple parts. A technical evaluation is conducted to draw conclusions regarding scalability, extensibility, and transferability of the implementation. Afterwards, two longitudinal case studies are presented which evaluated the software in the field by using questionnaires and user feedback.

Finally, Chapter 8 concludes the research, evaluation, and adoption of this dissertation and outlines future challenges.

2 Background

In this chapter, the fundamentals for the remaining thesis are covered. The first section introduces the field of learning analytics. This includes common definitions in literature, related research fields, learning analytics initiatives, and existing process models. Afterwards, Moodle is presented. Moodle is the central LMS at RWTH Aachen University and most of the implemented software in this thesis integrates or connects to it. The usage of Moodle in the higher education sector is discussed. In addition, the flexible plugin structure of Moodle is outlined, as the later presented implementation will make use of it. Finally, the concepts of data privacy, data protection, and ethics in the process of collecting and handling of data are explained. As this thesis creates a data privacy-friendly solution for learning analytics, the legal regulations in Europe as well as Germany are presented.

2.1 Learning Analytics

This section introduces the field of learning analytics by providing common definitions found in literature and presenting related research fields. Afterwards, learning analytics initiatives and networks that implement and promote the field of learning analytics are presented. Lastly, learning analytics processes and their underlying theoretical models are discussed.

2.1.1 Definition

There is no universally agreed upon definition for learning analytics. Various definitions have been provided in literature. One of the most common definitions was provided by George Siemens in the call for papers of the first international conference on learning analytics and knowledge (LAK):

“Learning analytics is the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs” – G. Siemens (Siemens, 2010)

Other common definitions for the term *Learning Analytics* in the literature include:

“Learning Analytics is about collecting traces that learners leave behind and using those traces to improve learning.” – E. Duval (Baker et al., 2012)

“Learning analytics is an emerging field in which sophisticated analytic tools are used to improve learning and education. It draws from, and is closely tied to, a series of other fields of study including business intelligence, web analytics, academic analytics, educational data mining, and action analytics” – T. Elias (Elias, 2011)

“Learning analytics refers to the interpretation of a wide range of data produced by and gathered on behalf of students in order to assess academic progress, predict future performance, and spot potential issues. Data are collected from explicit student actions, such as completing assignments and taking exams, and from tacit actions, including online social interactions, extracurricular activities, posts on discussion forums, and other activities that are not directly assessed as part of the student’s educational progress. Analysis models that process and display the data assist faculty members and school personnel in interpretation. The goal of learning analytics is to enable teachers and schools to tailor educational opportunities to each student’s level of need and ability.” – The 2011 Horizon Report (Johnson et al., 2011)

In summary, the goal of learning analytics is to use data of learners and their contexts to improve learning.

2.1.2 Related Research Fields

Learning analytics has strong connections to other related research fields including fields like information visualization as well as smaller research fields like educational data mining. In the following, the related research fields are presented. First, the origins of learning analytics are explained by introducing analytics in general and web analytics. Second, the differences to the closely related research fields educational data mining and academic analytics are discussed. Finally, the two supporting fields of information visualization and dashboard design are presented.

Analytics in General

Analytics has its roots in the field of business intelligence and management of information systems (Shum & Ferguson, 2012). The goal was to guide strategic decisions by automated performance indicators build on data. The term *analytics* itself is described by the Oxford Dictionary as “systematic computational analysis of data or statistics” (Simpson, 1989).

Most commonly, analytics is divided from a taxonomical view into three main categories: descriptive, predictive, and prescriptive analytics (Appelbaum et al., 2017; Davenport & Harris, 2017; Delen & Demirkan, 2013). Often, a fourth category, *diagnostic analytics*, is added in literature (Banerjee et al., 2013). The following lists outlines each type of analytics:

- **Descriptive analytics** describes what has happened or is happening. It summarizes data to provide meaningful information like views of monitoring data. It might also highlight trends or patterns in such data. The output is mostly visualized in form of tables and charts.
- **Diagnostic analytics** identifies why something has happened or is happening. It helps to understand the causes leading to the achieved performance. Diagnostic analytics is often used in environments where hierarchical data is present. An example might be a change of the layout of a website leading to different click rates.

- **Predictive analytics** predicts potential future outcomes by using statistical or data mining techniques. An example might be to predict the sales for a product.
- **Prescriptive analytics** goes one step further by focusing on identifying policies and decisions that might lead to the desired performance. This is done by using simulations and optimization techniques. As outcomes the predictive analytics system might show how a given system input needs to be changed to achieve the desired effect.

More detailed comparisons of the categories of analytics have been made by Delen and Demirkan (2013) as well as by (Shao et al., 2014). Table 1 summarizes their findings.

Table 1: Comparison of different types of analytics

Type	Questions	Methodology	Outcomes
Descriptive	What happened?	Data queries, dashboards, reports	Problems and Opportunities
Diagnostic	Why did it happen?	Simulation, data mining, forensics	Understanding of operations
Predictive	What will happen?	Data mining, statistics, linear regression, forecasting	Projections of future problems and opportunities
Prescriptive	What should be done?	Optimization, simulation, expert systems	Decisions

The implementation presented in this thesis will mostly focus on descriptive analytics but the related work in Chapter 3 will also feature applications for diagnostic and predictive analytics.

Web Analytics

Web analytics is the measurement, collection, analysis, and reporting of web data to understand and optimize web usage (WAA Standards Committee, 2008). It is used to measure the effectiveness of web pages concerning specific goals. The definition itself already shows how closely related learning analytics and web analytics are.

A common metric is the *visitor engagement* describing the degree of user interaction (Bekavac & Garbin Praničević, 2015). This metric includes information like how much time a user has spent on a website or how many pages he visited. Another common metric is the *conversion rate* (Beri & Singh, 2013) indicating how much of a website's visitors created an account or used the website in a way that goes beyond its normal usage. Regarding business intelligence, this metric is used to track *cost per conversion* indicating how much money needs to be spent for one user to register. These web analytics measures are often paired with user experience methodologies like A/B testing and customer journey analysis (Chaffey & Patron, 2012). Many of these approaches are directly applicable to the field of learning analytics.

Important data sources for web analytics can be similar or even identical to those used for learning analytics. These sources include server log files as well as client-side information communicated via JavaScript like the browser information. In contrast to other fields, web

analytics has well-established tools like Google Analytics¹ and Matomo² (formerly Piwik) which have been used by researchers for over a decade (Clark et al., 2014; Tyler & Ledford, 2006).

The obvious and most notable difference between web analytics and learning analytics is that web analytics is more focused on marketing and user conversion aspects leading to a monetary benefit while learning analytics focuses on the enhancement of learning. Still, there are large overlaps in the used techniques and tooling. Past research has already shown that tools like Google Analytics are capable of supporting learning analytics research (Filvà et al., 2014). In addition, many of the findings regarding the usability of analytics tools are applicable to learning analytics (Stewart, 2017).

Educational Data Mining

The International Educational Data Mining Society defines the term *Educational Data Mining* (EDM) as follows:

“Educational Data Mining is an emerging discipline, concerned with developing methods for exploring the unique and increasingly large-scale data that come from educational settings and using those methods to better understand students, and the settings which they learn in.” – Koedinger (2019)

The fields of learning analytics and EDM have strong overlaps. Both are data-intensive approaches to understand and optimize education. However, while both fields are similar regarding the processes and objectives, the used techniques for learning analytics and EDM can be different.

EDM applies data mining techniques like clustering or classification to educational data and has a greater focus on automated discovery while learning analytics places a considerably greater focus on leveraging human judgement. In learning analytics, the analysis of data is just another tool to support human judgement. Another difference is the adaptation and personalization of results. EDM is often used for automated adaptation like an automated feedback system for students. The results of learning analytics are instead designed with instructors and learners in mind (Siemens & Baker, 2012).

A more in-depth review of the two research fields was done by Siemens and Baker (2012) who concluded that the research field of EDM is more focused on the usage of methods and applying them in an automated way, whereas the field of learning analytics prefers human-centric systems that support instructors and learners in their decisions. A brief comparison of the fields is shown in Table 2.

¹ <https://analytics.google.com/>

² <https://matomo.org/>

Table 2: Comparison of Learning Analytics and Educational Data Mining (adapted from Siemens and Baker, 2012)

	Learning Analytics	Educational Data Mining
Discovery	Leveraging human judgement is key; automated discovery is a tool to accomplish this goal	Automated discovery is key; leveraging human judgment is a tool to accomplish this goal
Reduction & Holism	Stronger emphasis on understanding systems as wholes, in their full complexity	Stronger emphasis on reducing to components and analyzing individual components and relationships between them
Origins	Learning analytics has stronger origins in semantic web, "intelligent curriculum," outcome prediction, and systemic interventions	EDM has strong origins in educational software and student modeling, with a significant community in predicting course outcomes
Adaptation & Personalization	Greater focus on informing and empowering instructors and learners	Greater focus on automated adaption (e.g. by the computer with no human in the loop)
Techniques & Methods	Social network analysis, sentiment analysis, influence analytics, discourse analysis, learner success prediction, concept analysis, sensemaking models	Classification, clustering, Bayesian modeling, relationship mining, discovery with models, visualization

Academic Analytics

The term *Academic Analytics* was used originally by the *Blackboard* course management system to describe the data collection functionality of the software (Baepler & Murdoch, 2010). Later the term was used to describe any form of application from classic data mining techniques to educational data as the following citation shows.

“With the increased concern for accountability, academic analytics has the potential to create actionable intelligence to improve teaching, learning, and student success. Traditionally academic systems—such as course management systems, student response systems, and similar tools—have generated a wide array of data that may relate to student effort and success. Early academic analytics initiatives are seeking to predict which students are in academic difficulty, allowing faculty and advisors to customize learning paths or provide instruction tailored to specific learning needs.” – Campbell et al. (2007)

In contrast to learning analytics and educational data mining, academic analytics is aimed towards decision making at higher levels. Table 3 shows a comparison of both fields.

**Table 3: Comparison of Learning Analytics and Academic Analytics
(Siemens & Long, 2011)**

Type of Analytics	Level or Object of Analysis	Who benefits?
Learning Analytics	Course-level: social networks, conceptual development, discourse analysis, “intelligent curriculum”	Learners, faculty
	Departmental: predictive modeling, patterns of success/failure	Learners, faculty
Academic Analytics	Institutional: learner profiles, performance of academics, knowledge flow	Administrators, funders, marketing
	Regional (state/provincial): comparisons between systems	Funders, administrators
	National and International	National governments, education authorities

As the above table shows, academic analytics does not support individual instructors or learners and instead focuses on political or economic support on national or international levels (Ferguson, 2012).

Information Visualization

A key element in learning analytics is the visual presentation to support the instructors and learners in their interpretation and decision-making. For this reason, data visualization as well as usability and user experience play an important role. Meaningful visualizations are necessary to make results easy to interpret for instructors so that they can draw the right conclusions.

A common definition for the term *information visualization* is given below:

“Information visualization is the use of computer-supported interactive visual representations of abstract data to amplify cognition. Its purpose is not the pictures themselves, but insight (or rapid information assimilation or monitoring large amounts of data).” - Card et al. (1999)

The goal is to use graphical representations of data to help evaluate the data and to gain new insights from them. Information visualizations therefore support the communication and illustration of these findings (Fekete et al., 2008). There are many different visualization methods. For some data types, certain visual representation forms and diagrams have been established as standard, including representations for time series data, statistical distributions, maps, hierarchies, or network visualizations (Heer et al., 2010). Many of these visualization methods can be directly applied to graphical representations created for learning analytics. The field of information visualization is therefore an integral part of learning analytics to convey relevant information and many results from the field are directly applicable to learning analytics.

Dashboard Design

As part of the field of information visualization, the presentation of data in form of dashboard is widespread. A dashboard is a graphical user interface that contains multiple indicators, often presented in a way that makes them easy to understand at first glance (Few, 2006). In many cases, the user can click on visualizations to extend them or view detailed reports. A good design of the dashboard is important to communicate key information to users and to make information easily accessible (Brath & Peters, 2004).

Dashboard designs are part of the field of Human-computer interaction (HCI) and incorporate results from the field of information visualization, but the corresponding research is often focused on individual research fields (like learning analytics). A definition for a learning dashboard from the field of learning analytics have been coined by Schwendimann et al. (2016). It is a “single display that aggregates multiple visualizations of different indicators about learner(s), learning process(es) and/or learning context(s)” (Schwendimann et al., 2016). Dashboards are mostly used in web-based applications and can therefore often be integrated directly into existing learning environment (Verbert et al., 2014).

2.1.3 Initiatives and Networks

There are multiple national and international societies as well as networks of companies, universities, or research initiatives supporting research and development in the field of learning analytics. Many of these networks play a central role by organizing conferences or creating software implementations. These will be presented in the following. Multiple of these initiatives will be mentioned throughout the remainder of this thesis.

Society of Learning Analytics Research

The *Society for Learning Analytics Research (SoLAR)* is a network of international researchers and universities exploring the role and impact of analytics on teaching, learning, training and development. The society was founded by over 14 international universities. SoLAR’s vision is “to be recognized as the leading organisation promoting research and implementation of learning analytics in education” (SoLAR, 2021). It is well known for its yearly *Learning Analytics and Knowledge Conference (LAK)*, which also coined one of the most famous definitions for learning analytics (see Section 2.1.1). In 2021, the conference took place for the 11th time.

LACE

The *Learning Analytics Community Exchange (LACE)* was an EU-funded project that involved nine partners from across Europe (LACE, 2015) associated with over 51 European universities (LACE, 2016a). It aimed to integrate communities from the learning analytics and EDM community by sharing solutions between them. LACE established an *Evidence Hub* that collected evidences of the effect of learning analytics (Ferguson & Clow, 2017) and an evaluation framework for learning analytics (Scheffel et al., 2017). The EU funding ended in 2016, after which the group established a special interest group named *Learning Analytics Community Europe* allowing them to keep the acronym (Drachler & Scheffel, 2016). The formed group was integrated into SoLAR and focuses on learning analytics specific to European institutions. This includes topics like the application of the GDPR and non-English languages (SoLAR, n.d.–a).

SHEILA

SHEILA (Supporting Higher Education to Integrate Learning Analytics) is another EU-funded project that aims to build a policy development framework to promote formative assessment and personalized learning (SHEILA, n.d.). SHEILA joined the special interest group LACE in 2016 (LACE, 2016b). The project shares the goal of bringing learning analytics to higher education institutions in Europe. One result of the project is the SHEILA framework, that assists with strategic planning and policy processes for learning analytics (Tsai et al., 2018).

Apereo Learning Analytics Initiative

The *Apereo Learning Analytics Initiative (LAI)* aims to accelerate the operationalization of learning analytics software and frameworks and to support the validation of learning analytics development across institutions (Apereo, 2015). In contrast to other initiatives, LAI focuses on the software development. LAI has released a design of a learning analytics architecture, called *Shuhari - Open Learning Analytics from Apereo*, and is actively publishing open source software online, discussed in more detail in the next chapter.

Learning Analytics Latin America

Learning Analytics Latin America (LALA) is a project aiming to build capacity for learning analytics adoption in Latin America (Maldonado-Mahauad et al., 2018). The project is funded by the European Union and has multiple European partners. In cooperation with SoLAR, the special interest group *LALA* was formed as continuation of the efforts of the LALA project (SoLAR, n.d.-b). Similar to other projects, LALA has published a framework (*LALA Framework*) containing guidelines for learning analytics adoption taking into account the special requirements in Latin America (Pérez-Sanagustín et al., 2019).

National Initiatives and Projects

In addition to international initiatives, there are multiple national initiatives and projects, often funded by the respective ministries of education. In the following, national initiatives are presented, in no particular order.

The *Spanish Network Of Learning Analytics (SNOLA)*, funded by the Spanish *Ministry of Economy and Competitiveness* composes 13 Spanish research groups and has a strong technological focus. The goal of the network is to promote collaboration among the participants, organize learning analytics initiatives on a national level and to provide training (Caeiro-Rodríguez et al., 2016).

The *Innovation Fund Denmark* launched the *Danish Center for Big Data Analytics driven Innovation (DABAI)* in 2016 with a 17 million Euro budget (Alstrup et al., 2017). The projects aim to integrate computer science research, including machine learning and visualization, into the education sector. It has funded a variety of learning analytics projects (Ali et al., 2020; Lorenzen et al., 2017; Nielsen, 2019).

Jisc is a non-profit organization located in the United Kingdom that supports institutions in higher education by providing digital services and resources (Jisc, n.d.-a). Jisc is funded by federal function programs as well as memberships by individual institutions (Jisc, n.d.-b).

Other countries mostly rely on funding of individual projects without coordinated national initiatives. The nationwide association of new media for teaching and learning in Austria has published guidelines for use of learning analytics in the higher education sector (Leitner et al.,

2019). The ministry of education of Austria announced financial support for learning analytics applications in the next years (Nouri et al., 2019). The *International Council for Open and Distance Education* (ICDE), supported by the Norwegian ministry of education has published guidelines for ethics in learning analytics (Slade & Tait, 2019).

An in-depth analysis of learning analytics efforts in Europe has been done by Nouri et al. in 2019 (Nouri et al., 2019) listing research projects across Europe.

Learning Analytics in Germany

The adoption of learning analytics in Germany is scarce (Ifenthaler & Schumacher, 2016). An interview study by Ifenthaler and Yau (2019) comes to the conclusion that further resources are required to adopt learning analytics. One key issue is the handling of personal data and concerns of regarding privacy and ethics (Drachsler & Greller, 2016). Due to strong data privacy laws, the German higher education sector struggles to deliver basic descriptive course statistics to lecturers and students. The adoption of learning analytics has been stagnating and in some cases even been reversed following concerns of stakeholders regarding privacy and ethics over the handling of personal data (Drachsler & Greller, 2016).

In addition, recent research suggests that the infrastructure of universities complicates the adoption of learning analytics. Central IT systems, like the student management system or the learning management system, are often deeply embedded into the infrastructure and not designed to extract data for analytic purposes (Klasen & Ifenthaler, 2019). On top of technological and legal challenges, a new generation of educators needs to be equipped with fundamental competencies necessary to allow a widespread adoption of learning analytics (Ifenthaler, 2017).

2.1.4 Process Models and Concepts

Learning analytics is an iterative process to foster learning and teaching. Hence, the different phases of learning analytics are carried out in a series of steps. Multiple process models for learning analytics have been proposed in the past. I present these as well as relevant process models from related research fields in the following to explain how learning analytics can influence the teaching and learning process.

One of the first process models for analytics in the educational sector has been proposed by Campbell and Oblinger (2007) for the field of academic analytics. The authors divide the process for academic analytics into five steps: *capture*, *report*, *predict*, *act*, and *refine*. Based on the model of Campbell and Oblinger (2007), Clow (2012) developed the *Learning Analytics Cycle*, shown in Figure 1. The work is also based on work from Kolb's *Experiential Learning Cycle* (Kolb, 1984), work on reflective practice by Schön (1983) and Laurillard's *conversational framework* (Laurillard, 2002).

The cycle consists of four steps and includes the learners as separate one, in contrast to most other models. Starting from the learners, the model processes the data. This can be done automatically or by manual effort. The next step shows metrics or analytics, which provide insights in to the learning process. Clow (2012) explicitly mentions dashboards as one possible type of visualizations in addition to exemplary visualizations like listing the students at risk and aggregated data visualizations. In the last step, the metrics are used to drive interventions, resulting in an effect on the learners. The cycle does not necessarily need to reach the same learners the data was built on to be complete. An example might be a lecturer who reviews the

data from his students. Additionally, Clow (2012) stresses that learning analytics does not need to include all four steps, although this might lower the effectiveness of the process.

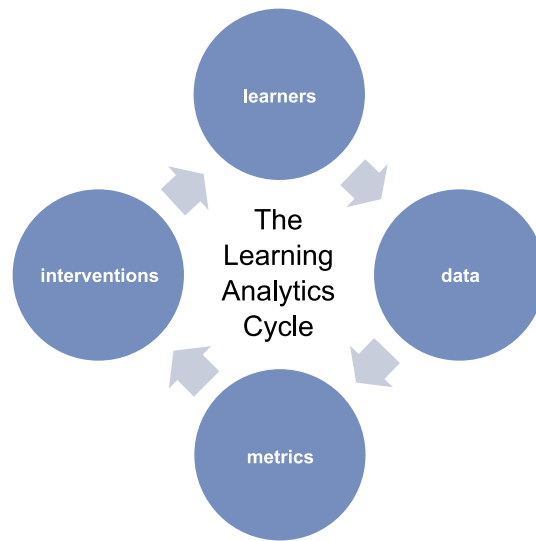


Figure 1: The Learning Analytics Cycle (Clow, 2012)

Another model, originally not designed for learning analytics, was introduced by Dron and Anderson (2009). They describe a process for the design of “collective applications”. Figure 2 shows the model.

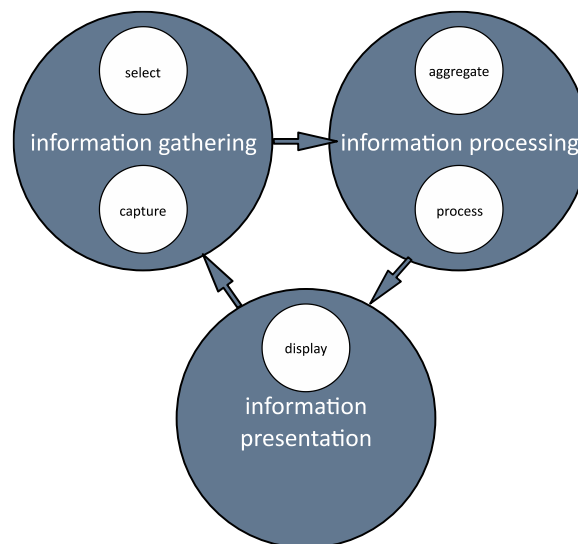


Figure 2: Collective application model (Dron & Anderson, 2009)

The process is divided into three phases: information gathering, processing, and presentation. In the first phase (*gathering*), the actions of interest are selected and captured. The second phase (*processing*) aggregates the data and processes it by applying selected algorithms needed for the task. Finally, the third phase (*presentation*) displays the results back to the user (Dron & Anderson, 2009).

Elias (2011) presented a process model for learning analytics based on the work of Dron and Anderson (2009) and Baker (2007). Baker (2007) introduced the concept of *knowledge continuum* in which data is transferred into wisdom. The concept describes that in the first step data is obtained and turned into information by giving it a meaning. After that, the information is turned into knowledge by analyzing and synthesizing it. Finally, based on the knowledge, wisdom is established to achieve goals. The resulting model features an iterative three-phase process, similar to the model of Dron and Anderson (2009). In contrast to other process models, it highlights the importance of organizations, computers, people, and theory by putting them into the center of it. The model is shown in Figure 3.

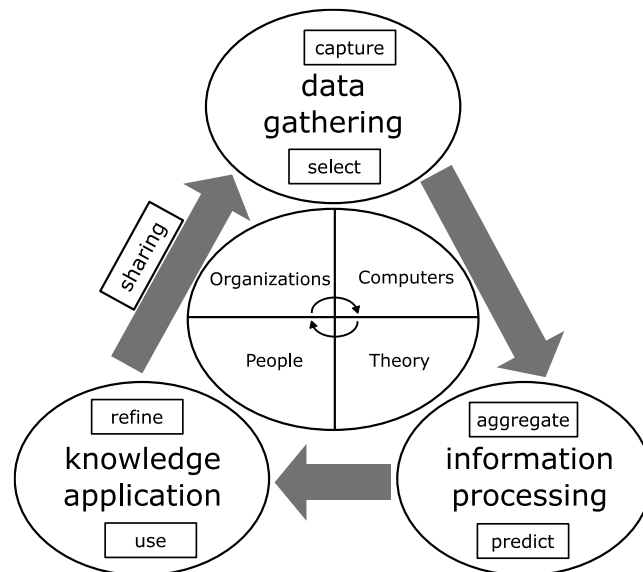


Figure 3: Learning Analytics Continuous improvement cycle (adapted from Elias, 2011)

The process model by Chatti et al. (2012) describes the process of learning analytics, similarly to Elias (2011), as an iterative cycle based on three steps: *data collection and pre-processing*, *analytics and action*, and *post-processing* (Chatti et al., 2012). Figure 4 shows the process.

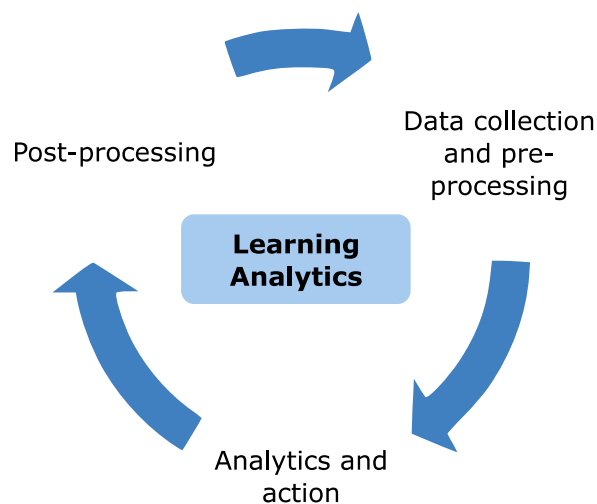


Figure 4: Learning Analytics process (Chatti et al., 2012)

The first step in the model is the collection of data (Chatti et al., 2012). This data might be derived from various sources and systems. This step includes the aggregation and pre-processing of the data as in some cases the data might be too large or requires to be cleaned before further processing. In preparation for the next step, the data is transformed into a suitable format making it easier to process the data for further tasks. In the next step, the data is analyzed (Chatti et al., 2012). Different methods might be applied in this step depending on the objective of the analytics task. This step includes the analysis and visualization, in addition to the execution of actions deriving from the data. Possible actions include monitoring, analysis, prediction, intervention, assessment, adaptation, personalization, recommendation, and reflection. The last phase aims to improve the analytics task. It might involve refining the existing data based on new data sources, modifying used variables or visualizations or even applying new analytics methods (Chatti et al., 2012).

Verbert et al. (2013) introduced a process model for learning analytics consisting of four phases (Figure 5). In contrast to other models, this model focuses on the application of data, not on the collection and aggregation. In the first stage (*awareness*), data is visualized. The *reflection* phase assesses how useful and relevant the presented information is by focusing on users' questions. The *sensemaking* stage answers questions identified in the previous phase. Finally, the *impact* phase induces new meaning or change behavior of users if necessary.

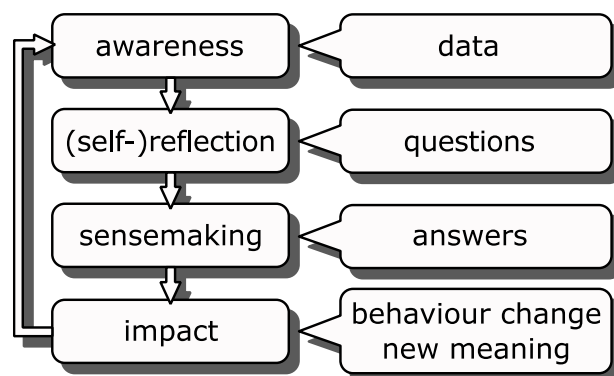


Figure 5: Learning Analytics process model (Verbert et al., 2013)

The process model by Siemens (2013) focusses on a systematic approach to learning analytics. The process is divided into seven components: *collection*, *storage*, *data cleaning*, *integration*, *analysis*, *representation and visualization*, and *action*. The full model is shown in Figure 6. In his work, Siemens emphasizes the importance of a “data team”, responsible for the implementation. According to the author, the “systemic approach to analytics requires a combination of skills and knowledge that are likely not in the possession of a single individual” (Siemens, 2013).

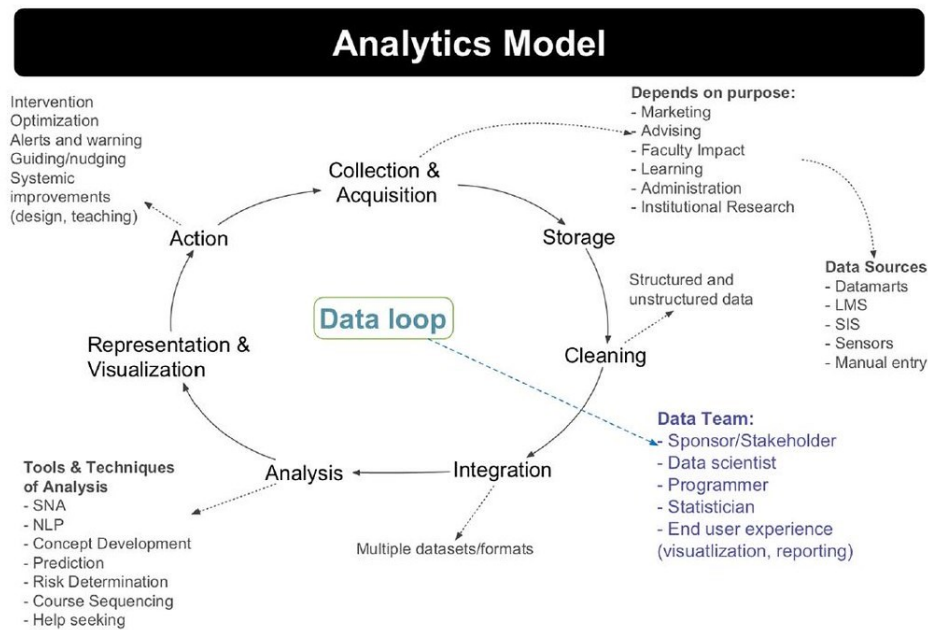


Figure 6: Learning Analytics model (Siemens, 2013)

A reference model by Cho et al. (2016) is based on privacy requirements as foundation, as shown in Figure 7. The phases of the model are known from previous processes: Similarly to the model of Clow (2012), the process starts with a learning activity. The second step collects the data, after which it is stored and processed in the third step. In the fourth step, the data is analyzed and visualized afterwards. In the final step, based on feedback, actions are taken and the steps are repeated. The cycle is based on accountability, information security and privacy compliance. Cho et al. (2016) emphasizes that privacy principles like getting the consent of user and transparency need to be applied prior to the process of data collection.

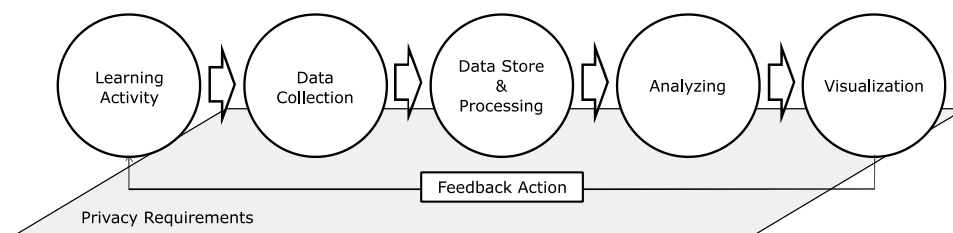


Figure 7: Assignment of privacy principles to the reference model for learning analytics (adapted from Cho et al., 2016)

The model was created as a reaction on the upcoming introduction of the GDPR (see Section 2.3.2). It was the first learning analytics model to include data privacy considerations. This stresses that learning analytics needs to be executed within the boundaries of ethical and legal requirements. The reference model has been standardized by the *Information Technology for Learning, Education and Training* committee as ISO/IEC TR 20748-1:2016 (ISO/IEC, 2016) as workflow of the reference architecture for learning analytics.

What all models have in common is the circular process structure, the incorporation of data or information, and the execution of a separate *action* or *impact* step. While some models precisely define all steps down to possible tools and techniques, other authors emphasize that not all steps

need to be executed for a full learning analytics iteration. In contrast to other processes, only the model of Cho et al. (2016) includes data privacy compliance. Overall, it can be stated that learning analytics is an iterative process that is implemented in a series of steps.

2.2 Moodle

Moodle is a web-based and free learning management system (LMS). It is written in the PHP development language and its code is released as open-source software under the *GNU General Public License*³ (GPL). The first version was developed by Martin Dougiamas in 2002 to help educators create online courses. The software focuses on interaction and collaborative construction of content and is in continual development. Moodle supports various pedagogical principles and learning scenarios like blended learning, flipped classroom, and other forms of e-learning teaching setups. The software supports additional plugins and features a rich plugin store allowing educational institutions and instructors to customize their installation.

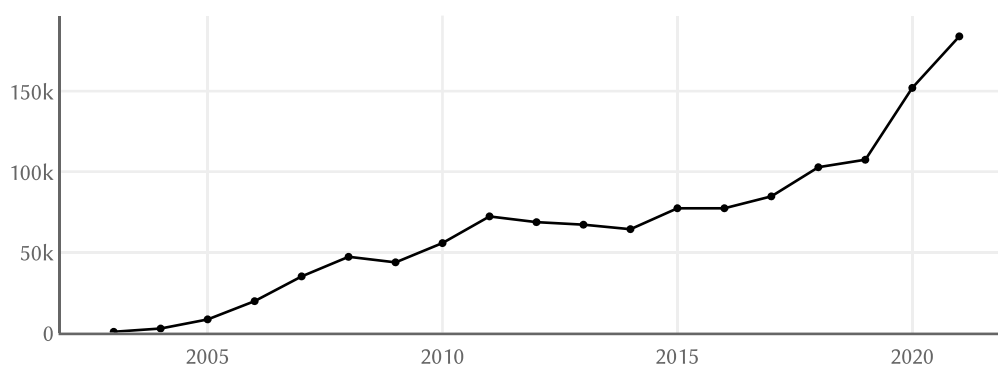


Figure 8: Number of registered Moodle pages since 2003

Moodle is being used by over 184,000 sites with 260 million users hosting over 293 million resources and 4 billion quiz questions (Moodle.org, 2021b). The number of sites using Moodle has been strongly increasing since the software was released in 2003 as shown in Figure 8⁴.

Moodle can be integrated into an existing university infrastructure as illustrated in Figure 9 (Hunt, 2012). Other components of the infrastructure are:

- **Authentication Provider:** A central login system for all university systems
- **Student Information System (SIS):** The database for student records including grades, enrolled courses, but also administrative information
- **Document repository:** Platform(s) to store files and collaborate (outside of the LMS)
- **ePortfolio:** A platform to build the resume based on course information, enrolled courses, etc.
- **Reporting:** A central reporting tool to generate high-level information

³ The GPL is a widely used software license, which guarantees end users the freedom to run, study, share, and modify the software.

⁴ The figure was created by manually checking the number of registered Moodle platforms for each year via <http://web.archive.org/> based on the Moodle statistics page (Moodle.org, 2021b)

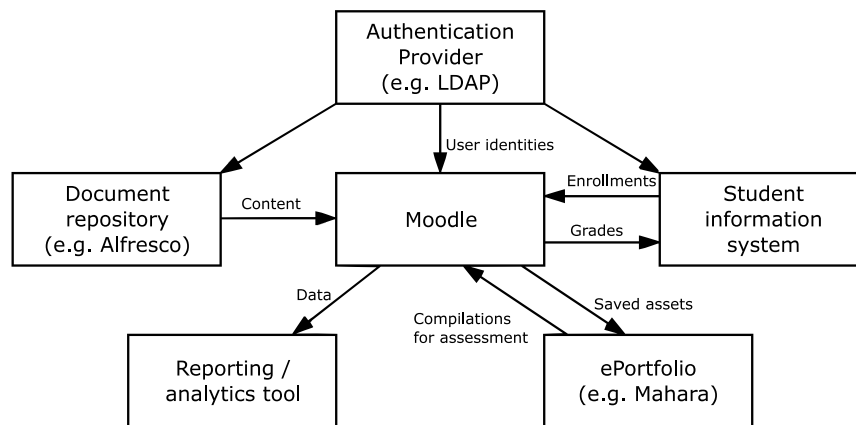


Figure 9: Integration of Moodle into a typical university infrastructure (Hunt, 2012)

Not all components are necessary in a university structure. Especially, the use of ePortfolios as part of a university's infrastructure has been declining, as more options become available for students to create portfolios of their work. Often students tend to use other platform, outside of the context of the university (Reese & Levy, 2009). In addition, technological alternatives like digital badges (Gibson et al., 2015) or even technical solutions like course certificates built on blockchain technologies (Technische Hochschule Lübeck, 2020) have been replacing the idea of a single portfolio for students. Still, the remaining components might be present at most universities in some kind of form.

2.2.1 Adoption and Usage in Higher Education

There are only few established learning management systems in the international higher education sector. According to Gartner, Inc. (2020), there are currently six vendors in the market for proprietary learning management systems (different product names in brackets): *Aula*, *Blackboard* (*Blackboard Learn*), *D2L* (*Brightspace Core*), *Instructure* (*Canvas*), *itslearning*, and *Teamie*. In addition, there are five vendors releasing their LMS as open source: *Apereo* (*Sakai*), *Claroline*, *Moodle*, *OpenOLAT*, and *Open edX*.

According to various publications, the two most used LMS in higher education are Moodle and Blackboard Learn. A national survey in the educational sector in the US from 2013 shows a 41% market share for Blackboard, with Moodle being the second-most used LMS with a 23% market share (Green, 2013). A survey by Capterra of over hundred LMS buyers from educational as well as other sectors found that Moodle was "far-and-away the most used" (Medved, 2015). According to recent market research by Gartner, Inc. (2020), the open-source use of LMS has been slowly but steady shrinking in the higher education sector in North America and Northern Europe, while open-source products remain strong in other parts of Europe, Latin America, Africa and parts of Asia (Gartner, Inc., 2020). Main drivers for the move away from open-source software towards commercial solutions are missing cloud functionalities as well as a lack of features (Gartner, Inc., 2020).

Regarding the German higher education market, Moodle and ILIAS are considered the most used products. Based on a report from 2009 (Graf, 2009), about one third of 171 investigated universities uses Moodle, while another third uses ILIAS. According to a recently updated list of *Technical University of Darmstadt* (Steitz, 2020) that keeps track of used LMS in the higher education sector of German-speaking countries, about 173 German universities use Moodle. The history of the list is plotted in Figure 10 and indicates a clear upwards trend.

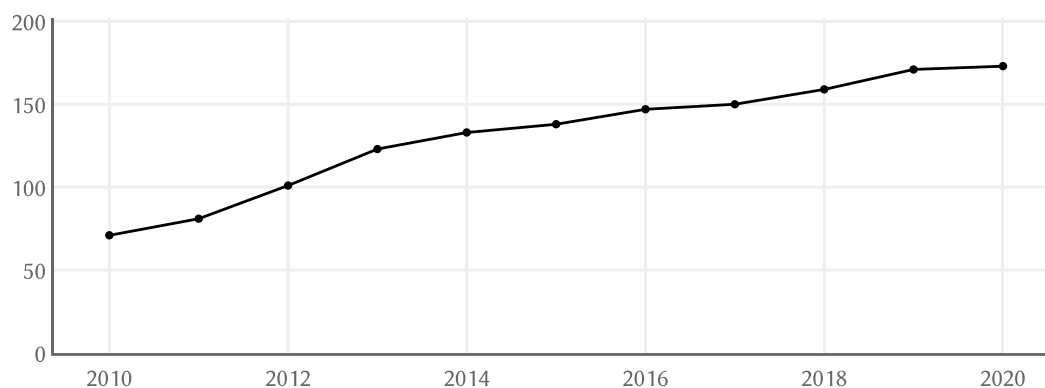


Figure 10: Number of Moodle pages at German universities since 2010 (based on Steitz, 2020 and the corresponding Internet Archive page⁵)

As part of the evaluation of this thesis, I investigated the number of Moodle instances myself and found that 63% of LMS in the higher education sector in Germany are Moodle, with six of the ten biggest universities using it (see Section 7.3.3 of the evaluation).

2.2.2 Moodle at RWTH Aachen University

Since the summer semester 2019, Moodle is used as central LMS at RWTH Aachen University. Previously, the university used a custom solution based on *Microsoft SharePoint*, which was introduced in 2007 as part of an university-wide e-learning strategy (Schroeder et al., 2008). Prior to the university-wide introduction, Moodle had already been used individually for several years by individual chairs (Dondorf et al., 2016).

Before the central rollout of Moodle in 2019, there were several transitional phases spread over the course of multiple semesters (Decker et al., 2018). Early after the decision was made to switch to Moodle, a first testing phase with 15 courses was conducted in the winter semester 2017/18. In the second phase, in summer semester 2018, about 50 courses participated (Decker et al., 2018). Several plugins were installed and code was added to improve the user experience. The final testing phase happened in the winter semester 2018/19, in which the focus was placed on the cooperation with lecturers and students (Decker et al., 2018). A evaluation of the use of Moodle after the last testing phase received positive feedback from over 80% of users (Kubiak & Grzemeski, 2019).

Today, Moodle is an integral part of the learning environment of the university. It has been used for multiple semesters and more than 3,000 courses are created each semester (Decker et al., 2018) supporting the teaching and learning of over 45,000 students (RWTH, 2020). RWTH Aachen has become a central member of the Moodle user group of universities in German-speaking countries⁶. In addition, the university is one of twelve gold members in the Moodle User Association, which funds projects for the Moodle core and decides which functionality is implemented in Moodle (Moodle Users Association, 2020).

⁵ <https://web.archive.org/>

⁶ The Moodle user group of universities in German-speaking countries uses the Moodle forum of the Humboldt University of Berlin for organization and discussion: <https://moodle.hu-berlin.de/course/view.php?id=37191>

2.2.3 Plugins

The acronym *Moodle* officially stands for *modular object-oriented dynamic learning environment* (MoodleDocs, 2013). As the full term is mostly useful to programmers and education theorists, it is rarely used anymore. Nevertheless, the first word (*modular*) gives away the technical structure that Moodle is based on. It uses a very flexible plugin system, where even core features are integrated as part of a bigger plugin architecture (Hunt, 2012). To distinguish between base functionality available after installation and functionality installed via plugins, the central parts of the software are referred to as *core API* or simply *core* (MoodleDocs, 2021b).

Thanks to the modularity, Moodle can be enhanced easily and it is possible to write software, which can be used across different Moodle sites. Currently, the Moodle plugins directory contains over 1,800 plugins (Moodle.org, 2021a). For Moodle core and plugin developers, there is an extensive developer documentation available, which contains many tutorials for various parts of the API as well as a references for the Core APIs (MoodleDocs, 2021c). In addition, the Moodle code itself is well documented and newly written code mostly coherent in its style thanks to coding style guidelines (MoodleDocs, 2021a).

The plugin system is based on over 58 different plugin types (MoodleDocs, 2020b). There are plugin types for all kind of functionalities in Moodle. Examples are very specific plugin types for resources like *Assignment feedback*, authentication providers for login routines, or *theme* plugins changing the look-and-feel of the page. In addition, a plugin can introduce new plugin types to extend its own functionality. These kind of plugins are called subplugins.

In this thesis, I will mostly mention three different plugin types: *report*, *local*, and *logstore*. Multiple plugins of type *report* are being presented in the related work (Section 3.3.2). These plugins provide visualizations of data for teachers and administrators. My own software implementation presented in Chapter 6 uses two plugin types: *local* and *logstore*. Local plugins are generic plugins and can be considered the most powerful kind of plugin. While normal plugins serve a very specific purpose and thus only have access to that specific core API, local plugins have access to a variety of different core APIs. This makes it possible to make the use of multiple features packaged as one plugin. Nevertheless, a second plugin type was necessary to develop the software. This is the *logstore* type. A logstore is able to receive triggered events fired inside of Moodle and can in the process write these logs into its own database table.

Thanks to the powerful plugin system of Moodle, the created implementation described in this thesis can be installed by other institutions using Moodle like any other plugin.

2.3 Data Privacy

In the last decades, costs for storage of data have been steadily declining, allowing the storage of terabytes or even petabytes of data (Markl, 2014; Sagiroglu & Sinanc, 2013). Previously, companies had to decide what to collect first. Today, companies often “collect first and ask questions later” (Croll, 2012). Collecting and processing data has become an integral task for many organizations and companies. *Data driven decision making* (DDDM) is a trend that is prevalent in almost all industries. Companies use data to innovate and gain a competitive advantage (Provost & Fawcett, 2013). Case studies even show that companies enforcing DDDM performed better on objective measures of financial and operational results (McAfee et al., 2012).

In 2017, *The Economist* published a story titled *The world’s most valuable resource is no longer oil, but data* (Parkins, 2017) highlighting the importance of data and discussing the implications of

it focusing on data usage in big technology companies. As the collection of data and DDDM has become prevalent, the negative aspects are becoming apparent. Data breaches have revealed personal data of billions of users (Karunakaran et al., 2018) including delicate information like credits card information, addresses, phone numbers, birth dates, and even political party affiliation (Ayyagari, 2012; Gupta & Sharman, 2012). While some people argue that data breaches need to be accepted as the new normal, arguing that “theft of personal information causes minimal harm to consumers” (Riedy & Hanus, 2016), others have advocated for stricter regulations (Perera et al., 2015).

DDDM has made its way into the educational sector. In the United States, universities and school are even required to use data and DDDM to run educational organizations (Piety, 2015). Decisions have to be based on data-backed evidence. Schools might collect and analyze behavioral data, medical records, attendance information, and other information (Mandinach, 2012). In contrast to that, the use of data in Europe is strictly regulated. Particularly, the enactment of the *European General Data Protection Regulation* (GDPR) has set strict regulations on how data can be collected and processed.

2.3.1 Anonymization and Pseudonymization

Distinguishing between pseudonymization and anonymization is important to understand later parts of this thesis. Most regulations only apply to personalized or pseudonymized data. As the final software implementation processes anonymized data (in contrast to most existing learning analytics implementations), the GDPR does not apply to the processed data. Therefore, the difference is explained below.

Table 4: Examples for Personalization, Pseudonymization, and Anonymization

Personalization	Pseudonymization	Anonymization
John Doe logged in.	I4KX71 logged in.	Someone logged in.
Alice Brown logged in.	A3PK0W logged in.	Someone logged in.
John Doe logged out.	I4KX71 logged out.	Someone logged out.

Table 4 shows fictitious log data, as it might appear in an LMS log system. It covers three events: Login of user *John Doe*, login of *Alice Brown*, and logout of *John Doe*. The personalized data contains a direct link between the real user and the logged events. The pseudonymization removes the direct link between the logged data and the real user. Instead, an artificial identifier for each user is used. In reality, this might be an ID created based on user names or just an integer value. After pseudonymization, one can still see that the user who logged out was the user who logged in first, even though one does not know the name of the user. After anonymization, this link is lost. There is no way to link two log entries to the same user anymore.

2.3.2 General Data Protection Regulation

The *General Data Protection Regulation* (GDPR) is an EU regulation on data protection and privacy that defines provisions and requirements related to the collection and processing of personal data of individuals (EU General Data Protection Regulation, 2016). Information related

to an identifiable natural person, like a name or identification number, is considered personal data. Research has shown that the implementation of the GDPR has notably increased the transparency in the web (Anderson & Seck, 2020).

The six core principles of the GDPR are (EU General Data Protection Regulation, 2016, Art. 5):

- Lawfulness, fairness and transparency
- Purpose limitation
- Data minimization
- Accuracy
- Storage limitation
- Integrity and confidentiality

The first principle ensures that the subject is always able to know what data is collected and what the data is being used for. *Purpose limitation* restricts the use of the data collection to an intended purpose and states that the data shall only be stored as long as necessary for that purpose. The third principle (*data minimization*) states that the collection of data must be limited to only the data that is relevant for the purpose. *Accuracy* describes that stored data must be accurate. False data must be deleted or corrected. The principle *Storage limitation* states that the data should only be collected for as long as necessary for the purpose. Exceptions exist for the processing of data in the public interest for scientific and historical research purposes or for statistical purposes. The last principle (*Integrity and confidentiality*) regulates the protection of data from unauthorized access, distribution, or loss.

Based on these six principles, multiple mechanisms and requirements regarding the collection and processing of data have been defined, including the consent for processing data, access to your own data, notification of users in case of data breaches, and the *right to be forgotten*. It is to note, that the GDPR does not apply to the processing of anonymous data while pseudonymization of data is covered by the GDPR (and even recommended).

2.3.3 Situation in Germany

The protection of data is considered important to many Germans. Compared to US citizens, Germans have less trust in big technology companies like Facebook (Krasnova & Veltri, 2010) and are more concerned about privacy of their data in general (Bellman et al., 2004). The reasons are manifold and go back multiple decades. In 1970, the worldwide first data protection law was passed in the state of Hesse in (West) Germany. It was passed as a reaction to technological advances and because of the fear of a dystopian surveillance state as depicted by George Orwell (DER SPIEGEL, 1971). At the same time, across the border in East Germany (German: *Deutsche Demokratische Republik, DDR*), the *Ministry for State Security* (German: *Ministerium für Staatssicherheit*, commonly known as *Stasi*) created a vast network of state employees and citizens to spy on the population fighting any political opposition (Gieseke, 2011). After the German reunification, the experiences from the Stasi provided significant momentum for data protection discussions and the importance of data protection once more became apparent (Schmidt & Weichert, 2012).

Today, the situation regarding data privacy in Germany is complex. There are various regulations and entities affecting the implementation of learning analytics in higher education:

- **GDPR:** As explained in the previous section above, the GDPR is the European regulation, which is implemented by the individual member states of the European Union (EU General Data Protection Regulation, 2016).
- **Bundesdatenschutzgesetz:** The federal regulation that implements the GDPR and specifies various requirements from the GDPR. (Bundesdatenschutzgesetz, 2018)
- **Landesdatenschutzgesetz:** The federal state law, created to regulate the data processing of public entities by state and local governments (DSG NRW, 2018).

In addition, each federal state has its own regulation for higher education, called *Hochschulgesetz*. It contains paragraphs related to controlling, evaluation, and reporting which also further specify data protection and processing (Hochschulgesetz NRW, 2014, §8).

As a requirement of the GDPR, each university has a data protection officer who oversees the handling of data. The data protection officer informs and advises the university but does not issue directives, nor is he responsible for the handling of data himself (EU General Data Protection Regulation, 2016). During the development of this thesis, the data protection officer of RWTH Aachen University was contacted multiple times to make sure the implementation respects all applicable regulations.

2.3.4 Ethics

When implementing learning analytics applications, not only regulations need to be considered but also ethical requirements. There are a number of ethical principles and requirements proposed in literature.

Slade and Prinsloo (2013) propose an ethical framework for learning analytics based on six principles (Slade & Prinsloo, 2013):

1. **Learning Analytics as moral practice:** Learning analytics should be used to help understand rather than measure.
2. **Students as agents:** Students are not just data providers, but collaborators of the system. They need to provide informed consent regarding collection and storage of data.
3. **Student identity and performance are temporal dynamic constructs:** Data needs to have an expiry date and students should be able to request deletion of data to make sure they are able to evolve.
4. **Student success is a complex and multidimensional phenomenon:** Data can easily be misinterpreted and biased. Mechanisms need to be in place for this.
5. **Transparency:** Institutions have an obligation to protect the data.
6. **Higher Education cannot afford to not use data:** Available data should be used to better understand and engage with students. Learning analytics assists all stakeholders.

A list of four ethical principles to be considered for learning analytics implementations has been provided by Pardo and Siemens (2014): transparency, student control over the data, security, and accountability and assessment (Pardo & Siemens, 2014).

A more practical list was proposed by Drachsler and Greller (2016) in form of the *DELICATE* checklist for learning analytics. The checklist contains eight requirements (Drachsler & Greller, 2016):

- **Determination:** Determine the purpose of the software
- **Explain:** State the objectives to users
- **Legitimate:** Clarify the legitimacy of the process
- **Involve:** Involve all stakeholder, including data subjects
- **Consent:** Ask for consent before using the data
- **Anonymize:** Anonymize the data as far as possible
- **Technical:** Guarantee privacy on a technical level
- **External:** When working with external providers ensure they follow the same rules

In addition to the presented ethical guidelines and frameworks, there are general privacy principles like *Privacy by design* (Cavoukian, 2009) and concepts on how to consider ethical challenges when designing software systems in times of big data (Richards & King, 2014). Overall, most authors advocate for a strong involvement of all stakeholders including lecturers, students, and technical personnel processing the data.

3 Related Work

This chapter presents related work regarding software architectures and implementations for learning analytics, and their evaluation. In the first section, software architectures are presented. These describe the technical approach towards learning analytics as a service for educational institutions. The described systems are designed for university-wide use or even for use across multiple institutions.

The second section discusses individual learning analytics projects and tools. In contrast to the presented architectures, most of the projects were designed as part of research projects, often not intended for long-term use or usage in university-wide settings.

In the third section, learning analytics tools for Moodle are presented. This includes core functionality, plugins, commercial software, as well as implementations that do not integrate into Moodle. In addition, I present research projects that incorporated Moodle into their learning analytics settings without releasing the source code. In total, more than thirty implementations are discussed. Figures are included for tools that influenced the software design or user interface of the software implementation.

Lastly, approaches for the evaluation of learning analytics are presented in the fourth section. This includes expert analysis approaches intended for use in the earlier stages of the development, as well as user evaluation methods to evaluate the created software.

3.1 Learning Analytics Architectures

Common learning analytics solutions focus on approaches suitable to be used across multiple systems. Often, not only the log data of an LMS is collected, but also student performance data from the student information system (SIS) and other IT systems. These kinds of architectures are often implemented as standalone systems, not directly integrated into the LMS or SIS. They serve as a central data repository to store data from different sources. In the following, I present existing software architectures used for learning analytics.

3.1.1 Apereo Learning Analytics Initiative

The Apereo foundation is a non-profit corporation with members from the higher education sector (Apereo, 2020a). It supports and develops a variety open-source projects for use in higher education. These projects include *Sakai*, an open-source LMS, a portfolio software, a single sign-on system, and the streaming server *Opencast* (Apereo, 2020b). As explained in Section 2.1.4 (Initiatives and Networks), the Apereo Learning Analytics Initiative (LAI) aims to accelerate the operationalization of learning analytics software. The development process is

carried out openly on GitHub⁷ and documented on a Confluence website⁸. The released software is published under open-source licenses (Educational Community License or Apache license⁹).

The created code base is split into individual software repositories with implementations for data processing, storage, software interfaces for different APIs and user interfaces. Figure 11 shows the overall concept. Most of the software is developed in the programming language Java. The architecture is divided into five components: *Collection*, *Storage*, *Analysis*, *Communication*, and *Action*. Two repositories implement the software interfaces for *xAPI* and *IMS Caliper*. The collected data is stored in a Learning Record Store (LRS), a single repository hosting all learning-related data. The implemented LRS is called *OpenLRW*. The analysis is conducted in the *Learning Analytics Processor* (LAP) software, which does data mining, data processing, prediction modeling, and reporting. The roadmap indicates that Hadoop software (Pig, Hive, Spark) will be integrated in the process in the future (Zeckoski, 2015). The communication and action parts of the architecture are implemented in the *OpenDashboard* component, which was iterated once. The old dashboard software is marked as deprecated/legacy in the software repository. The new *UX* dashboard replaces it. The software currently aims to implement the *Student Early Alert* and *Risk Assessment* models. As the overall architecture of the software is meant to be flexible, in the long term, different models will be supported as well as customizing and extending the software (Zeckoski, 2015).

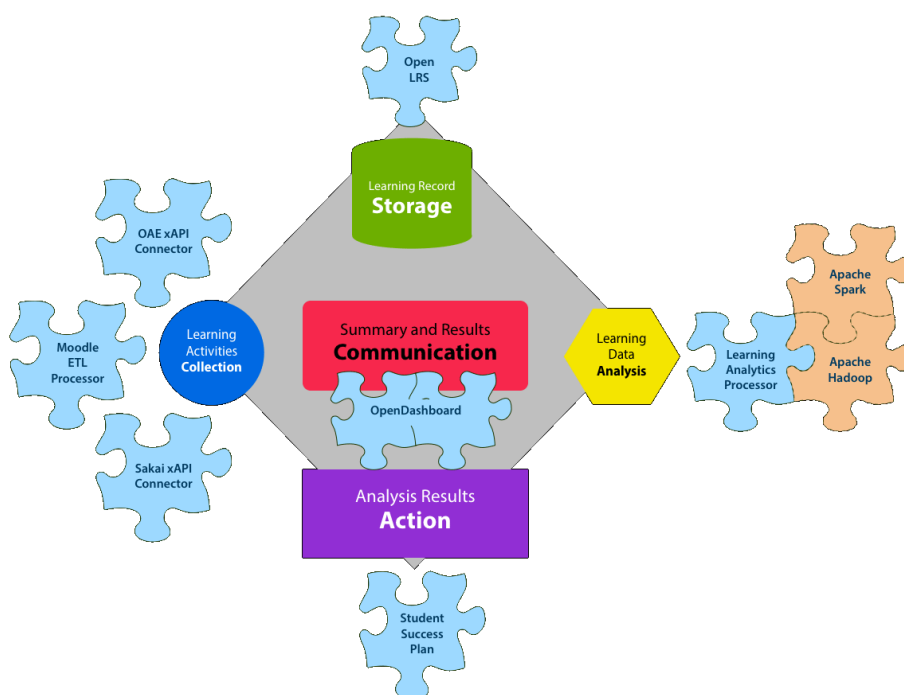


Figure 11: Apereo Learning Analytics Strategic Vision for an Open Learning Analytics Platform (Apereo, 2016)

⁷ <https://github.com/Apereo-Learning-Analytics-Initiative>

⁸ <https://confluence.sakaiproject.org/display/LAI/Learning+Analytics+Initiative>

⁹ The Educational Community License (ECL) and the Apache License are open source software licenses allowing the free distribution and modification of the software. In comparison, the ECL contains modifications regarding the handling of patents to make it better suitable for educational institutions (EDUCAUSE, 2016)

Many of the software repositories on GitHub were recently updated. The storage system, APIs, and the dashboard were updated in 2020. The xAPI provider was updated in 2021. Overall, the software seems to be in active development. Unfortunately, there is little information available regarding installation and operation of the overall architecture. Therefore, the software by Apereo is hard to adopt into existing settings. However, for institutions looking for a long-term learning analytics solution, the Apereo architecture needs be considered and evaluated.

3.1.2 Jisc Open Learning Analytics

Jisc is a non-profit organization supporting higher education institutions in the United Kingdom (see Section 2.1.3). The architecture developed by Jisc uses a three-layer approach, split into *Data Collection*, *Data Storage and Analysis*, and *Presentation and Action* (Jisc, 2018a). Figure 12 shows the core of the Jisc Learning Analytics architecture. The architecture is similar to Apereo as the author explains to build on Apereo’s vision for an open learning analytics platform (Sclater, 2017).

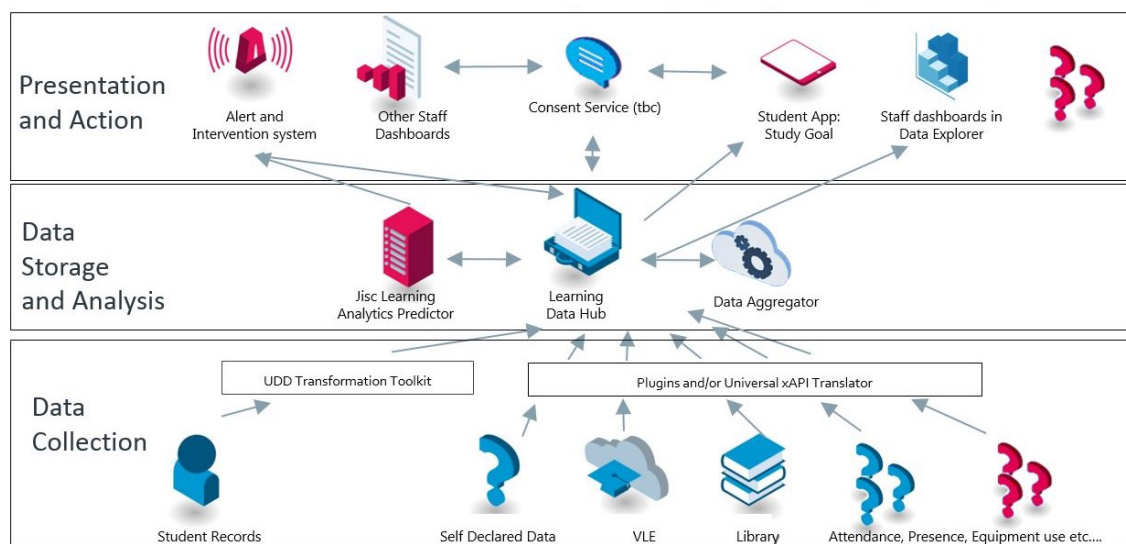


Figure 12: Jisc Learning Analytics Open Architecture (Jisc, 2018a)

The code base is split into several code repositories. Some of the repositories are shared openly on GitHub¹⁰ while other source code of features cannot be found online. The collection layer is implemented based on plugins for Moodle and Blackboard. The code of the Moodle plugin is openly available on GitHub. The *Data Collection* layer, called *Learning Data Hub*, is implemented based on the LRS LearningLocker (Betts, 2015). While the software LearningLocker is open source, the service is not operated by Jisc themselves, but by the company *LearningPool* (formerly *HT2 Labs*), who maintains the open-source version of LearningLocker (Jisc, 2018b). The software uses xAPI statements, which require a vocabulary to define possible statements. This vocabulary is available and documented on GitHub¹¹. The stored data is then processed based on a *Learning Analytics Processor*, which supports a variety of different models. This step

¹⁰ <https://github.com/jiscdev>

¹¹ <https://github.com/jiscdev/xapi>

can incorporate open-source solutions as well as solutions provided by external providers (Jisc, 2018a). The *Presentation and Action* layer is provided by multiple different products. One example is the *Data Explorer*, a web application providing multiple reports and visualizations for the processed data. Other presentation functionality includes insights for students and a framework for Predictive Analytics (Griffiths et al., 2016).

Overall, the Jisc Open Learning Analytics architecture is based on many different components that include open-source software as well as proprietary solutions. While parts of the software stack, mainly the Data Collection layer, are available online, most of the software remains closed source or is even provided by external service providers. Core features like the Data explorer are explained in detail on GitHub¹² including release notes, bug reports and even user documentations. However, the code base has not been published.

The first announcement of open learning analytics by Jisc was made in 2015 (Sclater et al., 2015). The latest information on the Jisc website regarding open learning analytics is from 2016¹³. It appears that Jisc changed their architecture from an open to a closed model, built on collaborations with contractors and service providers.

3.1.3 Notable Approaches

Besides the presented architectures for learning analytics by Apereo and Jisc, there are multiple architectures presented in literature, which I discuss in the following.

In 2011, the Society of Learning Analytics Research (SoLAR, see Section 2.1.3 for networks and initiatives), made a proposal to “design, implement and evaluate an open platform to integrate heterogeneous learning analytics techniques” (Siemens et al., 2011). The proposal describes the overall process of learning analytics and names four tools and resources that are needed for integrated learning analytics toolsets: *Learning analytics engine*, *Adaptive content engine*, *Intervention engine*, and *Dashboard, reporting, and visualization tools*. The document does not technically define implementation details but names multiple examples and shows visual prototypes on how a system could look like to an end user. Siemens et al. propose a modularized development approach in which individual researchers and research teams develop individual widgets or plugins. The goal is to create a modularized, but integrated open platform with standards for adding plugins. Unfortunately, the funding proposal was unsuccessful and no funding was obtained to start working on the proposal (Griffiths et al., 2016). Today, the proposal is often referred to in literature when learning analytics approaches are presented or discussed. To date, the presented architecture remains a theoretical concept.

Another approach was made in a dissertation by Muslim (2018), who implemented *OpenLAP*, a “User-Centered Open Learning Analytics Platform” (Muslim, 2018). The software is based on a conceptual approach of Chatti et al. (2017) for an open learning analytics ecosystem. The source code is published on GitHub¹⁴. The latest commit in the original software repository was made in 2018. Since then, the project was moved to another organization on GitHub¹⁵. Currently, little

¹² <https://github.com/jiscdev/data-explorer/wiki>

¹³ <https://analytics.jiscinvolve.org/wp/2016/11/10/notes-and-presentations-from-8th-uk-learning-analytics-network-event-at-the-open-university/>

¹⁴ <https://github.com/OpenLearningAnalyticsPlatform>

¹⁵ <https://github.com/ude-soco>

to no information is available online regarding the practical deployment of the software. After contacting the authors, they disclosed that the software is currently being actively developed. A new iteration of OpenLAP will be released in the near future.

There were more architectures and frameworks announced on conferences and in literature. Most of these are similar to the proposal of SoLAR: theoretical concepts without implementation details. An example is an *Open Architecture for Learning Analytics* by Hoel and Xiao (2018), who present their approach for a Chinese university. Other approaches mostly focus on specific LMSs or implement architectures for specific features for a particular LMS. An example is *Learning Analytics Profile (LeAP)* for the LMS ILLIAS by the University of Mannheim (Klasen & Ifenthaler, 2019).

3.1.4 Summary

While there are many processes and models presented in literature, the number of implementations is limited. There are only two generic architectures for learning analytics available with details regarding the practical implementation. Of these two, it is questionable if the Jisc Open Learning Analytics concept can even be used by members outside of Jisc as core functionality as well as storage layers are outsourced to service providers. The latest update regarding open learning analytics is from 2016. That leaves the Apereo Learning Analytics Initiative as the only available architecture that can be considered for use in practice. The source code is available online, published under open-source licenses and recently updated. Still, work has to be done concerning maintenance and installation to simplify adoption.

The presented architectures aim to provide learning analytics across different systems. In contrast to that, individual tools used for specific learning analytics purposes or research projects are presented in the following section.

3.2 Learning Analytics Tools

There are many individual tools for learning analytics featured in literature as well as at conferences. In the following, the most popular projects are presented and their purpose, usage, and findings are discussed. This section excludes any implementation that can be used together with Moodle as these are introduced in more detail in the next section.

3.2.1 Learning Analytics Projects

In the following, notable learning analytics implementations are presented, which received much attention in literature. It is to note, that there are more learning analytics implementations than named below. Only the most cited learning analytics implementations are mentioned.

Course Signals

One of the most popular learning analytics implementations is *Course Signals* (Arnold & Pistilli, 2012). Course Signals uses data from the university LMS to provide real-time feedback to students. It relies on data mining of grades, demographic data, past academic history, and other interaction data with the LMS to predict which students might be falling behind. The outcomes for individual students are sent via mail as well as presented to them in the LMS in form of a traffic signal.

Lecturers were given an overview of students through the faculty dashboard presenting detailed information as well as the traffic signal for each student. Course Signals was used by over 23,000 students across 100 courses, with over 140 lecturers utilizing the system over the course of four semesters. The conducted evaluation was based on an anonymous survey at the end of each semester. Most students reported positive experiences claiming that they found the feedback informative and motivating. Negative feedback remained rare. The negative aspects named were the increasing amount of messages from lecturers as well as concerns about over penetration. The lecturers and faculty members perceived the system as improvement and helpful (Arnold & Pistilli, 2012).

Student Explorer

Student Explorer (Krumm et al., 2014) is an early warning system that measures students' academic performance. The software has been iterated multiple times (Lonn & Teasley, 2014). It uses grades, assignment data, and login information from the LMS of the university, aggregates them and generates visualizations. The visualizations show the grades of students in comparison to their peers and performances of individual students. Similar to Course Signals, the relationship to other students was visualized as *Engage* (red), *Explore* (yellow), or *Encourage* (green). The goal of the software was to provide prescriptive guidance for students. The evaluation of Student Explorer showed that advisors used the software mostly during meetings between advisors and students, which was not intended by the developers (Aguilar et al., 2014). Overall, the use of the software was received positively by advisors. Comparisons to previous semesters showed that in one targeted population of students, the grade point average increased after the use of Student Explorer (Krumm et al., 2014).

OU Analyse

OU Analyse (Kuzilek et al., 2015) is a software developed at the Open University (OU) that uses machine learning methods to identify students at risk of failing a course. The software uses student demographic data and information about interaction with the LMS. Different models to predict whether students will succeed or fail in a given course are used. It uses Bayes models, k-Nearest Neighbors algorithm (Duda et al., 1973), and the *CART* method (Breiman et al., 1984) to create properties for each student. These properties are then aggregated and combined into a single value representing the likeliness of a student to fail a course. The course overview page summarizes a number of values and presents these weekly aggregated at the top of the page. Figure 13 shows a part of the OU Analyse dashboard.

The outcome of the machine learning algorithms is displayed to lecturers in an overview page inside the LMS. It shows descriptive data, like scores from assessments as well as course averages next to the predictions for each student. The predictions show whether the student is likely to submit assignments in the future and a student activity recommender. By selecting an individual student, a more detailed view is shown with individual information. It shows a "nearest students" visualization that presents a 2D plot containing students behaving similar to the selected student based on demographical data and other data criteria as well as historic performance of the student.

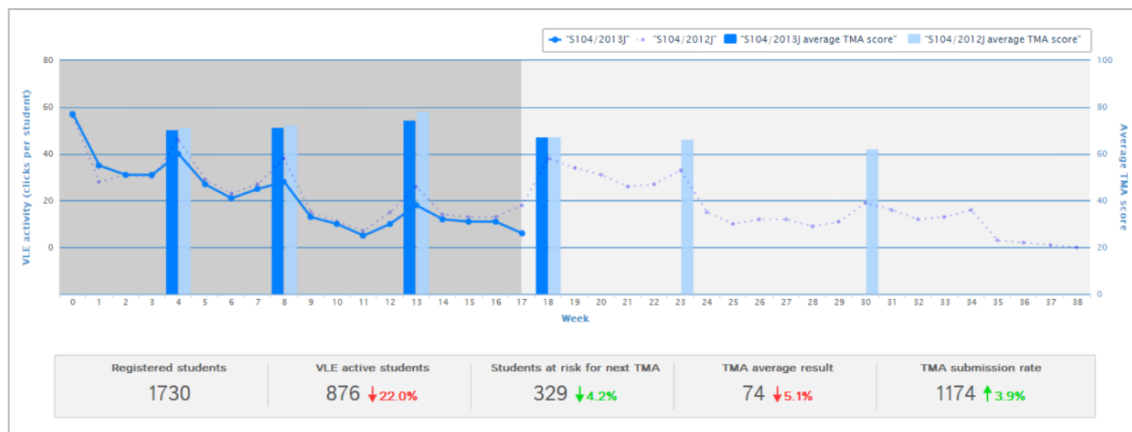


Figure 13: OU Analyse dashboard, part of the course overview page (Kuzilek et al., 2015)

OU Analyse was used over the course of multiple years and is still being actively developed today (Herodotou et al., 2020). The different features of the software have been iterated multiple times in the past based on quantitative (Herodotou et al., 2017) as well as qualitative (Herodotou et al., 2019) feedback from lecturers and students. Results showed that participating lecturers perceived OU Analyse as very useful (Herodotou et al., 2019). Overall, the developers argue that instructors using the software can positively affect students' performance (Herodotou et al., 2019).

Other implementations

Apart from these well-known examples, there are more implementations related to learning analytics. These can be clustered into *self-reflection*, *recommendation systems*, *early alert systems*, and classical data mining tools. The distinction of these tools is not always clear and a single tool could sometimes even be assigned to multiple categories.

The implementations regarding descriptive analytics are mostly tagged as *self-reflection* or *awareness raising* tools. This includes *LOCO-Analyst*, which is based on the *Learning Object Context Ontologies* (LOCO) framework (Jovanovic et al., 2006), that provides multiple visualizations, including a visualization of students' scores in online quizzes, interaction over time (Jovanović et al., 2007). *StepUp!* is a mobile app for awareness and self-reflection for teachers and learners (Santos et al., 2012) supported by a detailed evaluation (Santos et al., 2013). The *Student Activity Meter* (SAM) visualizes learner actions for awareness and self-reflection of students. It was iterated multiple times (Govaerts et al., 2012) and was evaluated based on two case studies (Govaerts et al., 2011). *PDinamet* (Gaudioso et al., 2009) is a framework to support teachers to promote learners' reflection and to assess the course. *Meta Analyzer* (Hwang et al., 2008) analyzes search patterns of students to assist lecturers.

Other implementation can be categorized as *recommendation systems*. In this category falls *SHERPA*, which helps students to find a substitute class if their first choice is full (Bramucci & Gaston, 2012). *E²Coach* is an intervention engine that was used on data from 49,000 physics students and gives advice via an individual web page that includes predictions for which final grade they might receive (McKay et al., 2012). Another software called *CourseAgent* provides course recommendations based on students' assessment of a course relevance to their career goals (Farzan & Brusilovsky, 2006).

The following software can be summarized as monitoring or *Early Alert Systems*. This is the *Automated Wellness Engine (AWE)*, which presents evaluations for lecturers based on feedback of students (Leece & Hale, 2009). *Social Networks Adapting Pedagogical Practice (SNAPP)* creates visualizations based on social network analysis (Bakharia & Dawson, 2011). Evaluations of these systems have shown that early intervention with students can improve students' academic performance (Singell & Waddell, 2010; Tinto, 2012). Lastly, the *Student Success System (S3)* creates a *success index* for individual students (Essa & Ayad, 2012). A comparison of more early alert systems was done by Atif et al. (2015).

In addition to these learning analytics tools, there are tools for classical data mining tasks, often for educational data mining purposes (Avouris et al., 2005).

While this literature review focused on projects describing the actual application or even the implementation of the tools, there are more tools announced in literature. Of these, many were used for a single course or study program and often their implementation and application were not explained. I excluded these tools from my list, but sophisticated literature reviews including these tools have been done multiple times in the past (Romero & Ventura, 2013; Venkatachalapathy et al., 2017).

3.2.2 Generic Analytics Tools

In addition to learning analytics tools, there is a lot of general-purpose analytics software. This includes data analytics applications that make it easy to create visualizations based on data like *Tableau*¹⁶, *Google Data Studio*¹⁷, or even *Microsoft Excel*. Other software suitable for analytics purposes includes programming languages used for data science like *Python* or *R*. In addition, there is a high number of data processing tools, often aimed at processing big data. One software is *Apache Spark*, an analytics engine for big data processing. Another software suite is the *Apache Hadoop* ecosystem, which is very popular and used commercially as well as in research. It features a collection of several tools aimed at collection, processing, and visualizing massive amounts of data. There are more tools in the field. These are the most popular ones.

Outside of data science, there are tools focused on web analytics. The two most prominent examples are *Google Analytics*¹⁸ and *Matomo*¹⁹ (formerly *Piwik*). Both tools are general-purpose web analytics services. Nevertheless, both tools, Google Analytics as well Matomo, have been used multiple times in the past to evaluate access patterns, course material usage, and user behavior (Ellis, 2012; Klindžić et al., 2019; Łodzikowski, 2021; Manzo, 2017).

In addition, Google Analytics has been evaluated by Filvà et al. (2014) for its use as an evaluation tool in educational settings. The gathered data was displayed in a series of graphs to visualize the interaction of students with the course material. Overall, the study concluded that Google Analytics could be used as tool for learning analytics in Moodle, although some limitations were found (Filvà et al., 2014).

Besides individual software solutions, there are multiple companies providing learning analytics and educational data mining. Over thirty software vendors offer related software

¹⁶ <https://www.tableau.com/>

¹⁷ <https://datastudio.google.com/>

¹⁸ <https://analytics.google.com/>

¹⁹ <https://matomo.org/>

products. Established companies are *McGraw Hill Education*, *Knewton*, *D2L*, and *Realizeit* (Johnson & Samora, 2016). Some of the vendors focus on institutional education processes like *ALEKS PPL* from *McGraw Hill Education*, which helps to design courses for the needs of students and their degree programs (McGraw Hill, 2021). Other commercial solutions like *Performance Plus* from *D2L* embed into the LMS and aim to provide analytics for lecturers as well as students.

3.2.3 Learning Record Stores

Learning Record Stores (LRS) are databases that store information received from learning environments. An LRS represents the storage layer in the process flow for using the *Experience API* (xAPI). xAPI is a specification that defines how learning content and data is transmitted between learning systems (ADL, 2017). There are similar specifications like *Caliper* by *IMS Global* (IMS Global, 2015) or the predecessor of xAPI, SCORM (ADL, n.d.). De-facto, most LRS implementations build on the xAPI specification, which I therefore focus on in the following.

Some LMS come with built-in support for xAPI. Moodle supports parts of the specification²⁰ while other features can be added via plugins. There are currently multiple plugins²¹ adding support for external LRS. Integrating xAPI into an LMS with the use of an LRS results in all triggered events being sent to the LRS. Similar to Matomo and Google Analytics, the LRS then generates reports and visualizations from it. In contrast to generic web analytics tools, xAPI statements contain more information about the event that was triggered. To give an example: A student conducting a quiz triggers an event indicating that a quiz was started. While web analytics tools only record the HTTP request (and possibly browser information), the xAPI statement contains information about the kind of event and its context. The statement contains information about the quiz that was conducted, possible questions, and even its answers. This makes the LRS more suitable for educational purposes.

To understand the use of a Learning Record Store, one has to understand how xAPI statements work. Each xAPI statement can contain multiple properties. Three central properties are the following:

- **Actor:** The actor property contains information about who executed an action. This does not necessarily need to be an ID but can also be account information (for example from the Moodle page).
- **Verb:** The verb represents what action was executed. Verbs are represented as *Uniform Resource Identifier* (URI) that links to a registry, which contains more details about the verb.
- **Object:** The object property contains information related to the verb.

Additional properties might contain information about the time or the context in which they happened. The full xAPI statement is then encoded as JSON (Bray, 2017).

Listing 1 shows an exemplary xAPI statement produced by the Moodle plugin *logstore_xapi*²² triggered by a learner accessing the course page. The actor is *John Doe*, with an account on the Moodle platform *moodle.example.com* (ID: 12345). The verb of the statement represents that a

²⁰ [https://docs.moodle.org/dev/Experience_API_\(xAPI\)](https://docs.moodle.org/dev/Experience_API_(xAPI))

²¹ Search results for xAPI in the Moodle plugin store: <https://moodle.org/plugins/?q=xapi>

²² https://moodle.org/plugins/logstore_xapi

viewed event happened. The corresponding object refers to the viewed course (name: *Demo Course*). In addition, the statement contains context information about the Moodle platform and the timestamp indicating when the event was triggered. In a real scenario, the statement would be stored inside an LRS and used to create reports or visualizations.

```
{
  "actor": {
    "name": "John Doe",
    "account": {
      "homePage": "https://moodle.example.com",
      "name": "12345"
    }
  },
  "verb": {
    "id": "http://id.tincanapi.com/verb/viewed",
    "display": { "en": "viewed" }
  },
  "object": {
    "id": "https://moodle.example.com/course/view.php?id=7",
    "definition": {
      "type": "http://id.tincanapi.com/activitytype/lms/course",
      "name": { "en": "Demo Course" }
    }
  },
  "timestamp": "2020-01-07T13:12:28+01:00",
  "context": {
    "platform": "Moodle",
    "language": "en",
    "extensions": {
      "http://lrs.learninglocker.net/define/extensions/info": {
        "http://moodle.org": "1.0.0",
        "https://github.com/xAPI-vle/moodle-logstore_xapi": "0.0.0-dev",
        "event_name": "\\core\\event\\course_viewed",
        "event_function": "\\src\\transformer\\events\\core\\course_viewed"
      }
    }
  },
  "contextActivities": {
    "grouping": [
      {
        "id": "https://moodle.example.com",
        "definition": {
          "type": "http://id.tincanapi.com/activitytype/lms",
          "name": { "en": "Name of the Moodle platform" }
        }
      }
    ]
  },
  "category": [
    {
      "id": "http://moodle.org",
      "definition": {
        "type": "http://id.tincanapi.com/activitytype/source",
        "name": { "en": "Moodle" }
      }
    }
  ]
}
```

Listing 1: Example xAPI statement

There are many different LRSs available. The authors of xAPI, the *Advanced Distributed Learning Initiative* (ADL), maintain a list of LRSs, which conform the xAPI specification. Currently, the list contains 37 LRSs (ADL, 2021). The list contains multiple commercial products like the *Watershed LRS*²³ as well as open-source implementations like *Learning Locker* or *Trax LRS*²⁴. ADL has even published a guide for choosing an LRS (Berking, 2016). In addition, a comparison of different LRS has been published that takes operational performance into account (Labba et al., 2020).

The most popular open-source LRS is *Learning Locker*²⁵, developed by *Learning Pool*. A popular user of Learning Locker is Jisc (Betts, 2015), who is using Learning Locker to provide learning analytics to higher education institutions in the United Kingdom (see Section 3.1.2). The software is released as open source under the GPL, but also available as paid-for Enterprise edition with more functionality. Learning Locker is primarily used to store data and query the stored data, but comes also with a user interface allowing the creation of visualizations and dashboards based on defined queries. To integrate these visualizations into an LMS like Moodle additional integrations need to be developed.

Overall, there are many learning analytics projects and implementations mentioned in literature and at conferences. In addition, multiple tools from related fields like data science or web analytics are applicable to learning analytics. Unfortunately, the number of ready-to-use learning analytics tools for higher education institutions is low. Most publications do not provide the related source code making it impossible to practically build upon the results. An exception to this are learning records stores as one of the most popular ones, Learning Locker, is published as open source. However, an LRS takes only care of the storage of learning data. Tools for analyzing data and integrating the results into learning environments are scarce.

3.3 Learning Analytics for Moodle

In the following section, learning analytics projects and implementations for Moodle are presented. The list does not only contain software suitable for learning analytics, but also software that focuses on educational data mining, academic analytics, and related fields. I investigated the published code base for each tool, for which this was possible, and present corresponding evaluations based on literature if present.

The tools are clustered into the following six groups. First, the features that Moodle has integrated by default (*Moodle Core*) are presented. Second, plugins of type *report* for Moodle that implement Analytics features or similar functionality are discussed. Afterwards, I present any kind of plugin for the Moodle LMS that is not a report but still provides analytics functionality. Following, commercial solutions are discussed. The fifth subsection presents implementations, which are not directly integrated into Moodle. These implementations are external web application or also browser extensions. Afterwards, software is presented that is discussed in literature but without published source code. In the last subsection, the findings are summarized and the presented tools are compared.

²³ <https://www.watershedlrs.com/>

²⁴ <http://traxlrs.com/>

²⁵ <https://docs.learninglocker.net/>

3.3.1 Core Functionality

The following two features of Moodle are available by default as part of the Moodle installation. The first tool is a prediction system based on a machine-learning framework while the second one is able to display descriptive course statistics.

Moodle Analytics

Moodle Analytics, sometimes referred to as *Learning Analytics* in the Moodle documentation and previously named *Project Inspire* (Monllaó & Dalton, 2017) provides a system for predictive analytics. It uses a machine learning approach based on logistic regression and neural networks (using *Tensorflow*) and comes with the following five models by default:

- Courses at risk of not starting
- Students at risk of dropping out
- Students who have not accessed the course recently
- Students who have not accessed the course yet
- Upcoming activities due

The first model is for administrators of the platform and predicts which courses will stay empty. The second model predicts which students are likely to drop out of the course during the last quarter of the course. The model has to be trained with data from the Moodle system itself and does not come preconfigured. An evaluation conducted by the developers showed a 92% accuracy in identifying students at risk of dropping out by using a neural network (Olivé et al., 2018).

For the last three models, I did not find information why Moodle classifies them as predictive analytics. I assume that this is the case for technical reasons, as all three models feature basic descriptive analytics. Overall, of the five models, only the second one does actually provide predictive analytics for lecturers (*Students at risk of dropping out*). The user interface of Moodle Analytics is provided to teachers via the report *Insights*. Figure 14 shows the user interface visualizing whether students of a course are at risk of dropping out or not.

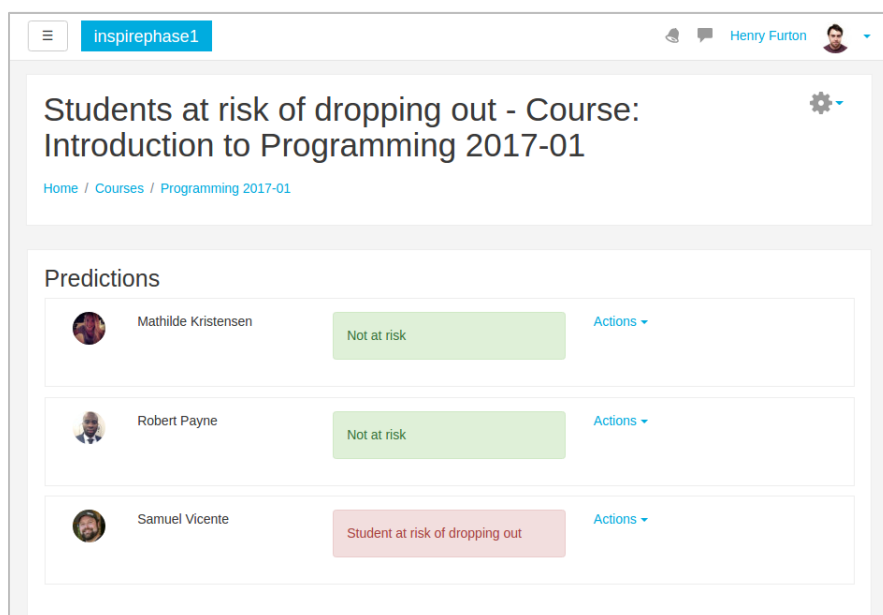


Figure 14: Prediction of students at risk (Monllaó & Dalton, 2017)

Moodle Course Reports

Moodle's internal report system called *Course Reports* is available to instructors. By default, it contains the following eight reports, ordered by their appearance in the Moodle menu:

- **Competency breakdown:** Describes the level of understanding for individual users as defined by metrics setup by the instructor of the course.
- **Insights:** Shows results from Moodle Analytics (discussed above).
- **Logs:** Allows filtering Moodle's standard log data related to this course. It can be filtered for specific users, activities, days, or actions.
- **Live logs:** Shows the latest logged events that are triggered inside the course and shows all related data, including the users and IP addresses.
- **Activity report:** A table showing activity, how often it was accessed, and by how many users, as well as the last time it was accessed for each course.
- **Course participation:** Shows for a specific activity, who accessed it in a defined period.
- **Statistics:** Shows in a line chart the number of hits over time grouped by students, teachers, administrators, and other user groups.
- **Event monitoring rules:** Allows custom event monitoring to be defined to react on specified events.

In the default theme of Moodle, these reports can be accessed by executing the following steps starting from the course start page:

1. Click on *Actions Menu* on the course start page.
2. Click on *More...* in the appearing menu.
3. Click on the tab *Reports*.
4. Click on the name of the desired report.

That means for a lecturer to access any of the reports mentioned at least four actions have to be executed to display the visualizations. Depending on the kind of report, even more actions need to be performed inside the report page as multiple reports start with a blank page or require the user to apply a filter before showing data.

Of all reports, only the *Activity* report displays meaningful information without requiring any prior knowledge or setup for lecturers. It can be read intuitively as it presents the information (views and users) for each activity as table. From my experience, based on talks with lecturers, the *Activity* report is essentially hidden behind multiple layers of required user interaction. Only very few lecturers seem to be aware of this particular report or even of the Moodle reports at all. While Moodle's course reports have been used several times to evaluate courses (Alves et al., 2012; Nagi & Suesawaluk, 2008), the reports themselves, their integration into Moodle, or their user interfaces have not been evaluated. Figure 15 shows a screen capture of the activity report of one of my courses.









Activity	Views	Related blog entries	Last access
 Schedule	74 views by 22 users	-	Thursday, 31 January 2019, 12:13 PM (2 years 110 days)
 01 - Introduction	126 views by 31 users	-	Thursday, 31 January 2019, 12:13 PM (2 years 110 days)
 02 - PHP	147 views by 31 users	-	Thursday, 31 January 2019, 12:13 PM (2 years 110 days)
 03 - JavaScript	115 views by 31 users	-	Thursday, 31 January 2019, 12:13 PM (2 years 110 days)
 04 - Server	65 views by 30 users	-	Wednesday, 31 July 2019, 2:19 PM (1 year 294 days)
 05 - Security	66 views by 30 users	-	Thursday, 31 January 2019, 12:18 PM (2 years 110 days)
 Code Samples	55 views by 25 users	-	Sunday, 28 July 2019, 1:22 PM (1 year 297 days)
 Questions and Discussion	68 views by 22 users	-	Friday, 8 November 2019, 2:41 PM (1 year 194 days)

Figure 15: Activity report of a Moodle course

The existing reports can be extended by Moodle's internal plugin system. All plugins of type report are added to the list of plugins accessible through the same menu as other reports. Available reports are presented in the following section.

3.3.2 Report Plugins

In addition to the functionality provided by Moodle after its installation, Moodle features a rich plugin system with over 1,800 plugins²⁶. The category for reports in the Moodle plugin store contains 47 plugins as of April 2021²⁷. Of these plugins, a few target administrators (e.g. benchmark statistics or database information) while others show detailed information like the pictures of students. Report plugins extend the list of course reports by using the same user interface. In the following, I present the reports that could be used for learning analytics or related purposes in Moodle, ordered by the number of sites that are currently using the plugin as reported by the Moodle plugin store, with the most used plugin listed first.

Overview statistics

The plugin *Overview statistics*²⁸ offers side wide statistics regarding users and courses. The plugin generates reports showing the number of logged in users per day as well as visualizations regarding the language or country of users. The plugin targets administrators of the platform but might also be suitable for lecturers using Moodle instances with only few courses. The plugin is currently used at 3,300 websites. The latest release was in 2021, with the latest Moodle version being supported.

²⁶ <https://moodle.org/plugins> (March 2021)

²⁷ Moodle Report plugins: <https://moodle.org/plugins/browse.php?list=category&id=4>

²⁸ Overview statistics plugin: https://moodle.org/plugins/report_overviewstats

Courses Usage Statistics

The plugin *Course Usage Statistics*²⁹ aggregates data on how Moodle courses are used. The development of the software was started in 2018. The latest version was released in July 2020. The software evaluates Moodle courses and puts them into one of three categories:

- **Course is used as communication channel:** Mainly the forum is used.
- **Course is used as file repository:** Mainly files or links are used.
- **Course is used effectively:** The course contains quizzes or similar activities.

One question the plugin tries to answer is how courses are being used by professors. The primary user group is therefore not teachers but decision-makers or administrators trying to understand how the platform is used. It is therefore rather a useful tool for the field of academic analytics than learning analytics, but might still be a tool to be considered for institutions looking to implement analytics.

Monitoring of learning plans

Similar to the previous plugin, the plugin *Monitoring of learning plans*³⁰ does not provide analytics for lecturers and students. It focuses on the creation or evaluation of learning plans and aims at learning plan managers as user group. The latest version of the software was published in 2019. It is currently being used by over 1,000 sites. The offered statistics make it possible to compare students' skills and participation in courses.

Events Graphic Report

The plugin *Events Graphic Report*³¹ visualizes the Moodle logs by showing the most active users and most triggered events. The report shows the data in a simple page view with multiple visualizations. The latest release was in 2014. Recent versions of Moodle are not supported anymore.

Forum Graph

*Forum Graph*³² visualizes interactions in a Moodle Forum. The software was developed in 2013. As of now, it is still marked as being in an "early stage" on the plugin page. The latest version was released in 2016. The software reads all threads and answers in a forum and visualizes them as force-directed graph. It is useful to conduct social network analyses and has been used for that task in multiple course evaluations, even in recent years (Gkontzis et al., 2017; Polo et al., 2019; Verykios et al., 2017).

3.3.3 Generic Plugins

While many analytics plugins are developed as plugins of type report, the plugin type comes with several technical restrictions. Therefore, some plugins related to analytics are released as

²⁹ Course Usage Statistics plugin: https://moodle.org/plugins/report_coursestats

³⁰ https://moodle.org/plugins/report_lpmonitoring

³¹ https://moodle.org/plugins/report_graphic

³² Forum Graph plugin: https://moodle.org/plugins/report_forumgraph

different plugin type. In the following, these are listed. As before, the list is ordered based on the number of sites using the plugin.

Configurable reports

The plugin *Configurable reports*³³ is widely with over 12,000 installations according to the plugin page. The tool was created in 2011 and recently received an update for the latest version of Moodle. It allows administrators as well as lecturers to create custom reports for courses by using a custom set of defined queries as well as writing SQL queries on their own. After installation, the tool can be added to courses as block. Lecturers can then configure the added report and even get a detailed view of the provided visualization. After the creation of a report, it can also be defined who is able to view a report. That means a lecturer can for example create a report only meant for himself or for all of his students. The plugin comes with a variety of different visualizations supporting the presentation of timelines in form of bar charts as well as simple pie charts. In addition, the data is displayed as table.

Although, the plugins is widely used, its usage is rather complicated requiring technical knowledge and a good understanding of the underlying data sources. In addition, the great flexibility of the software comes with the downside of having many options. Lecturers without technical experience would likely need to complete a training first. Nevertheless, the plugin has been used to evaluate courses and teaching activities in the past (Genkov & Kesova, 2014; Saqr et al., 2017).

Analytics graphs

*Analytics graphs*³⁴ provides visualizations and reports for lecturers based on the Moodle standard log. The plugin adds five reports to the Moodle Course reports. In addition, it can be added to the course as a block. The five visualizations are the following:

- **Assessment distribution:** Visualizes the distribution of grades in a box plot.
- **Access Graph:** Shows for each activity how often it was accessed.
- **Number of active users:** Shows when users are active over the day in form of a bar chart.
- **Quiz and Assignment submissions:** Visualizes the number of submissions for each quiz or assignment.
- **Hits distribution:** For each enrolled user, it shows a line chart visualizing the activity over the past weeks.

In addition to the visualizations, it allows lecturers to submit a mail to a group of students (for example when they have not accessed an activity). Some of the plots are available immediately while other plots took up to a minute to be generated in my local tests.

The plugin is installed on over 1,800 sites and has been used multiple times in the past to evaluate courses and teaching activities (Nilakusmawati et al., 2021; Saqr et al., 2017).

³³ Configurable reports plugin: https://moodle.org/plugins/block_configurable_reports

³⁴ Analytics graphs plugin: https://moodle.org/plugins/block_analytics_graphs

Web Analytics

Web Analytics³⁵ is a Moodle plugin that acts as a bridge between conventional web analytics tools and Moodle. It supports *Matomo* as well as *Google Analytics* by adding their respective JavaScript code snippets to the Moodle page. Only the tracking snippet is integrated, meaning that no user interface is available inside of Moodle. Therefore, only administrators with access to the Analytics services can use the services to create visualizations for lecturers.

xAPI Plugins

There are multiple plugins connecting external LRSs to Moodle. In the following, each one is discussed.

The plugin *logstore_xapi*³⁶ implements a logstore for Moodle meaning that it receives all triggered events. It then sends them to an external LRS in form of xAPI statements (see Section 3.2.3 for information on xAPI and LRS). According to the Moodle store statistics, the plugin is currently being used in 455 Moodle installations.

*mod_tincanlaunch*³⁷ is a plugin that uses xAPI activities and integrates them into Moodle course rooms. The integration format includes content created with Articulate Storyline or Adobe Captivate among others. The launched activities are then tracked and the resulting xAPI statements are sent to an external LRS.

Other plugins are developed to connect specific LRSs. There is *logstore_trax*³⁸, which acts similar to the first xAPI plugin, by sending triggered Moodle events to an external *Trax LRS* instance.

The presented plugins focus on sending data from Moodle to the external LRS to log events. A different approach is taken by the plugin *block_annulab_lrs*³⁹, which embeds as block into Moodle courses and provides a user interface to access statistics provided by the *Annulab LRS*. After integration into Moodle, the teachers of a course can access these statistics in form of tabular data and charts.

SmartKlass

The plugin *SmartKlass™ Learning Analytics Moodle* is a dashboard that supports teachers by providing visualizations and reports for their courses. It was developed in 2014. The latest version was released in 2016. The plugin stores data on an external server (*smartklass.klassdata.com*) by using xAPI (ADL, 2017) but also uses the Moodle standard log to generate events. Overall, the plugin contains a number of different indicators, presented in form of dashboards. Based on my analysis of the source code and information from the repository on GitHub⁴⁰, the plugin cannot be installed anymore since 2017 as the external server is offline.

³⁵ https://moodle.org/plugins/tool_webanalytics

³⁶ https://moodle.org/plugins/logstore_xapi

³⁷ https://moodle.org/plugins/mod_tincanlaunch

³⁸ https://moodle.org/plugins/logstore_trax

³⁹ https://moodle.org/plugins/block_annulab_lrs

⁴⁰ SmartKlasse source code on Github: https://github.com/klassdata/moodle-local_smart_klass

GISMO

*GISMO*⁴¹ is a tool created by Mazza and Milani (2004) for Moodle that analyzes the internal log of Moodle and creates various visualizations. The project was built on a previous research project called *CourseVis* (Mazza & Dimitrova, 2007) that created visualizations for learning management systems. The tool focuses on instructors and tutors of a course, as it is embedded into the course as a block but not visible to students. GISMO implements visualizations regarding discussions in Moodle forums, results of quizzes and assignments, and accesses to the course. The tool was created around 2004. The latest release is from 2014.

MOCLog

MOCLog (Mazza et al., 2012) extended *GISMO* by adding indicators for students. It was created by the same authors as GISMO. It defined a variety of use cases including use cases for lecturers and students as well as administrators and study program managers. In contrast to GISMO, users had to configure the tool more in order to reflect the different user roles. In its presentation at the *Moodle Research Conference* in 2012, a preliminary evaluation of the tool was announced including an expert analysis and a user analysis to evaluate the efficiency and satisfaction of the software. Unfortunately, the announcement of the presentation of the tool in 2012 remains the last information available on the project website⁴². In contrast to GISMO, the software is not available in the Moodle plugin store.

Engagement Analytics

The *Engagement Analytics* plugins⁴³ are a set of three plugins that provide visualizations regarding the engagement of students. The plugin consists of three types: A block that can be added to courses, a report that adds the visualizations to the Moodle Reports, and a *mod* plugin that can be added as activity. The plugins were developed as part of a research project of an Australian university in 2012. The last release of the software was in 2014. In corresponding publications (Liu et al., 2015; Liu et al., 2016), the set of plugins is referred to as *Moodle Engagement Analytics Plugin* (MEAP), although that name is not used on the official Moodle documentation page⁴⁴.

The plugin uses three indicators (*login activity*, *forum activity*, and *assessment activity*) to create a single percentage metric for each student indicating which students are at risk. The software was evaluated based on feedback from lecturers. The talks are summarized as a “positive step in providing staff with relevant information” (Liu et al., 2015).

3.3.4 Commercial Software

In contrast to the presented open-source plugins, there are multiple commercial software products available for Moodle. In the following, these are presented.

⁴¹ https://moodle.org/plugins/block_gismo

⁴² The project website is not available anymore, but its content can still be accessed via <http://web.archive.org/web/20180308032715/http://moclog.ch/>

⁴³ Engagement Analytics plugins: <https://moodle.org/plugins/browse.php?list=set&id=20>

⁴⁴ Engagement Analytics in the Moodle documentation: https://docs.moodle.org/22/en/Engagement_Analytics_Plugin

IntelliBoard

The company *IntelliBoard, Inc.* offers reports and analytics for a variety of learning management systems, including Moodle, but also Blackboard Learn, Canvas, and Google Classroom. According to their website, over 3,000 websites use the software (IntelliBoard, 2021). It can be installed as a Moodle plugin, but requires a software license if used longer than the testing period. Thanks to the testing period, I was able to install the software locally and try it myself.

In contrast to other Moodle plugins, IntelliBoard integrates itself into the Moodle navigation making it easier to access. The starting page features a dashboard showing various information like the number of courses for the lecturer, the progression of students (presented as a pie chart), correlations of grades and time spent, most often occurring events, and an overview of the graded activities and their outcomes. In addition, many visualizations feature different view options allowing to change the form of presentation. Often, the data can be filtered for specific courses. Figure 16 shows a screen capture of the course dashboard.

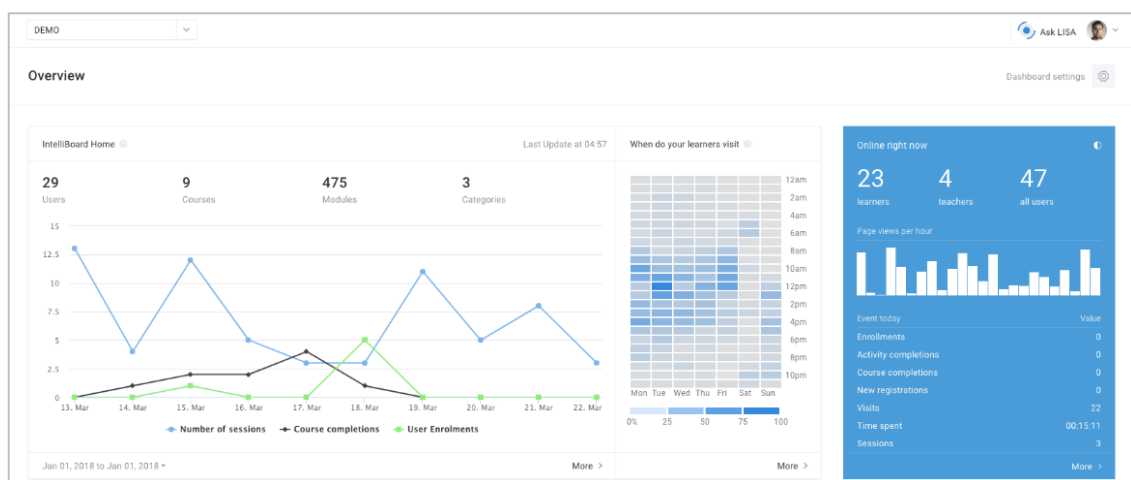


Figure 16: IntelliBoard course dashboard (Kochnev, 2021)

Next to the dashboard of IntelliBoard, four more pages can be accessed:

- **Courses:** Shows a table listing various information about each course, like the number of enrolled students, the average grade reached, number of activities, overall visits to the course and more.
- **Courses Sessions:** Shows the amount of time spent and visits in a course for users.
- **Reports:** Contains more than 20 more reports with more detailed information about various aspects including information about course completion, activities of courses, assignments, grades, and more.
- **Monitors:** The page presents a dashboard containing time series plots showing when students accessed a course over time and which courses were visited the most.

The amount of data IntelliBoard can display far exceeds all other presented tools. I looked at the source code⁴⁵ and found that the tool does this, by bringing its own logging system with

⁴⁵ Parts of IntelliBoard are published on Github: <https://github.com/intelliboard/intelliboard>

own database tables. That way, IntelliBoard does not depend on Moodle's own logging system and can create visualizations (like course sessions) that the Moodle logging system itself is not able to provide.

Overall, the IntelliBoard software felt very user friendly and feature-rich. The only problem I encountered was that the sheer amount of different visualizations and reports makes it hard to find specific reports. Especially the menu *Reports*, which presents more than 20 links in form of a drop-down menu, makes it hard to understand which reports might be of interest. Nevertheless, of all tested tools, IntelliBoard was by far the most mature and useful for lecturers. As of April 2021, the annual price for a university like RWTH Aachen (45,000 students) is \$21,000 according to their website.

LearnerScript

*LearnerScript*⁴⁶ is developed by *eAbyas Info Solutions*. Similar to IntelliBoard, it integrates into the Moodle navigation and provides a dashboard as entry point. The dashboard contains three boxes on top that display the number of courses, the site visits, and the average amount of time spent in Moodle. Below that, multiple tables provide information about courses (similar to the courses table of IntelliBoard), grading, quizzes, assignments, and other resource types.

A link to a page *Reports* presents the users with 18 more reports. Many of the reports only show a single number (like the total amount of time spent on the platform). Others provide tables with more information about activities or users.

As I was not able to have a look at the source code, I cannot make any claims on how the software works in the background. However, due to the plugin being of type *block*, I imagine that it uses the Moodle log (in contrast to IntelliBoard, which creates its own logging tables).

Compared to IntelliBoard, the software did not feel as user friendly and feature-rich. This is also reflected in the price of the software, which comes at \$2,450 (one-time payment).

Other commercial products

There are other commercial products on the market labeled as “Analytics for Moodle”. I found *Zoola Analytics*⁴⁷ and *Edwiser Reports*⁴⁸ that claim to integrate into Moodle. However, I was unable to find code or a demonstration to try. *AspirEDU* offers a plugin for Moodle called *Dropout Detective*⁴⁹, which informs lecturers about potential dropping out students. Additionally, the company offers an *Instructor Insight* dashboard.

Apart from integrations into Moodle, there are customized Moodle instances (often offered as part of a hosting package) that included custom reporting and analytics features. This includes *Lingel Learning*⁵⁰, which claims to “take Moodle reporting to a whole new level” (Lingel Learning, n.d.) and *OpenLMS*⁵¹, which integrates a feature, called *X-Ray Learning Analytics* providing “deep insight into learning behavior” (Borbon, 2020).

⁴⁶ <https://learnerscript.com/>

⁴⁷ <https://www.zoola.io/>

⁴⁸ <https://edwiser.org/reports/>

⁴⁹ https://moodle.org/plugins/local_aspiredu

⁵⁰ <https://lingellearning.com/>

⁵¹ <https://www.openlms.net/>

3.3.5 Implementations not integrated into Moodle

In addition to plugins available in the Moodle plugin store and commercial products, there are tools that can be used for analytics purposes in Moodle. Most of these tools were created as part of research projects and their source code is unfortunately not available online. The exception to this is the project *LeMo*, which I was able to try out myself and therefore describe in more details. Unfortunately, of the projects described in the following, no software is currently being maintained anymore.

LeMo

LeMo (German: *Lernprozess-Monitoring*) is a standalone web application that supports the import from different learning management systems including Moodle (Fortenbacher et al., 2013). It is released as open source under the GNU General Public License on GitHub⁵² and can be downloaded from the project website⁵³. It does not embed into Moodle, meaning that users need to create additional accounts to access course reports. The user interface features a dashboard as well as detailed course visualizations. LeMo supports 14 different types of visualizations, which can be summarized in three categories: Use of the learning management system, navigation, and performance. The first category presents data points as activities over time, visualized as bar charts, box plots, heatmaps, or treemaps. The second category analyzes the time sequence in which students accessed the courses by using navigation graphs, path visualizations, or circle graphs. Lastly, the performance category presents the performance of students in form of box plots and bar charts. The software was evaluated by a survey of ten participants and received overall good feedback (Elkina et al., 2014). Figure 17 is a screen capture of a visualization showing the number of accesses of a course throughout a semester.

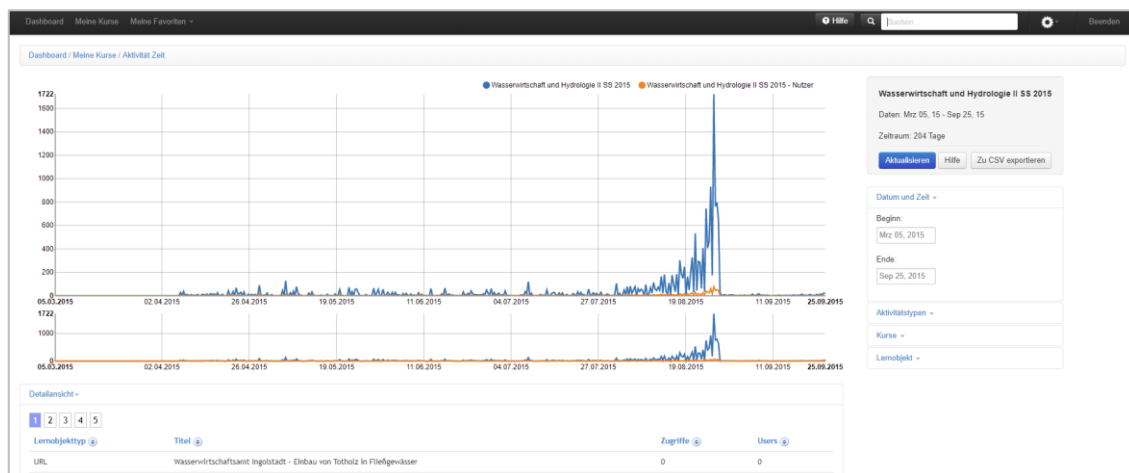


Figure 17: Activity over time visualized in LeMo

⁵² <https://github.com/LemoProject>

⁵³ <http://www.lemo-projekt.de/>

Moodle Analytics Dashboard

The *Moodle Analytics Dashboard* (Einhardt et al., 2016) is a browser extension developed for Moodle. In contrast to previous approaches, this software was developed as a browser extension for Google Chrome. It creates visualizations by aggregating data from multiple Moodle pages. The tool was installed by over 70 users and evaluated based on a questionnaire with 15 users. The questionnaire asked for the importance of features and used the System Usability Scale (SUS) questionnaire for evaluation receiving a score of 83. The features that were rated most important by users were the number of participants, the number of page views, and the number of activities. The software is released as open source under the MIT license on GitHub⁵⁴ and published in the Google Chrome extension store. The latest release is from May 2017.

MDM Tool

MDM tool (Moodle Data Mining) is a data mining extension for Moodle (Luna et al., 2017). It can be added to courses as a block⁵⁵. After execution, MDM tool first extracts information from the Moodle database. After that, it preprocesses this data and allows lecturers to add or modify the shown data. This step can be used to modify the data or anonymize data sets. Lastly, the transformed data is used as input for the data mining software, which can perform three data mining methods: Clustering, Classification/Decision tree, and association. The results can be downloaded as text or PDF file. To show the usefulness, a case study was conducted that used a real-life data set of engineering students to apply all three data mining methods. The models and results were analyzed and used as feedback for instructors (Luna et al., 2017).

Meerkat-ED

*Meerkat-ED*⁵⁶ developed by Rabbany et al. (2011) analyzes students' activity in Moodle courses and visualizes relations of participants in discussion forums using social networks analysis techniques. The software was developed in the Java programming language and uses Moodle backup files of courses to generate visualizations. The software was evaluated based on a case study in 2010 and the authors conclude that the integration of the software can support the fair evaluation of students' participation in online courses (Rabbany et al., 2011). The software does not contain any license information but is available for download from the project page. It was updated the last time in 2012 and is only compatible with outdated Moodle versions up to version 2.

3.3.6 Implementations without published Source Code

There are more analytics and reporting tool mentioned in literature. Unfortunately, a lot of the software is only described without releasing the code base itself. These implementations are briefly described in the following paragraphs, but excluded from the comparison, as I was unable to find detailed information regarding the implementation or code base.

⁵⁴ <https://github.com/ldseinhardt/moodle-dashboard>

⁵⁵ Available for download: <https://www.uco.es/kdis/research/software/>

⁵⁶ Project page with information regarding code and installation: <http://www.reirab.com/MeerkatED/>

Student Centered e-Learning Environment (SCELE)

The *Student Centered e-Learning Environment* (SCELE) is an extension of the Moodle learning management system by the faculty of computer science of large public university in Indonesia (Santoso et al., 2018). It enhances Moodle by adding multiple visualizations for instructors. The development was based on a context and user analysis based on interviews with potential users as well as a literature review. SCELE features various visualizations regarding monitoring, analyzation, and assessment of different activities in Moodle. It features a dashboard as entry point with more information available on linked pages. Figure 18 shows a screen capture of the course overview page.

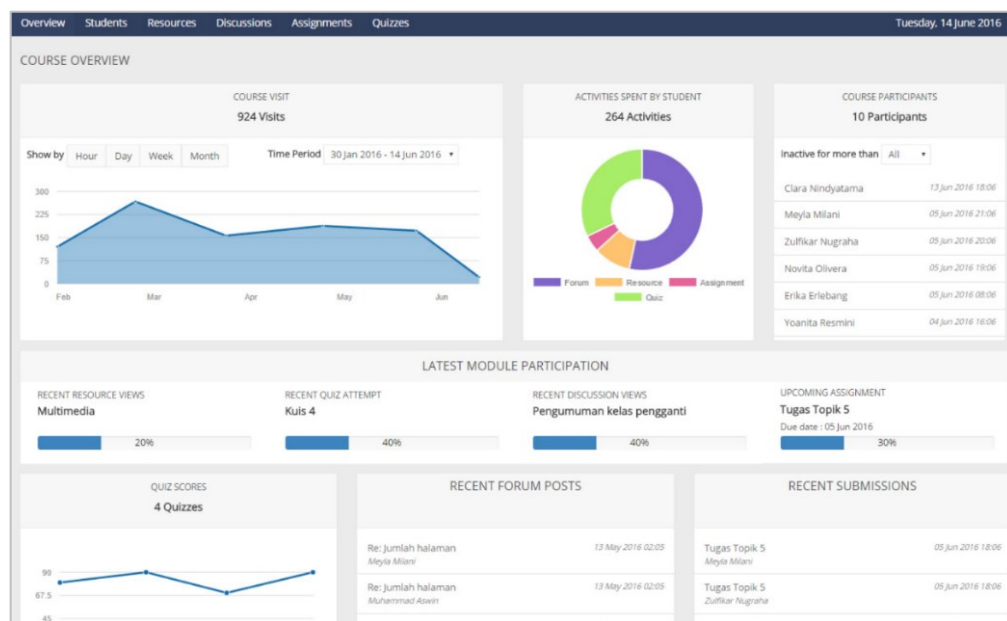


Figure 18: Course Overview page of SCELE (Santoso et al., 2018)

The development of the software has been iterated multiple times based on user evaluations (Junus et al., 2015). The most recent evaluation (Santoso et al., 2016) conducted the usability of the software and was evaluated using the *User Experience Questionnaire* (UEQ). It concluded that the software scored well regarding pragmatic quality aspects (efficiency, perspicuity, dependability) while receiving neutral scores for hedonic quality (stimulation and originality).

CVLA

The software *CVLA* by Dragulescu et al. (2015) extracts data from the LMS database to run an offline analysis outside of the Moodle platform. The results are sent back to the LMS via a REST interface. The resulting visualizations are presented in form of a report plugin. The software provides a variety of different reports, including a heatmap to show user interaction, network analysis graphs, and other visualizations based on the JavaScript visualizations library *D3.js*.

Academic Analytics Tool

The software *Academic Analytics Tool* (AAT) developed by Graf et al. (2011) is implemented as Moodle plugin using a more technical approach focused on extraction and collection of data. The software provides a user interface for defining custom queries. The user first selects which

attributes he is interested (i.e. quiz grade and course name) and applies filters to limit the resulting data set. The software then generates an SQL query, which is shown to the user before it is run against the Moodle database. The resulting data set can be viewed online as table or downloaded as CSV file for further offline analysis.

CosyLMSAnalytics

One of the first approaches towards analytics for Moodle was developed by Retalis et al. (2006). They created *CosyLMSAnalytics*, a tool to analyze log files of Moodle. It is a Visual Basic application that reads the log file of Moodle and provides a variety of Analytics tools in a graphical user interface. The user can monitor events of individual students, courses, or activities. The software is also able to perform a social network analysis.

CIECoF

CIECoF (*Continuous improvement of e-learning course framework*) developed by García et al. (2009) provides a recommendation system for course content created with authoring tools. It extracts data from the Moodle log and uses association rules mining to pre-process and analyze the data. In the next phase, the results are classified to identify the “interestingness” of the results. Afterwards, a knowledge base is created which then returns recommendations for authors proposing possible modifications to the course content to improve the effectiveness of the course. The software was implemented in Java. The implementation uses a client-server model based on an implementation that reads the data from the Moodle course and a user interface to be used by the lecturers.

ViMoodle

Another software is *ViMoodle*, developed by Aguilar et al. (2008), which is a standalone Java-based application. It does not integrate into the Moodle interface but instead connects to the Moodle database and downloads the log data to create visualizations. The use cases are focused on network graphs and user relationships.

Data Mining Tools

In addition, to the presented tools, there are tools that have been developed to evaluate courses and course content. These tools focus on the data extraction from Moodle to run an offline analysis, often for educational data mining purposes. These tools focus on the extraction of data and their results and do not provide a user interface.

One example is a software developed by Conijn et al. (2017) that predicts the performance of students in Moodle courses. Another software developed by Petropoulou et al. (2014) serves as assessment tool for criteria-based assessments. *MoodleMiner* (Akçapınar & Bayazit, 2019) focusses on the data mining of Moodle logging data and also does not provide a user interface for lecturers or students. Similarly, a project by Kadoić and Oreški (2018) analyzes the Moodle logs. An older approach was by Zafra et al. (2013) who developed a data mining software that indicates which activities learners should conduct to pass a course. Lastly, a Java software developed by López-Zambrano et al. (2018) creates data sets based on Moodle logs to be processed by other data mining tools.

3.3.7 Summary

The previous subsections listed all software implementations related to learning analytics for Moodle. In Table 5, I compare the tools, excluding implementations without published source and implementations without a user interface (like the xAPI stores). A filled circle indicates a *Yes*, an empty circle a *No*. The half-filled circle represents that the item is partially fulfilled.

The columns list the most important features for each software. After listing the name of the software in the first column, the list the year of the latest release of the software (as of April 2021) is noted. The year is colored in gray in case the software has not received an update in the last two years making it easier to distinguish recently updated software from non-maintained software. I excluded the dates for software that is part of the Moodle core. For the commercial software LearnerScript, there was no information available stating when the latest release was published.

The next three columns indicate the target audience of the software. I clustered the groups into lecturers, students, and coordinators and/or administration. The last group was merged as the tools targeting study coordinators expect them to have administration rights on the platform.

The following four columns contain the analysis regarding the functionality of the software. The column *Open Source* indicates whether the code base of the software was released under an open-source license. As I exclude implementations without published source code from the comparison, no software is closed-source. Nevertheless, some implementations only release parts of the code as open source. As the Moodle software itself was released as GPL, all Moodle plugins are also released as open source as required by the license. *IntelliBoard* and *SmartKlass* are marked as partially fulfilled, as both release the client-side software under GPL, but the server-side software, necessary to operate and use the software is not released. Similarly, *Meerkat-ED* publishes the software online, but the software itself does not contain an open-source license making the terms of usage unclear. The plugin *Web Analytics* was marked as partially open source as the software itself only acts as a bridge between an external software and Moodle. Therefore, while the plugin itself is open source, the actual used software (for example Google Analytics) might not be open source.

The column *Moodle integration* indicates if the software is integrated into the Moodle user interface or if users need to download the software to run it or log into another website. Following, I indicate if the software can be extended by custom reports or visualizations. For some tools, I marked this column as partially fulfilled, as the software allows users to setup custom reports or visualizations, but users are still bound to multiple limitations inside the user interface.

The last column (*Own logging*) indicates whether the software uses its own data store to log the events or whether it uses the Moodle standard log. For LearnerScript, I cannot say for sure if an external system was used, but based on the description on the website, I figured that the internal log of Moodle was used. The half-filled circle indicates that the Moodle log was used as data source but the logging data was that transferred into another format to be used for the reporting system.

Table 5: Comparison of analytics tools for Moodle

Tool	Latest release	Lecturers	Students	Coordination/ Administration	Open Source	Moodle integration	Extendible	Own logging utility
Core Functionality								
Moodle Course Reports		●	○	●	●	●	●	○
Moodle Analytics		●	○	●	●	●	●	○
Report Plugins								
Overview statistics	2021	●	○	●	●	●	○	○
Course Usage Statistics	2020	○	○	●	●	●	○	○
Monitoring of learning plans	2019	○	○	●	●	●	○	○
Events Graphic Report	2014	○	○	●	●	●	○	○
Forum Graph	2016	●	○	○	●	●	○	○
Generic Plugins								
Configurable reports	2020	●	○	●	●	●	●	○
Analytics graphs	2018	●	○	○	●	●	○	○
Web Analytics	2020	○	○	●	◐	◐	◐	●
SmartKlass	2017	●	●	○	◐	●	○	○
GISMO	2014	●	○	○	●	●	○	◐
MOCLog	2012	●	○	○	●	◐	○	◐
Engagement Analytics	2014	●	○	○	●	●	○	○
Commercial software								
IntelliBoard	2021	●	●	●	◐	●	◐	●
LearnerScript*	N/A	●	●	●	◐	●	◐	○
Other Implementations								
LeMo	2015	●	○	●	●	○	●	◐
Moodle Analytics Dashboard	2017	●	○	○	●	○	○	○
MDM Tool	2014	○	○	●	●	◐	○	○
Meerkat-ED	2012	○	○	●	◐	○	○	○

* LearnerScript: Statements regarding the open-source license and logging are assumptions.

Although I presented over 30 implementations in this section, the number of maintained implementations for Moodle is rather small. Of the presented plugins, only five have received an update in the last two years. Of these five plugins, only two (*Overview statistics* and *Configurable reports*) address the target group of lecturers. Other tools require administrative rights. Of these two tools, Configurable reports requires the user to have a deep technical understanding. Overview statistics provides six reports in total, only one for lecturers. Overall, the number of free tools that can be used in practice is therefore very limited.

In contrast to that, the commercial solutions were very feature-rich. Especially *IntelliBoard* provided a variety of features with user-friendly dashboard solutions but also more detailed and customizable reports. Interestingly, IntelliBoard was also the only solutions (excluding Web Analytics) that comes with its own logging system. All other implementations relied of the existing standard logging of Moodle to create reports. The implication of this are discussed in further detail in the Analysis in Chapter 5 as the built-in logging system of Moodle (Furukawa et al., 2017) logs personalized data and cannot considered to be data privacy-friendly.

3.4 Learning Analytics Evaluation

To assess a software regarding usefulness, usability, and impact on learning and teaching, evaluations need to be conducted. In the following, multiple evaluation strategies are discussed as they were presented and used in literature. This includes evaluation strategies based on feedback from experts, mostly conducted before the final release of the software, as well as feedback from actual users.

3.4.1 Evaluation through Expert Analysis

The evaluation of the software through experts does not require the software to be available to the actual end users. That makes this evaluation method particularly useful during the development phase. The idea is to ask experts to identify parts of the software that are likely to cause difficulties when presented to users (Dix et al., 2003).

Heuristic Evaluation

Heuristic evaluation was introduced by Nielsen and Molich in 1990 (Nielsen & Molich, 1990). It uses reviews from a small number of experts to identify usability problems of interfaces. Based on ten given heuristics, the experts evaluate the system and note the severity of any discovered problem. The 10 proposed heuristics are (Nielsen, 1994):

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

Studies showed the effectiveness of the method, as very few evaluators are able to identify the majority of usability problems in an application (Hollingsed & Novick, 2007).

Cognitive Walkthrough

Cognitive walkthrough by Wharton et al. (1994) is a method in which an evaluator is given a set of tasks to fulfill in order to provide detailed feedback. The process comprises a preparatory phase and an analysis phase. Four things are required to do a Cognitive walkthrough: a specification (or prototype) of the system, a description of the task the user shall perform, a written list of actions needed to complete the task and an indication of who the users are (Wharton et al., 1994). Based on this list, the evaluator then tries to fulfill all tasks. For each individual task, the evaluator answers the following four questions (Wharton et al., 1994):

- Will the user try to achieve the right effect?
- Will the user notice the correct action is available?
- Will the user associate the correct action with the effect they are trying to achieve?
- If the correct action is performed, will the user see that progress is being made toward the solution of their task?

Compared to heuristic evaluation, this evaluation method requires more preparation and post-processing and is in general more time consuming.

Pluralistic Walkthrough

In contrast to the previous two methods, the *Pluralistic walkthrough* by Bias (1994) incorporates representative users as well as developers and product members in the process. The method is based on five characteristics (Hollingsed & Novick, 2007):

1. Inclusion of representative users, product developers, and human factors professionals.
2. The application's screens are presented in the same order as they would appear to the user.
3. All participants are asked to assume the role of the user.
4. Participants write down what actions they, as users, would take for each screen before the group discusses the screens.
5. When discussing each screen, the representative users speak first.

This method is a mix of evaluation based on expert feedback as well as user feedback. It has the benefit to incorporate the feedback of real users while incorporating experts to discuss the results. This combination has led to a widespread adoption of the pluralistic walkthrough. It is widely used for assessing usability and often used as the standard practice (Hollingsed & Novick, 2007).

3.4.2 Evaluation through User Participation

While the previous methods are mostly applied while a software is in development, as soon as a prototype or the final product is available, it is more beneficial to incorporate real users. Observation techniques like the thinking-aloud method can be applied to smaller groups of users. Other techniques like surveys and questionnaires are more suitable for larger groups of users.

Thinking Aloud

The *Thinking-aloud* (Lewis, 1982) method encourages users to think aloud while using the software. Before the start of the evaluation, the user is given a set of tasks asked to complete. An observer writes down what the user says and does without interpreting the actions. Evaluation sessions might also be recorded on audio or video allowing an in-depth analysis of the evaluation later on. The observer might ask questions during evaluation to clarify actions or to elaborate what or why an action was conducted. Based on the notes, the developers can identify problematic or unclear features of the software.

Questionnaires

Questionnaires and surveys are typical tools for evaluating software. In contrast to previous methods, questionnaires are less flexible: they are planned and if done anonymously do not allow further questions in the same go. Therefore, questionnaires need to be well planned and designed. One positive aspect is that questionnaires allow getting a complete picture of all users. The collection of numerical values makes it possible to quantify the occurrence of a trend or to discover trends based on human input. Depending on the number of filled responses, the result is less subjective than other methods with less participants. However, as a negative, a higher number of participants is needed for successful results.

Specific to usability, there are many questionnaires defined in literature. Table 6 lists the most commonly used questionnaires used to evaluate aspects of software quality ordered by their date of publication. Some of the listed questionnaires are technology agnostic as they were developed before the web as we know it today existed while other focus on the use of websites. The number of used items ranges from 2 items up to 50.

The reliability of each questionnaire is based on Cronbach's alpha stated in the publication. Cronbach's alpha (Cronbach, 1951) is an indicator for the internal consistency of items. The value ranges from 0 to 1. Higher values indicate a better reliability. According to George and Mallery (2003), a value above 0.9 is considered "excellent", a value above 0.8 "good" (George & Mallery, 2003). Values below 0.5 are considered "unacceptable" (George & Mallery, 2003).

There is a lot of discussion in the literature about the use of usability questionnaires. Many standard questionnaires have been criticized and are controversial (Berkman & Karahoca, 2016; Lewis, 2013). As a quantitative measurement of their usage in literature, I included the number of citations in Table 6 based on the citation count according to the *Web Of Science*⁵⁷ to indicate how widely used each questionnaire is.

⁵⁷ <https://webofknowledge.com/WOS>

Table 6: Questionnaires measuring usability and quality of software

Abbrev.	Name	Source	Measures	Items	Reliability	Citations*
QUIS	Questionnaire for User Interface Satisfaction	Chin et al. (1988)	User Satisfaction	27	0.94	551
PSSUQ	Post-Study System Usability Questionnaire	Lewis (1992)	System Usability	16	0.94	98
SUMI	Software Usability Measurement Inventory	Kirakowski and Corbett (1993)	Software Usability	50	0.92	259
SUS	System Usability Scale	Brooke (1996)	System Usability	10	0.92	3,604
WAMMI	Website Analysis and Measurement Inventory	Kirakowski and Cierlik (1998)	Website Usability	20	0.90	38
UPWQ	User-Perceived Web Quality	Aladwani and Palvia (2002)	Perceived Quality	25	0.92	530
UEQ	User Experience Questionnaire	Laugwitz et al. (2008)	User Experience	26	0.80	343
UMUX	Usability Metric for User Experience	Finstad (2010)	User Experience	4	0.94	128
UMUX-LITE	Usability Metric for User Experience Lite	Lewis et al. (2013)	User Experience	2	0.82	52
SUPR-Q	Standardized User Experience Percentile Rank Questionnaire	Sauro (2015)	User Experience	8	0.86	33
UEQ-S	User Experience Questionnaire - Short Version	Schrepp et al. (2017)	User Experience	8	0.85	48

* Citation count of the publications according to the *Cited Reference Search* of the *Web Of Science* (March 2021)

By far, the most popular questionnaire is the *System Usability Scale* (SUS) by Brooke (1996). It is based on ten items using a Likert scale ranging from 1 (Strongly disagree) to 5 (Strongly agree). The calculation of the final score is done by subtracting 1 from each normalized item score and multiplying the sum with 2.5. The final value ranges between 0 and 100. It is important to note that SUS scores are not percentages. The average SUS score is 68 (Sauro & Lewis, 2016). Bangor et al. (2009) even developed a grading system for SUS as shown in Figure 19 making it easy to interpret SUS scores. The SUS is also well established for non-English speakers (Finstad, 2006). Several translations are available (Bargas-Avila, 2001; Rummel, 2015). This thesis makes use of the German translation by Bargas-Avila (2001).

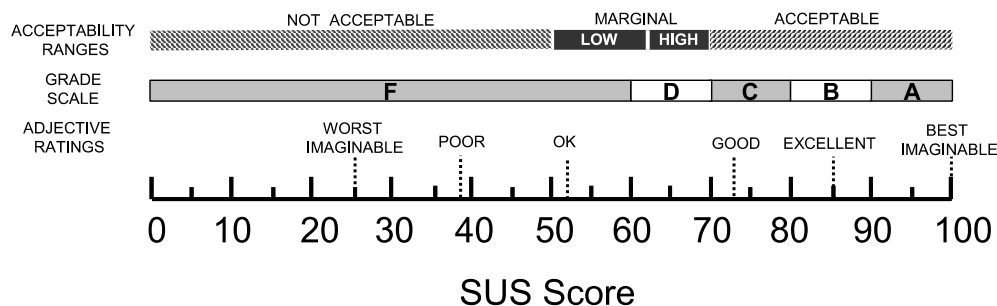


Figure 19: Assessment of usability based on the SUS score (Bangor et al., 2009)

In addition to the questionnaires listed above, a questionnaire specific to the evaluation of learning analytics software has been proposed by Scheffel (2017). The final version (4th iteration) of the *Evaluation Framework for Learning Analytics* (EFLA-4) features two questionnaires: one for learners and one for teachers. It consists of three dimensions (data, awareness & reflection, impact) with eight items each.

3.4.3 Practice

In the following, I list the most common methods used to evaluate learning analytics applications. Most learning analytics implementations are evaluated regarding multiple dimensions. As explained above, one common aspect is the evaluation of usability. Another one is the evaluation of features and their perception of usefulness. Multiple evaluations were conducted by using a quantitative approach based on questionnaires while others focused on qualitative interviews.

For more context, many evaluations incorporate status quo questions or questions regarding the course or lecturers. This might include the size of the course or the overall topic. In some cases, even the profession (professor, teaching assistant, student helper) was asked (Dyckhoff, 2014; Einhardt et al., 2016; Elkina et al., 2014).

Most usability evaluations were conducted using the System Usability Scale. The resulting SUS scores for learning analytics tools ranged from 53 to 83. An indicator editor tool received a score of 53 based on 12 participants (Muslim et al., 2016). The Student Activity Meter (SAM) received a score of 73 based on five participants (Govaerts et al., 2012). A mobile analytics application received a score of 76 based on feedback of 36 students (Tabuenca et al., 2015). An evaluation of a learning analytics application for Open edX platform received a score of 78 based on feedback of 40 students (Ruipérez-Valiente et al., 2017). The best evaluation was received by the Moodle Analytics Dashboard (Einhardt et al., 2016) with a score of 83 based on 15 participants.

The usability evaluation for the Student Centered e-Learning Environment (SCELE) was done based on UEQ, a 26-item questionnaire.

Questionnaires were also the tool of choice when individual features or metrics of the software were evaluated. A standard Likert scale was often used to rate the importance or usefulness of features (Einhardt et al., 2016; Elkina et al., 2014). Other evaluations incorporated the use of semi-structured interviews with lecturers (Arnold, 2010; Santoso et al., 2016).

Many evaluations were conducted using a mix of different methods. Some evaluations analyzed the logged data to gather information about time and frequency of use (Lukarov, 2019). The software Course Signals was evaluated based on surveys of students in addition to follow-up focus groups (Arnold & Pistilli, 2012). Student Activity Meter was evaluated using a variety of different methods. Interviews with participants and thinking-aloud tests were conducted. In addition, the SUS questionnaire was used to evaluate the usability (Govaerts et al., 2012).

4 Preliminary Work

Before the start of the implementation of the learning analytics service, two prototypes were developed as part of previous projects and initiatives. These were developed before the RWTH Aachen University switched to Moodle. Therefore, both implementations had a different purpose and focus. Nevertheless, both implementations influenced the design of the final learning analytics implementation.

In the first section of this chapter, the initial prototype is described, which covers details about the logging architecture of Moodle. In addition, performance data of students was merged with activity data from Moodle. The resulting visualizations are presented. The section ends with a short evaluation of the prototype.

The second prototype explains how a learning analytics application can be integrated into Moodle. The developed software uses an external server to provide learning analytics. During the development of the projects, evaluations showed that it was possible to integrate the external server into Moodle to simplify the data exchange. This led to the creation of the final implementation as explained in the following chapters. I conclude the chapter with a short explaining of the lessons learned from the development.

4.1 Prototype 1: Data Sources and Processing

The first prototype was created as part of the project *AIX Future Teaching and Learning* funded by the *Donors' association for the promotion of humanities and sciences in Germany*. The goal was to create a learning analytics software for Moodle that supports descriptive analytics based on Moodle log data. The software was mainly focused on providing analytics to the Academic and Research Department Engineering Hydrology (LFI) of RWTH Aachen University.

The prototype was based on the *Contextualized Attention Metadata (CAM)* model by Schmitz et al. (2011) and featured a flexible event logging system. It was not integrated into Moodle via the plugin system and was instead installed by adding the code to the Moodle source code directory. No direct integration into Moodle was implemented.

The evaluation of the software showed problems regarding the performance of the database queries. The used data model was not able to handle millions of data points in an adequate time. Using this approach, the implementation of real-time statistics would have been impossible. Another problem was that data privacy was not included well enough as core requirement when developing the software, ultimately leading to the creation of the second prototype.

In the following sections, the event handling and data model is presented. Afterwards, the implementation is outlined and the practical results are shown. Finally, the evaluation is explained.

4.1.1 Moodle Logging and Events

Moodle features a rich and flexible logging system that allows any internal dispatched event to be written to a data store. The internal logging system underwent a major overhaul in Moodle version 2.7⁵⁸. Previously, the old logging system often led to unorganized data making it hard to process the written log files. Version 2.7 was released in 2014, but the old logging algorithms are still present in today's Moodle releases due to backwards compatibility. Due to that reason, some Moodle pages do still use the old logging system. At the start of this project, most logging data was stored in the old logging format, making it necessary to support both log stores. Therefore, I briefly explain both log stores in the following.

Moodle Legacy Logstore

The legacy logstore saves its data in the database table called *log*. Figure 20 shows the simplified database schema. The *event* entity contains multiple attributes. The most important ones are the time when the event was triggered, the IP address of the user, the type of action that was executed and the URL, which was called by the user. Each log entry also references the user, the course and its course module in which the event was triggered. The user table contains more information about the user like his username and mail address. The course contains information like the name and format of the course. The corresponding course module contains information about the activity that triggered the event. An example could be a resource, which the user downloaded, or a quiz. The course module and the course entities are linked to more tables as indicated in the figure.

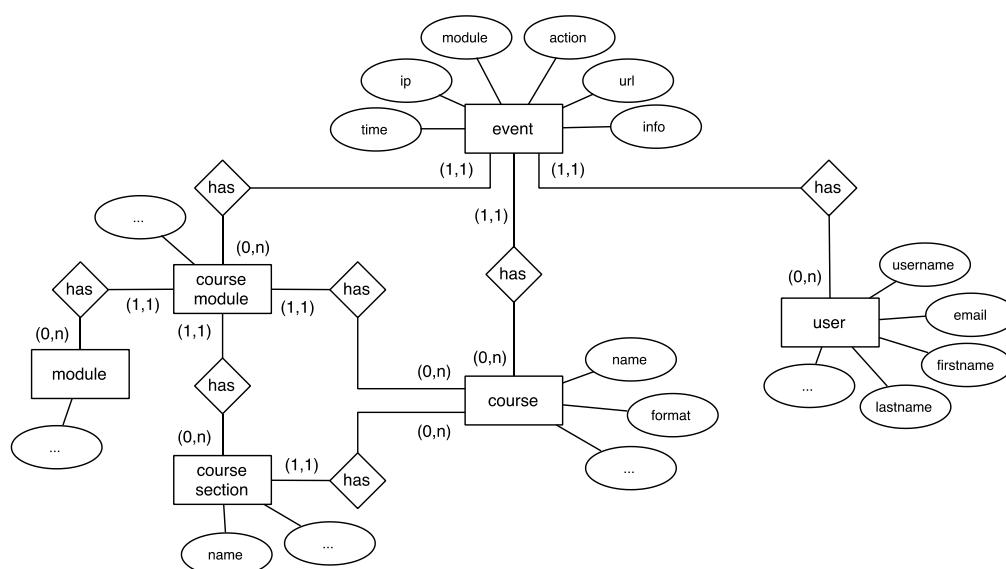


Figure 20: Moodle Legacy Logstore

⁵⁸ For more information why the logging system was changed see: https://docs.moodle.org/dev/Events_API

Overall, Figure 20 gives a good impression why the logstore was considered not suitable for logging. It has multiple connections to other tables which itself need to be merged with data from even more tables.

Moodle Standard Logstore

The new logging system called *Standard Logstore* has changed the way Moodle is storing events. It will be referred to multiple times during the remainder of this thesis. Figure 21 illustrates the database schema. In contrast to the old logging approach, the event entity itself stores more information of the event in its own attributes (e.g. *name*, *component*, *action*). The attribute *crud* indicates whether the action was a *create*, *read*, *update* or *delete* operation. *edulevel* stores information about the level of education received by the students. The full list of attributes can be found in the Moodle developer docs⁵⁹.

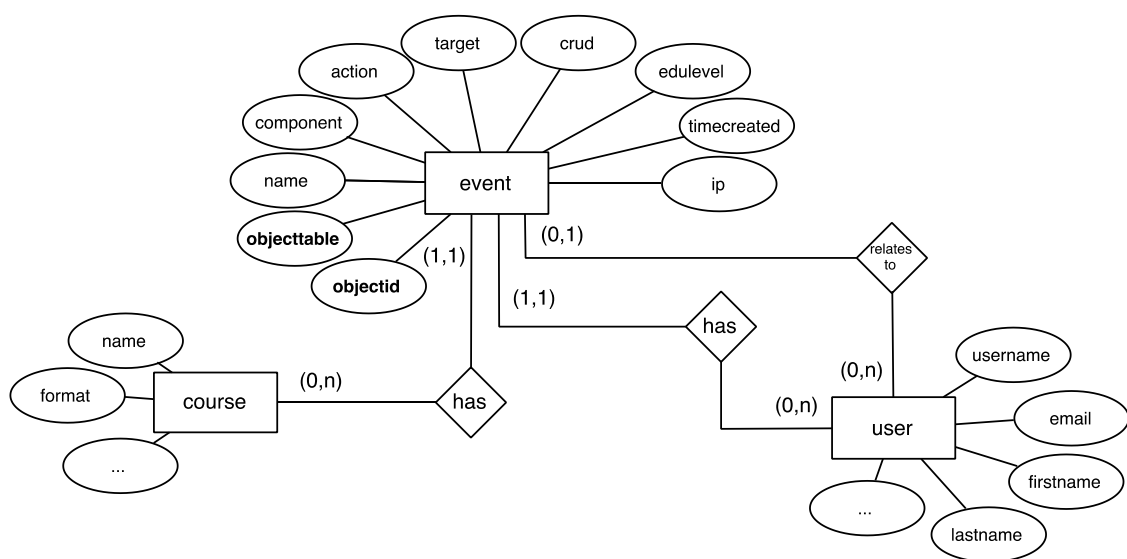


Figure 21: Moodle Standard Logstore

Like the old logging system, the new logging system links the user, which triggered the event and the related course. In addition, each entity not only stores the user triggering the event, but also the user related to it. As an example, if user *A* changes the grade of user *B*, user *A* is saved as the user who triggered the event and user *B* is recorded as the user who was related to that change.

The most important change lies in the way, the event entity refers to other entities. In the old logging system, all information related to an event needed to be stored as part of the event in the object itself. The new log is much more flexible. It allows referencing any combination of database table and row identifier. Therefore, the event log can reference any object existing inside the database of Moodle. This allows the logging to be flexible while at the same time not duplicating data.

⁵⁹ Events API in the Moodle developer documentation: https://docs.moodle.org/dev/Events_API

4.1.2 Data Model

Moodle uses a complex data model for its logstore, as outlined above, which does not work well for applications focused on creating reports containing correlation graphs or aggregated data. Therefore, the following data models were evaluated before the start of the development:

- *Contextualized Attention Metadata* (CAM) schema by Schmitz et al. (2011)
- *JSON Activity Streams* defined by a consortium of IBM, Google, Microsoft, and multiple other big software companies (Snell et al., 2011)
- *Paradata specification* (DR, 2011) building on *JSON Activity Streams*

CAM was chosen as the most suitable model as the other models focus on streams of activities. In the following, the model will be presented in more detail.

Contextualized Attention Metadata

Contextualized Attention Metadata (CAM) was designed by the *Fraunhofer Institute for Applied Information Technology FIT* to monitor user interactions within learning environments (Schmitz et al., 2011). Since its creation, the model design has been iterated and changed multiple times. The used model was based on the latest (5th) version shown in Figure 22. It stores events similar to Moodle while at the same time offering a more flexible solution for storing additional data. The data is stored in the main entity *event*. Only the name of the event, the time the event happened and the sharing level are stored as part of the event entity. The name of the event is a string like *login* or *view resource*. The sharing level depicts which types of users are allowed to access the event. For example, a value of *Public* would represent that everyone is allowed to view the event.

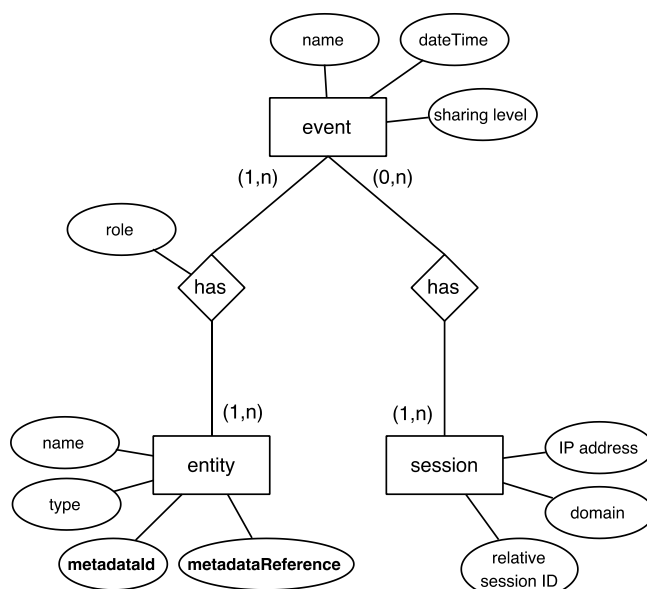


Figure 22: Contextualized Attention Metadata model (Schmitz et al., 2011)

Each event can have an attached session, which depicts how the user triggering the event accessed the learning management system. Therefore, the domain of the LMS, the IP address of the user, and a relative session ID are saved as part of the session entity. The session event is optional as an LMS might also produce events automatically without the need of a user.

The property named *entity* is able to store all kind of information related to an event. Examples are information about users, courses, or forum topics. An entity has the attributes *name*, *type*, and stores *metadataId* and *metadataReference* as a reference to further data. The reference contains the name of a database table and the id of the row. This is similar to the Moodle attributes *objecttable* and *objectid* of the entity event and allows to reference any kind of data. In addition, each entity related to an event has a role. An entity might be used with different roles. An example is a person that sends a message in the LMS to another user. An example event is shown in Listing 2.

```
Event -> name=message_send; dateTime=$TIMESTAMP;
        sharing_level=private
Entity -> role=sender; type=person;
        reference=(users, $SENDER_USER_ID)
Entity -> role=receiver; type=person;
        reference=(users, $RECEIVER_USER_ID)
Entity -> role=message; type=message;
        reference=(messages, $MESSAGE_ID)
Entity -> role=community; type=context;
        reference=(community, $COMMUNITY_ID)
```

Listing 2: Representation of an event in CAM

In the example in Listing 2, the event contains information about the type of action, in this case *message_send*. The attached entities represent information about sender, receiver, the message, and the context, which would be the community or course in which the message was send. The attributes *metadataId* and *metadataReference* are in this simplified example combined to the *reference* attribute. Because of the reference, which can be stored in an entity, the CAM schema is flexible and integrates well into an existing application, where database tables are already present.

Altered Model

While it is easy to transfer existing Moodle data into CAM format, some information required by the CAM model cannot be easily extracted from the Moodle logs or is not necessary. As an example, it is not necessary to give each entity a *name* information as the Moodle tables already contain sufficient information about each referenced object. The name attribute is therefore not necessarily needed when transferring data from Moodle to CAM. Additionally, the sharing level which indicates who is allowed to access the data is not available in Moodle as the access to data is handled on the application level.

Similarly, the sessions of Moodle users are not referenced from the log table, but it is possible to identify session based on the log data by using the time component. As each action of the users is known it is easy to define a session as a number of events from the same user as long as the events are not interrupted by more than a predefined time span (like one hour).

One problem that arises with all data transferred into CAM representation is, that their focus is only placed on the event. All additional information is packed into entities, which are referenced by the event. Therefore, additional database queries used to extract that information are needed to get the information about the events. A simple question like "How many students are enrolled into a course?" therefore needs to query all related events to a course. It is not possible to query only the number of users. As my application should support real-time queries, I enhanced the

CAM schema to make it easier to query metadata like the number of users and other course related information.

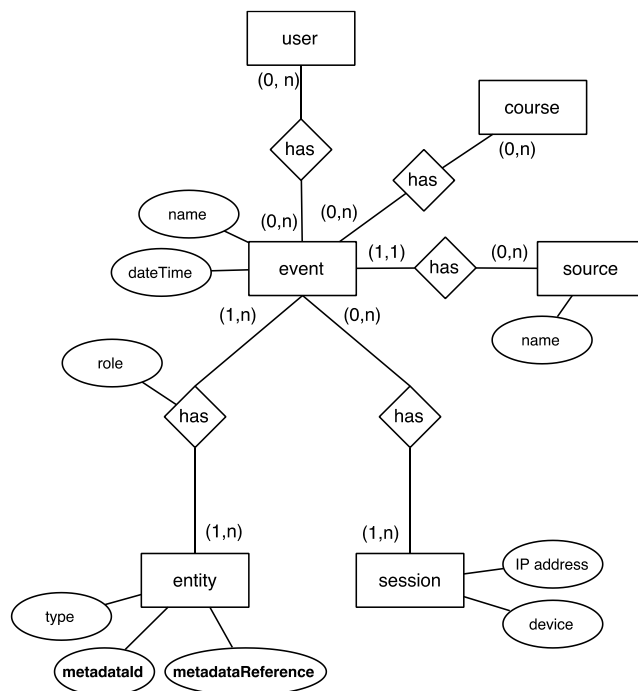


Figure 23: Data model of the first prototype

Figure 23 shows the final data model. My model adds the entities *user* and *course*. This makes it easy to query the database for information related to the course or to query events from a user. These metadata events are not produced by the learning platform but can be used to add information to either a user or a course. As an example, an event named *META_USER* could attach multiple entities via the entity relation to a user. This allows adding information like exam grades without the need that the user entity needs to have an attribute for the value. Likewise, the event *META_COURSE* can contain additional information related to a course, e.g. the term in which the course took place.

I also removed some of the properties from the CAM schema as discussed above. First, I removed the attribute *name* of the entity *property*, as it is unnecessary as explained earlier. Second, I removed the sharing level for each event as currently only administrators of the LMS are allowed to view database records. I also stripped the session object for its *domain* and *relative session ID* attribute but replaced them with other information: Instead of referencing the domain in each session, I reference a *source* entity from the event. The source has a name specifying where the piece of information came from. This might be a domain name or a unique identifier. Additionally, I introduced the attribute *device*, which tracks how the user was accessing the LMS when the event was triggered. That way, one is able to tell a session which was opened from a mobile phone apart from a session which was opened from a desktop computer.

4.1.3 Implementation and Visualizations

The implementation was done in the programming language of Moodle, PHP. As this was the first prototype, the user interface was not integrated into the Moodle user interface itself. The visualizations were displayed on a separate web page instead, not linked to from the Moodle

courses themselves. As the prototype was developed for courses of a German university, the following screen captures do also feature German content.

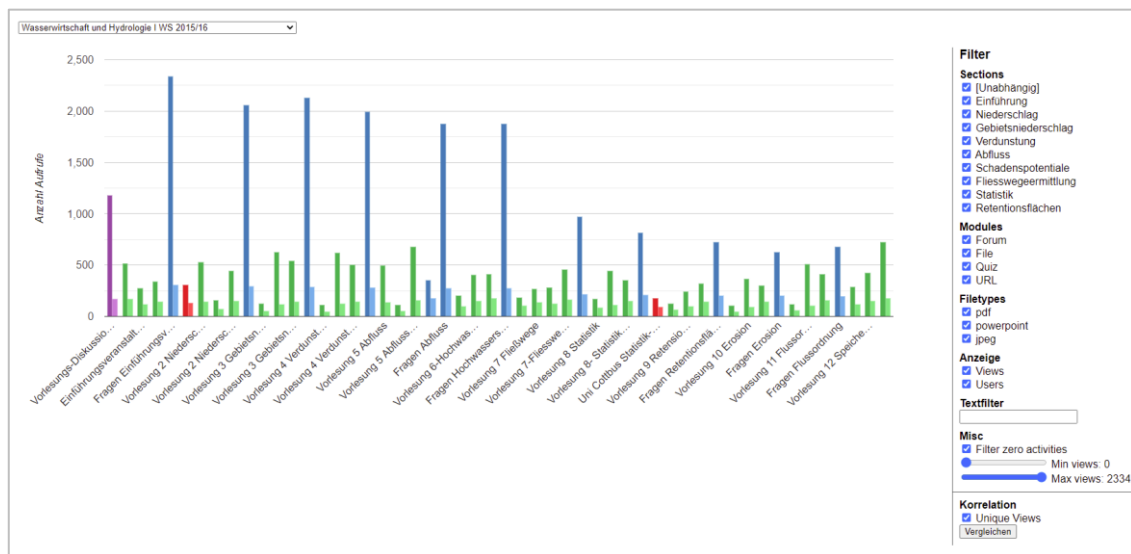


Figure 24: User interface of the first prototype

Figure 24 shows the user interface of the prototype. The visualization was based on the *Google Charts*⁶⁰ library. For more complicated scatter plots (as shown later in Figure 27) the library *HighCharts*⁶¹ was used.

The software used *REST* as base for client-server communication (Fielding, 2000). On the top, a drop-down list contained all Moodle courses. After selection, a JavaScript function would load the course data. For further data processing, the interface on the right allowed to apply multiple filters like specifying the module or file type. This kind of filtering was applied on the client-side without the need of further requests to the server.

The software featured a mode to show correlations to exam grades. By clicking the button on the bottom right as shown in Figure 24, a file dialogue opens in which a user can open an Excel sheet. That file was not uploaded to the server, but processed inside the browser using the library *xlsx.js*⁶². That way, the sensitive files that contained exam grades of the students were not uploaded to the server. Figure 25 shows the user interface. After selecting the Excel sheet, the JavaScript code presented a preview of the Excel sheet in the browser and allowed the user to select which columns contained grade information and which ones included information necessary to identify the student to merge the data with the LMS log data.

⁶⁰ <https://developers.google.com/chart>

⁶¹ <https://www.highcharts.com/>

⁶² <http://sheetjs.com>

First Sheet														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2	Bachelor ...													
3		vorname	nachname	bewertung										
4		Demo Vor...	Demo Nac...	100										
5		Demo Vor...	Demo Nac...	500										
6		Demo Vor...	Demo Nac...	230										
7		Demo Vor...	Demo Nac...	230										
8		Demo Vor...	Demo Nac...	400										
9		Demo Vor...	Demo Nac...	NE										
...

Daten	Spalte	Setzen
Vorname	B	Setzen
Nachname	C	Setzen
Note/Wert	D	Setzen

Statistik erzeugen

Figure 25: Interface to select names and grades

After adding the grades to the data, a visual representation of the correlation between grades and course activity was calculated and displayed. The representation adds a small amount of random noise to each value. This technique is called jittering (Jacoby, 1997) and improves visualizations having many superimposed data points. Figure 26 shows the resulting visualization. Each point represents one student. The x-axis represents the exam grade, the y-axis shows the number of times the student has accessed the course. The red bar visualizes the trend line. In this example, the trend line indicates that there is a correlation between the number of times a student accessed the course and his exam grade: The more students used the system, the better they performed in the exam.

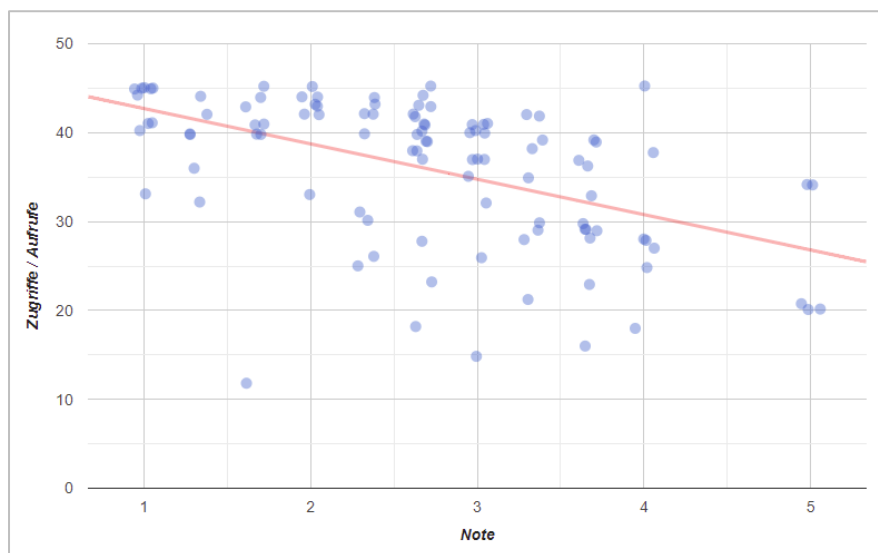


Figure 26: Correlation between exam grade of the course and course activity

In addition to the client-side visualizations of the data, which was developed as data privacy-friendly solution, a server mode was implemented. It made use of the presented data model and stored performance data of students on the server. This part of the software behaved similar to the software presented so far, but in addition uploaded the data to the server where it was stored

in the database. This way, more data could be stored, as the browser was not the limiting factor. In the following, the results are presented.

4.1.4 Results

The department of Engineering Hydrology (LFI) has been using Moodle for multiple years, prior to being used at RWTH Aachen. Over the time, multiple courses have been given for which the new standard logstore and previously the legacy logstore of Moodle had been enabled. Multiple of the following visualizations are built on the data. The usage of the data was discussed with the lecturer and students. The intermediate results were handled pseudonymously. The final visualizations and reports were anonymized.

LFI Moodle

Before the data from the LFI Moodle instance was used, a script was written to transform the data from Moodle's logstore into the chosen data model. The script read each logged data row from the Moodle logs and created the corresponding event and entity relations. In addition to Moodle data, exam results were added to the database. For this, the script was extended to support multiple data sources.

The first data source was the Moodle logs. The second data source consisted of Excel sheets with exam data. Based on the presented implementation, the script would match user data based on first and last names and try to find the corresponding matching data from the exam sheets. For this, the algorithm would use multiple different strategies. For example, a common occurrence was that students mixed up their first and last name when registering on the platform. Therefore, the algorithm would additionally check for that. In addition, many students would leave out any middle names, which were therefore not used for comparison. Lastly, for users where no matching entry was found, the *Levenshtein distance* algorithm was used. The Levenshtein distance is able to measure the difference between two string sequences (Yujian & Bo, 2007). For example, the strings *abc* and *a1cd* have a Levenshtein distance of two as two actions are needed to transform the first string into the second⁶³. Based on the distance names with a distance of two or less were merged. In total, over 98% of all users from the exam grades file could be assigned the corresponding user object. For data protection and privacy reasons, the real name of the user was removed after merging was complete.

The resulting visualizations are shown in the following figures. Figure 27 shows the visualization as scatter plot of the number of learning days on the x-axis and number of total events on the y-axis for each student. The color indicates the grade of the student, where dark green represents the best possible grade and the dark red represents failed exam attempts. Each of the small points represents a single student, while the larger bordered circles represent the average for each grade class. The size of each of the bigger circles represents the number of students with that grade.

⁶³ The two actions needed to transform the string *abc* into *a1cd* are (1) the substitution of „1“ for „b“ and (2) the insertion of „d“ at the end.

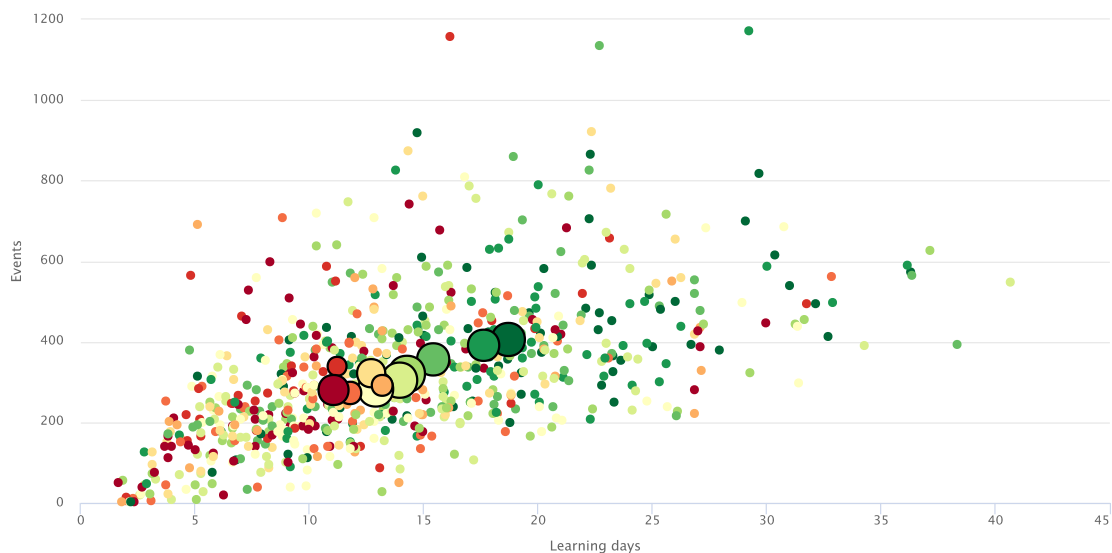


Figure 27: Visualized correlation of number of learning days, number of events, and exam grade (course 1)

Figure 27 shows a representation for a course that has been held three times (course 1). Each time, between 200 and 250 students participated in the lecture resulting in 725 data points in total. The visualization indicates a correlation, which is mathematically confirmed by its Pearson correlation coefficient (Freedman et al., 2007) as listed in Table 7.

Table 7: Correlation coefficients and p-values (course 1)

Correlation between	Correlation coefficient	p-value
Grade and number of learning days	-0.29	<0.001 (✓)
Grade and number of events	-0.18	<0.001 (✓)

The correlation between the grade and the number of learning days is -0.29 meaning that the more learning days you had, the lower (better in German grading system) the exam grade is. This strength of correlation is weak according to Evans (1996). The correlation between the exam grade and the number of events is very weak showing that the number of dedicated learning days is a better predictor for the outcome of the exam than the raw number of events. The corresponding t-tests (Student, 1908) for both correlations show statistical significance.

Overall, the analyzed data shows multiple facts:

- The best students (grade of 1.0) used the LMS on average on 18.7 learning days.
- The worst students (grade of 5.0) used the LMS on average on 11.0 learning days.
- There is a weak correlation between the number of learning days and grades.
- The correlation between the grade and the number of learning days is stronger than the correlation between grade and number of events.

The visualization allows lecturers to see in a visual presentation if there is a correlation between the course usage and exam performance. By showing the average of grades, it is even possible to spot correlations visually as indicated in Figure 27.

Figure 28 shows the same visualization methodology but applied to a different course. Again, the data of three semesters of the lecture were used. This time about 50 students were enrolled. In total, the sample contains over 151 data points. In contrast to Figure 27, there is no visible correlation in Figure 28. The visualization appears to be random. The averages of each grade are all located somewhere in the middle with no visible pattern.

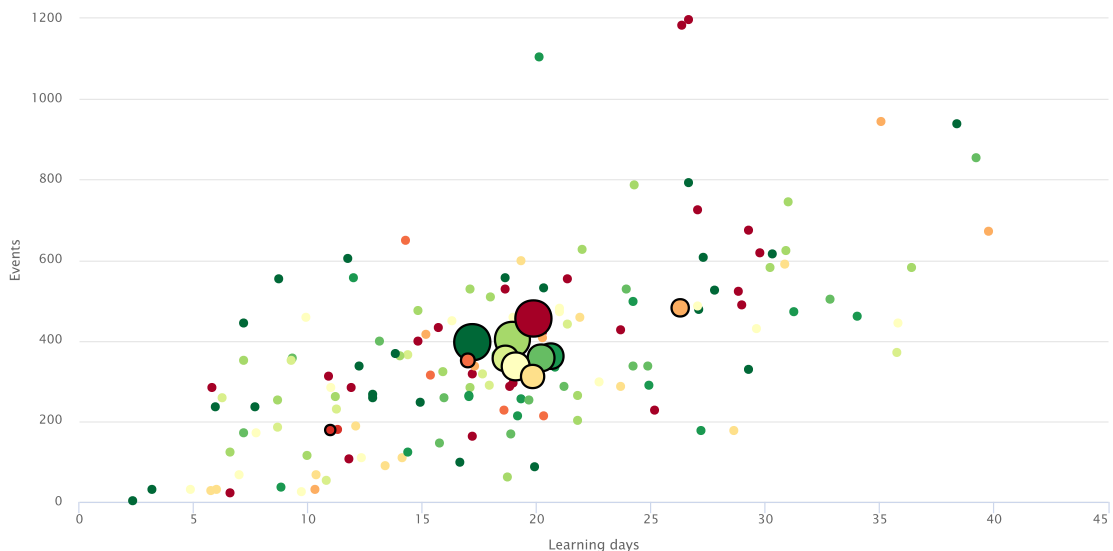


Figure 28: Visualized correlation of number of learning days, number of events, and exam grade (course 2)

Again, the visual representation is confirmed mathematically as shown in Table 8. As the p-values for both correlations indicate, the shown correlation is not statistically significant.

Table 8: Correlation coefficients and p-values (course 2)

Correlation between	Correlation coefficient	p-value
Grade and number of learning days	0.05	0.533 (✗)
Grade and number of events	0.08	0.329 (✗)

Figure 29 shows a different type of visualization based on the Moodle log data. As previously, each data point represents one student. The y-axis, as before, shows the number of events. The color of each data points indicates the exam grade. The x-axis reflects the *average learning day*. This is the average of the time component of all events of a single student. The time component is stored as an integer as UNIX timestamp in the database and contains the number of seconds since the start of January 1, 1970 Coordinate Universal Time (UTC). The average is determined by taking the arithmetic mean and calculating the day from the resulting timestamp.

For students who started learning earlier in the semester, the time component is therefore lower/earlier. Students who started learning later have their average learning time later in the semester. Figure 29 shows the period until the day of the exam (December 20). The data set contains the grades of the first exam of a course consisting of two exams. No fail grades (5.0 at

German universities) were given. As in the previous visualizations, the big data points represent the average for each grade.

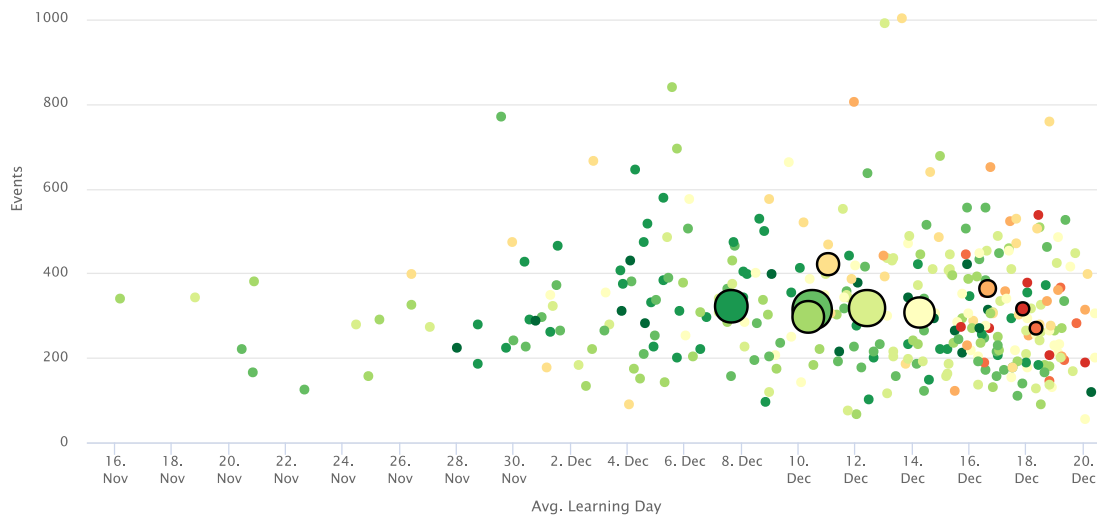


Figure 29: Visualization of average learning day, events, and exam grade

In contrast to the analysis of previous courses, there is no correlation between the grade and the number of events. Instead, the average number of events remains nearly constant for the aggregated averages for each grade. The outcomes were also confirmed by their respective correlation coefficient and p-values as shown in Table 9.

Table 9: Correlation coefficients and p-values

Correlation between	Correlation coefficient	p-value
Grade and average learning time	-0.29	<0.001 (✓)
Grade and number of events	0.03	0.588 (✗)

The findings are summarized below:

- There is no correlation between grade and events.
- The number of events is nearly constant for the aggregated means of grades.
- There is a weak correlation between the grade and the average learning time.

The calculations were repeated for multiple courses and different semesters, focused on the correlation between grade and average learning time. Table 10 shows the statistically significant results for multiple courses of the department of Engineering Hydrology. The tests have also been repeated for other courses, where no statistical significance could be found. These are excluded from the table below. For the courses with a statistical significance, I found correlations with a coefficient ranging from -0.16 to -0.33.

After discussing the results with the lecturers of the courses, I conclude that courses with more interactive learning content (like quizzes) seem to lead to a greater significance of the correlation.

Table 10: Course sizes, correlation coefficients between exam grades and average learning times, and corresponding p-values

Course	N	Correlation coefficient	p-value
Hydrology I – WS 13/14	340	-0.29	<0.001
Hydrology I – WS 14/15	243	-0.16	0.006
Hydrology I – WS 15/16	228	-0.33	<0.001
Hydrology II – SS 15	104	-0.26	0.009

Future Aix – Teaching and Learning

As part of the *AIX Future Teaching and Learning* project, I was able to receive data containing the historical student records of several students. The most significant information included in this data was the final school exam grade (German: *Abiturnote*) of students. As in previous approaches, I had to merge multiple data sources by using names of students. The same script was using incorporating strategies like comparing first names and last names as well as the Levenshtein distance. After merging was completed, names of students were removed.

The created visualization that shows correlations between the school grade, learning days, and exam grade of a course is depicted in Figure 30. The x-axis represents the number of learning days. The y-axis represents the final school exam grade. For privacy reasons, the data containing school grades was clustered into “buckets” of values. The number 1 represents the worst possible grade and the number 6 the best. Jitter was added to prevent overplotting of data points. The color-coded z-axis represents the grade in the examined course, as in previous visualizations.

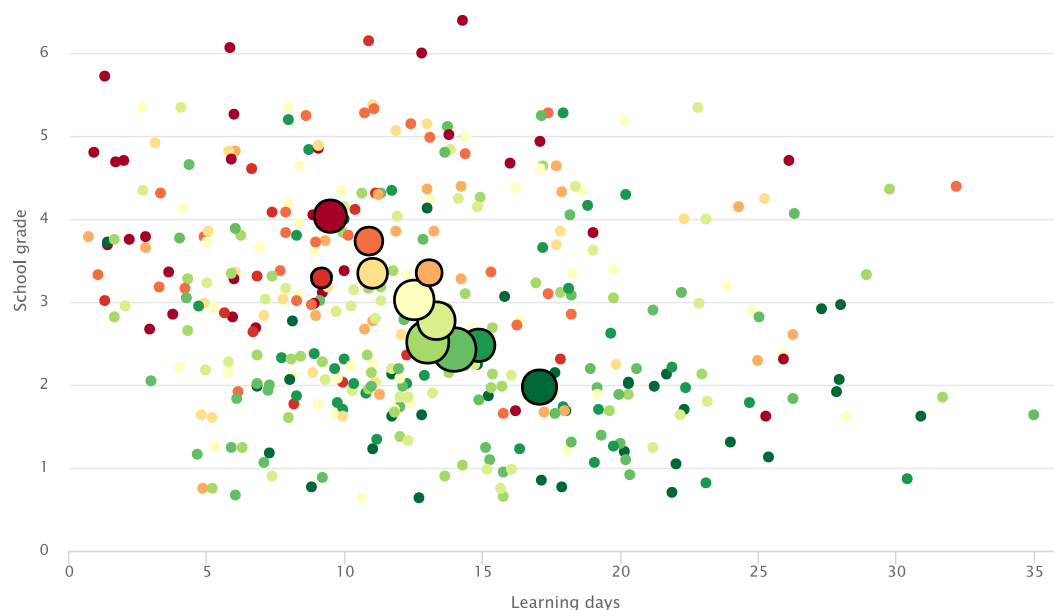


Figure 30: Visualization of student record data, learning days, and exam grade

Multiple obvious correlations can be spotted in the visualization in Figure 30, which are confirmed by their correlation coefficient and t-tests. The highest correlation is found between the course grade and final school exam grade. A correlation coefficient of 0.45 represents a moderate correlation (Evans, 1996). The correlation between the final school grade and the number of learning days is -0.28. Table 11 lists correlation coefficients and their respective p-values.

Table 11: Correlation coefficients and p-values

Correlation between	Correlation coefficient	p-value
Grade and number of learning days (X)	-0.28	<0.001 (✓)
Grade and final school exam grade (Y)	0.45	<0.001 (✓)

Overall, the analyzed data shows multiple facts:

- There is a moderate correlation between the final school exam grade and the grade of the exam of the examined course.
- There is a weak correlation between the number of learning days and the grade of the exam of the examined course.

Due to data protection restrictions, only a limited data set of historical student data was used. In addition, the experiment was only conducted on a single course.

4.1.5 Evaluation

To evaluate the created software on a technical level, I transferred the existing Moodle logs into the specified data model. Administrative events that were not important for learning analytics were ignored. In total, over 22 different event types were transferred from the Moodle logs. I also added metadata including the exam grades and course terms. My tests included the data from six different courses each containing information of roughly one hundred students. All events could be transferred without issues. However, based on the data model, I discovered one technical issue: The querying of the data.

The used SQL queries turned out to be very complex and hard to optimize. Filtering across the event table would have required an index on the event name, which would have meant that a lot more data space would have been necessary. The simple data model made writing the database queries very complex and hard to optimize regarding the runtime behavior. The reason for these complex data queries was that specific events had many entities attached. Even with an index, many database joins happened between the entity and event table. The system as planned would likely struggle with displaying real-time data for millions of logged data entries.

In addition to the technical evaluation, lecturers were asked for feedback on the provided visualizations. The easy to understand visualizations were praised. The correlations between exam grades and behavior of students in the course room showed great potential and interesting research results might stem come from further analyzation with the taken approach. Nevertheless, one major issue was data privacy regulations. As part of the project, I was able to rely on exemption defined in the privacy protection laws allowing to work on data that I could have not relied on if the goal was to release the software as a service for lecturers (Chassang,

2017; Shabani & Borry, 2018). As this was a central role of this dissertation, the focus was changed for future software creations to incorporate data privacy and protection as central issue from the start, not relying on exceptions for research purposes.

4.2 Prototype 2: Integration into Moodle

Based on the results of the first prototype, the second prototype was focused more on the actual adoption of the software and its usage inside of Moodle. The development was split into two repositories: one implementing the integration into Moodle and the second one providing the server that stores the data and generates the reports.

4.2.1 Data Model

Based on my bad experience with the CAM schema used previously, I used a different approach for the second prototype regarding the data model. The database schema was based on the *logstore_standard_log* table schema from Moodle, but most redundant data was removed. This led to a table with columns for *actionid*, *objected*, *userid*, *courseid*, *date*, and *time*.

Regarding the storage of the data, I followed a common practice, that was adopted by most other learning analytics research projects before (Alonso-Fernández et al., 2020; Lukarov, 2019; Tlili et al., 2017), and used a separate server to store and handle the data.

This data model made the transfer from the Moodle database very easy. The server was setup in a way that it would query the data from the Moodle database in an asynchronous way. By default, the data was loaded once per day. This architecture had the advantage that intensive data mining tasks could be offloaded to another server. At the same time, this approach made it impossible to show real-time data, as the data would be loaded from the Moodle server to the storage server in regular intervals.

4.2.2 Implementation and Visualization

The implementation was split into two repositories. The name of the project was *LAIXMO* (*Learning Analytics for Moodle*⁶⁴).

- **laixmo-server**: The server managing the log database and generating reports
- **moodle-block_learning_analytics**: Moodle plugin for visualization and the bridge between Moodle and the server

The server was written in JavaScript based on Node.js. The Moodle plugin was written in PHP. The development of the Moodle block plugin is described in more detail in the bachelor thesis of Marcel Behrmann (Behrmann et al., 2017). The plugin was responsible for downloading the visualization from the server to display it as block inside of Moodle as well as the overall embedding into the Moodle LMS. The communication was implemented via REST (Fielding, 2000).

⁶⁴ The *AIX* in the name *LAIXMO* refers to *Aix-la-Chapelle*, the French name for the city Aachen, in which the RWTH Aachen University is located.

The data handling and processing was done on the external server. Figure 31 shows the embedding of the block inside of a course. The block on the right shows the total number of accesses over the timespan of the course.

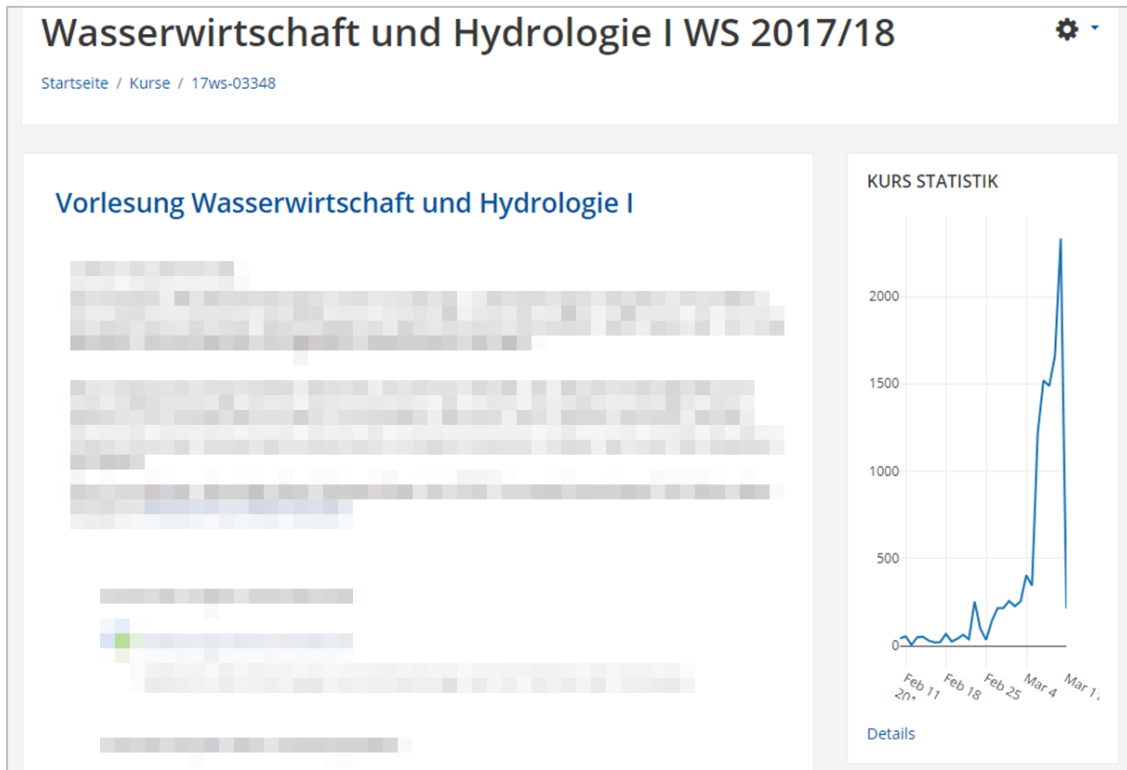


Figure 31: Visualization of course accesses as Moodle block (course content pixelated)

After clicking on the *Details* link, the plugin expands the view and shows a more detailed visualization of the data as shown in Figure 32.

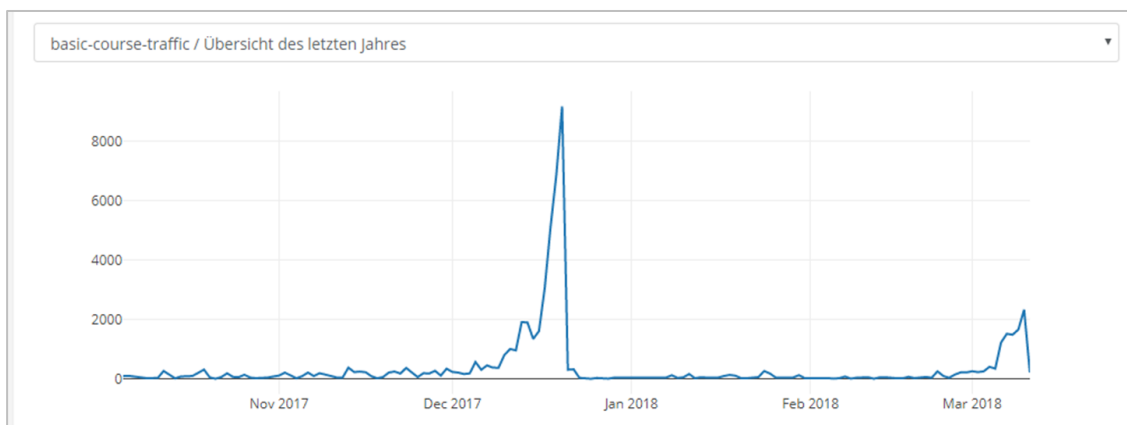


Figure 32: Maximized visualization

The Plotly.js library (Plotly Technologies Inc., 2015) was used for visualization. It features different kind of plot types including line charts, scatter plots, heatmaps, and more. An example of a bar chart is shown in Figure 33. At the top of the chart, more filter options are shown.

Selecting a filter would issue another request to the storage server and visualize the returned data. All computational workload was offloaded to the external server outside of Moodle.



Figure 33: Visualization of student activity in the course room

The server implemented a plugin system that was powered by *package.json* files. These files described each underlying plugin and made it possible to define what kind of visualization is created. In addition, they contain dependencies and caching information. Listing 3 shows an example. Thanks to using *package.json* files, the *Node Package Manager*⁶⁵ (NPM) can be used to manage plugins. Using reports that others have created is as simple as calling “npm install <name of plugin>”, publishing of a plugin to share it with others could be done via “npm publish <name of plugin>”.

⁶⁵ <https://www.npmjs.com/>

```

{
  "name": "laixmo-basic-course-traffic",
  "version": "0.1.0",
  "description": "Shows basic traffic ...",
  "main": "index.js",
  "dependencies": {
    "moment-timezone": "^0.5.13"
  },
  "laixmo": {
    "languageSupport": ["de", "en"],
    "output": [
      {
        "key": "block-30-day",
        "display": "block",
        "name": {
          "en": "Access Count",
          "de": "Anzahl an Zugriffen"
        },
        "targets": ["course", "global"],
        "groups": ["teacher", "student"]
      }
    ],
    "cache": {
      "enable": true
    }
  }
}

```

Listing 3: Example *package.json*

To create a plugin (or report) only two files are needed: The `package.json` as explained above and a file containing the code that queries the database and generates the report. An example is shown in Listing 4.

```

module.exports = async function moduleBasicCourseTraffic({ knex, tbl }) {
  return {
    async run({ courseids, userid }) {
      const rows = await knex.select('userid')
        .count('* as count')
        .from(tbl.events)
        .whereIn('courseid', courseids)
        .where('userid', userid)
        .groupBy('userid');

      const count = rows[0].count;

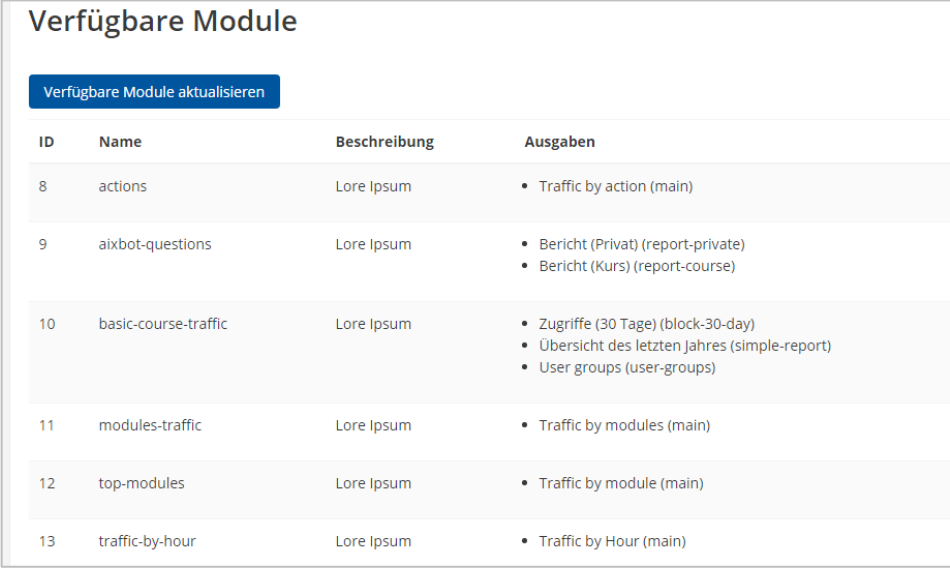
      return {
        title: 'Zugriffe',
        output: [
          {
            type: 'html',
            data: `

## 


```

Listing 4: Example *index.js* returning the number of users who accessed the course

The plugin exports a function that is given access to the database. That function can then query the database and return the data to be visualized. In the given example, the *Knex.js*⁶⁶ SQL query build is used to query the total number of users that have accessed the course from the database. Afterwards, the resulting data is returned as HTML. The surrounding software takes care of data caching, refreshing of data and rerunning the queries automatically. This makes it very easy for researchers and developers to focus only on the reports when creating new reports and visualizations.



The screenshot shows a web interface titled 'Verfügbare Module'. At the top left, there is a blue button labeled 'Verfügbare Module aktualisieren'. Below this is a table with four columns: 'ID', 'Name', 'Beschreibung', and 'Ausgaben'. The table lists seven different visualization modules, each with a unique ID, a name, a placeholder description ('Lorem Ipsum'), and a list of output types.

ID	Name	Beschreibung	Ausgaben
8	actions	Lorem Ipsum	<ul style="list-style-type: none"> Traffic by action (main)
9	aixbot-questions	Lorem Ipsum	<ul style="list-style-type: none"> Bericht (Privat) (report-private) Bericht (Kurs) (report-course)
10	basic-course-traffic	Lorem Ipsum	<ul style="list-style-type: none"> Zugriffe (30 Tage) (block-30-day) Übersicht des letzten Jahres (simple-report) User groups (user-groups)
11	modules-traffic	Lorem Ipsum	<ul style="list-style-type: none"> Traffic by modules (main)
12	top-modules	Lorem Ipsum	<ul style="list-style-type: none"> Traffic by module (main)
13	traffic-by-hour	Lorem Ipsum	<ul style="list-style-type: none"> Traffic by Hour (main)

Figure 34: Available visualizations

In contrast, to the first prototype, this prototype was fully embedded into Moodle and was configurable via the Moodle administration interface. Using that, the server API and authentication tokens were specified. In addition, an administration interface provided information on the different reports a server was able to supply. Figure 34 shows the screen capture of the administration interface showing the visualizations that are supported by the server.

4.2.3 Evaluation

During the development, I discovered that offloading the extensive calculations to an external server was in fact not necessary when database indexes and proper data handling was implemented. Only minor CPU peaks could be determined even when complex data queries were executed. In addition, the external server turned out to be problematic for data privacy and protection reasons. After talks with the data privacy officer of the university, the current situation likely would have not been able to be run outside of research projects. Plans to store log data, user information, as well as exam grades on a central separate server would have been a delicate situation regarding data protection and privacy. Even with proper data protection measures in place, I would have needed to agreement from all students to collect and merge

⁶⁶ <http://knexjs.org/>

data. To solve this issue, I discussed the use of a data privacy mode that would show a checkbox to users in the Moodle interface and ask for consent before processing the data. In the end, the approach was discarded as I evaluated the possibility of getting rid of the server and running the complete process inside of Moodle. No user evaluation was conducted for this prototype.

In addition to the listed problems, there were more downsides to having a dedicated external server powering the reports. The external server would have made it harder for other institutions to use the software. In addition to the Moodle plugin, a second storage server would have been needed.

Another downside of the approach of using an external server was that important information that was easy to query from the Moodle database was not available on the external server. An example is the number of enrolled students in a course. This information is easy to query when using a Moodle plugin but complex to get when relying on logged events. Database queries executed on the external server therefore often had minor issues like visualizations utilizing the wrong number of students when showing how many students were enrolled in a course.

The state of development was presented by me at the *MoodleMoot* conference in 2018. The approach received positive feedback. Nevertheless, several months after the presentation, the decision was made to create another iteration of the software, the final iteration.

In contrast to most other research projects, I concluded that an external server to handle the data was not necessary for the amount of data I was handling. Instead of splitting the code into a server and client, I focused on creating a software for Moodle that was capable of logging data and providing reports without the need of an external server.

4.3 Conclusion

As part of previous projects and research efforts, the presented software prototypes were developed for learning analytics purposes. The first prototype collected log data from the Moodle standard log to create visualizations for individual lectures. Based on additional data, correlations between the performance of students in exams and their use of Moodle were found for multiple courses. For other courses, no link was found between Moodle course activity and exam grades. The results from the first prototype indicate that learning analytics can be used as a tool to support teaching activities and to highlight links between digital teaching activities and students' performance with the goal of optimizing teaching and learning processes.

The first prototype used a data scientific approach to find correlations and create visualizations. The created software was not integrated into Moodle directly. The second prototype was split into two parts: a plugin embedded into the Moodle course room and an external server. The second prototype was never used in production as the technical evaluation revealed a more suitable approach, which was doing the analysis of the data inside of Moodle.

The following chapter discusses that usage of Moodle in more detail while explaining and discussing the used approach for the development of the final learning analytics software.

5 Analysis

In order to answer the research questions from the first chapter, this chapter presents the analysis of related work and preliminary approaches. In the first section, the analysis of related work is presented. Similar approaches are discussed as well as how this dissertation benefits from them. Afterwards, the lessons learned from the preliminary work are presented in Chapter 4. The third section presents the approach taken to provide learning analytics as a service to lecturers. The chosen process model is laid out and necessary requirements from the university are explained. Finally, the functional and non-functional software requirements are presented and discussed.

5.1 Analysis of existing Learning Analytics Tools

In Chapter 3, I discussed related learning analytics architectures and implementations. The analysis showed that there are many theoretical approaches concerning learning analytics architectures but the number of practical implementations is very limited. In fact, only one initiative, the *Apereo Learning Analytics Initiative*, is actively developing software as part of an architecture for learning analytics. Other approaches are either purely theoretical, not easy to adopt as they build on many external service providers and third parties, or not maintained anymore.

Most practical implementations for learning analytics were created as part of research projects or for a specific research purpose. Very few projects aim to provide long-term solutions for learning analytics. Quite to the contrary, for most projects, the resulting data is the outcome. Therefore, little attention is given to the code base itself. Few projects release their source code and even some projects that release the source code online might not contain open-source licenses or are missing crucial information regarding installation and usage.

There are a few notable exceptions. One exception is the *LeMo* project (Fortenbacher et al., 2013). The researchers published all source code online with documentation regarding the installation and usage of the software. The project was even compatible with Moodle. Unfortunately, the project has not been maintained anymore since 2014.

Another exception are multiple Moodle plugins released under the open-source GPL license. Similar to the presented (non-Moodle) research projects, many of the plugins are not maintained anymore, some of which are even impossible to install within recent Moodle versions. Of the discussed software, only few do actual provide meaningful reports for lecturers as many plugins provided visualizations for administrators. The best solutions for learning analytics for Moodle are currently commercial. *IntelliBoard* offers detailed reports for lecturers and students.

Unfortunately, not a single plugin is able to work in a data privacy-friendly way. Most of the plugins use the standard log system of Moodle, which logs personalized data as the logged events contain the reference to the user who triggered the action. In practice, that means it is impossible to use any of the plugins relying on the standard logstore unless users are asked for consent. While IntelliBoard brings its own logging solution, it also logs the reference to users similar to the standard logstore of Moodle. Therefore, all Moodle plugins that I presented can only be used with consent from users.

Another option to consider when implementing learning analytics is the use of xAPI to transfer events from the LMS to an external LRS. There are multiple plugins for Moodle making it easy to connect such a data store. However, regarding the integration of results into Moodle, another software implementation is needed. In addition, as shown in the preliminary work, using an external server comes with other downsides related to the processing and handling of data.

Overall, the number of available learning analytics tools is very limited. Most software is provided by commercial players. Most open-source implementations are not maintained. Based on current data privacy regulations like the GDPR, the user always needs to be asked for consent before using the tools due to the kind of data that is stored. There is no single code base, which could be built on when implementing learning analytics for Moodle. Nevertheless, many evaluations regarding user interface design (especially dashboards) and visualizations can be used when prioritizing features to implement and design.

As the goal of this dissertation is to create software that can also be used by other institutions, the hurdles regarding the adoption of the software should be as low as possible. This can be done by providing good documentation regarding use and installation as well as support for essential questions like data privacy and user documentation.

5.2 Lessons Learned from Prototypes

Based on the two prototypes I came to the following conclusions, which influenced the final iteration of the learning analytics software.

5.2.1 Learning Analytics inside the Learning Management System

One major conclusion from the prototypes was that an external server is not mandatory for learning analytics. Related projects argue that the number of data is too big to handle as part of the LMS or that the needed processing power to generate good visualizations can only be provided by an external server. I observed the opposite: By logging only the necessary data and by properly normalizing the data in the database, only few data records need to be stored. In addition, good optimized database queries make it possible to show data in real-time. Thanks to database indices, even millions of rows of log entries can be processed in few milliseconds.

Another benefit of integrating the learning analytics software into the LMS is that APIs from Moodle can be used directly. As an example: Figuring out how many students are enrolled into a course is not a trivial task for an external record store. Moodle has a complex system that includes managers that are enrolled in every course and additional plugins that might enroll users into courses. Not all of these features might necessarily trigger events leading to a log in the external record store. Therefore, a simple question asking how many users are enrolled in a course might return a wrong response when only looking at the log data. By using the Moodle APIs, the software can be sure to answer the question correct. This principle applies to many

other cases, as the Moodle database contains more information than the information that might be triggered via the event system.

One last benefit of not making use of an external server or record store is that it is a lot easier to adopt the software. When the software is developed as a plugin for Moodle, other interested institutions can simply install the software without the need of the creation of external servers. Thanks to the good plugin system of Moodle, even the support of subplugins to support future use cases is possible. This lowers the barriers for other universities to try out the software and therefore can lead to a greater adoption.

5.2.2 Extensibility

The second prototype made it very easy to create visualizations and reports. The needed files were often only a few lines of code. This has proven very effective, as creating new visualizations and applying them to the productive system was very easy. This also makes it possible to make the software usable and extendible by other researchers and developers. The approach of having a surrounding framework that takes care of heavy tasks like data visualization or data logging while having small individual reports was very effective and needs to be considered for the final software creation.

5.2.3 No Personal Data Collection by Default

The collection of personal data has led to many issues related to data privacy and ethics. During the creation of the software prototypes, many questions related to data protection and privacy were raised. In some cases, it even was unclear what is possible or whether specific use cases are supported based on existing data privacy regulations. The data privacy officer of RWTH Aachen was consulted several times during the development. The discussion on whether to implement a mode that logs personal data has been held multiple times. At the start of the development, the plan was to implement an option to ask students for consent. Unfortunately, the integration into Moodle turned out to be very complex in the prototypes. Moodle supports policies⁶⁷ that users need to agree to before entering the LMS, which would have been convenient to use. Unfortunately, in their current form, these policies are mandatory and user have to agree before entering the LMS, making them useless for this approach.

In general, I found that data privacy was one of the major pain points. Interviews with experts in the field confirm this as stated by Ifenthaler and Yau (2019). Therefore, the decision was made to remove any personal identification from the logs and rely only on anonymous data. Nevertheless, the implemented system needs to be extendible to be used by other researchers or universities so that even use cases including personal data can be supported.

5.3 Learning Analytics as Service

This dissertation started as a research project with the aim to create learning analytics for the department of Engineering Hydrology. In 2019, RWTH Aachen switched from the old LMS *L²P* to Moodle, shifting the focus from providing learning analytics visualizations and reports for individual courses of the chair to implementing learning analytics for the university-wide

⁶⁷ A detailed description of Moodle policies is available on <https://docs.moodle.org/310/en/Policies>

Moodle LMS. This resulted in a change of requirements including functional and non-functional requirements. In addition, there were regulatory differences. For example, the GDPR allows exceptions for handling of data in the context of research projects (Bundesdatenschutzgesetz, 2018, §27) which do not apply when running learning analytics as a service outside of a research projects. In the following, I discuss how the overall learning analytics process looks like when building the software implementation and I explain what else influenced the implementation of learning analytics as a service.

5.3.1 Process

There are multiple process models for learning analytics, which were presented in Section 2.1.4. The basic process of my learning analytics application is based on the process of Cho et al. (2016) who developed the draft for the *ISO/IEC TR 20748-1:2016* standard (ISO/IEC, 2016). The process is illustrated in Figure 7. Based on it, all technical implementations consider and follow respective data privacy regulations presented in Section 2.3. Data protection and its privacy-friendly implementation build the foundation of the created software. Based on that, my implementation starts collecting data based on learning activities in a course room. The data is then processed, stored, and analyzed before showing the results to the user in form of appropriate visualizations. After these steps, the implementation is evaluated and the feedback looped back to the start of the process. During the development phase, the *feedback action* step includes iterations on the software part. Long-term, appropriate steps need to be implemented to support lecturers and learners to incorporate learning analytics into their day-to-day teaching activities.

5.3.2 Institutional Policies

When running learning analytics as service, university-wide policies must be considered. As before, the main requirement here are data privacy regulations. To work out how the application works and what needs to be done regarding data privacy and protection, the data privacy officer of the university needs to be consulted. During this dissertation, the data privacy officer was consulted multiple times. Legal key phrases were in parts written or corrected by the data privacy officer. The whole approach was presented to him multiple times.

In addition to legal requirements, ethical requirements were considered. As my implementation did not feature personalized visualizations for students, it was decided that for transparency reasons all reports and visualizations that were shown to lecturers, should also be presented to students, excluding any sensitive information. The goal was to introduce students to learning analytics without fearing that they might be monitored or controlled.

5.3.3 User Support

When I started implementing learning analytics for a single chair, all participating lecturers were known to me and I exchanged mails with everyone at least once. When scaling up learning analytics to the whole university this is not possible anymore. Therefore, the support of end users must be considered. Users might have questions or complaints that needs to be communicated to a clearly defined channel.

In my case, the *Center for Learning Services* (CLS) and the *IT-ServiceDesk* of RWTH Aachen University were involved in the creation of the software. The ServiceDesk is the central point of contact for students, staff, or anyone involved at the university. Their team was informed

about the software integration into RWTHmoodle and the running case studies as well as the evaluations. Some user questions were answered by the ServiceDesk directly while other requests (mostly feature requests) were forwarded to me. The ServiceDesk was also part of the first group of testers, as they used the software in some of their internal training course rooms.

In addition to the involvement of other parties in the university, I created documentation for users, including textual descriptions as well as screen captures of the created software. While this documentation was first shared by me with participants of the case studies, the documentation was later made part of the university-wide IT documentation.

5.3.4 Infrastructure Requirements

As the software was implemented for the university-wide Moodle platform, I did not operate the software on my own. Instead, the central IT department, the *IT Center* of RWTH Aachen University, who operates the Moodle of the university, ran my software in production. This means, that not just documentation for end users was needed, but also documentation for the IT operators. In addition, some precautions had to be made to make sure the operation of the software did not influence the operation of the Moodle platform as a whole.

One important requirement is that the needed storage space of my software needed to be communicated in advance to make it possible for the operators to make precautions regarding database storage. In addition, it was made sure that as few data as possible was stored to minimize the required storage space.

One benefit of the approach of having a separate party running my software is that I received immediate feedback from the IT operators. That means it was even easier to make the software adoptable by other institutions as IT services of the university were in close contact to me and were able to give feedback directly. This included bugs as well as discussions around the operability and scalability.

5.4 Software Requirements

In the following, I summarize my analysis in form of requirements. The functional and non-functional requirements have been collected through literature review as well as talks to lecturers, Moodle operators, and system administrators.

5.4.1 Functional Requirements

Regarding the functionality of the software, I did not want to limit the created software to specific visualizations or use cases. Instead, the software requirements focus on user experience and integration into the LMS. Therefore, most of the requirements are non-functional. Individual reports or visualizations are not defined in the requirements. Instead, during implementation and iteration of the software, appropriate visualizations are chosen and implemented.

User Interface

The software shall be integrated into the Moodle LMS. Therefore, it has to be a web application. Any lecturer or students, independent of the used operating system or device, should be able to use the tool. The software should therefore be tested in common browsers and devices.

To make the software easy to access for users, the software shall integrate into the Moodle navigation bar. The proper way to integrate into the Moodle user interface would be to make the software a Moodle report. Unfortunately, Moodle reports are practically hidden to normal users, as they require a number of clicks before the user is able to access them (see Section 3.3.2). In addition, these reports might be completely disabled (like at RWTH Aachen) for users. Although, this goes against recommendations of the Moodle developers (MoodleDocs, 2020a), this is the only way to make them approachable and discoverable by users using the system for the first time.

The user interface itself shall be based on ideas and interface design from other learning analytics software. In particular, the use of dashboards as first entry point has often been used and proven as user-friendly approach. A user interface similar to the implementations from *OU Analyse* (Kuzilek et al., 2015) shall be used, with a weekly activity graph at the top, followed by aggregated information below (see Section 3.2.1, Figure 13). Additional pages shall be linked from the dashboard to provide detailed information.

Technically, the software shall integrate into the Moodle LMS using the Moodle plugin structure. It shall be installable like any other Moodle plugin. First, this makes it possible for other institutions to adopt the software. Second, the software is easy to maintain, as it becomes part of the Moodle installation. The software shall use the Moodle administration interface to make it possible to maintain and control the functionality of the software.

Logging & Data Privacy

The fundamental requirement for the software is the compliance with privacy regulations and German privacy laws presented in Section 2.3. Therefore, the data privacy officer of RWTH Aachen University shall be involved in the development of the software. He must be informed about logged data and the development process. For data privacy reasons, the software shall only log anonymized data. That means the ID of users shall not be referenced in the database logging table when storing information about events. This implies that no information can be given regarding any kind of user information. In practice, that results in charts only showing how often a resource was accessed instead of how many users accessed it. In the first prototype, exceptions for research purposes (that also required anonymizing the data after aggregation) allowed the use of personalized data. The second prototype included a data privacy mode that was able to get consent from users. Due to problems with these approaches, the final implementation shall refrain from collecting personal data. As the plugin system of Moodle shall be used, the software shall still provide a way to show other data sources. Nevertheless, by default, no personalized data shall be stored by the developed software.

In addition to legal requirements, ethical principles shall to be considered. Therefore, the created software shall present all analytics results (visualizations, reports) for lecturers to students as well. This might exclude any data that should not be visible to students (like activities that were hidden by lecturers).

In addition to showing the logged data, the software shall use Moodle's internal data to gather relevant information to show. Examples are the number of enrolled students or the list of activities of a course. If that data includes personalized information (e.g. how many students clicked on an activities), the data shall only be shown when at least ten data points are present. This makes it impossible to track individual users.

Administration and Maintenance

The software shall be maintained via the Moodle administration interface to simplify the administration of the software. There shall be multiple options to select whose data is being logged as well as in which courses the learning analytics user interface is shown. These options support the operation of the software for research purposes when only selected users or courses are taking part in a case study. Therefore, the logging administration shall have options to log only the events triggered in specific courses. In addition, the administration shall contain filters to make it possible to only log the data of specific user roles (e.g. students) and exclude any data from other roles (e.g. administrators). Similarly, the administration for the user interface shall contain options to show the links only to the user interface in selected courses.

5.4.2 Non-functional Requirements

The list of non-functional and qualitative requirements are based on work by O'Brien Liam et al. (2007) who defined a list of quality attributes to be considered for service-oriented architectures. The list is similar to other existing literature defining list of requirements attributes (IEEE, 1998; ISO, 2018).

Software Architecture

In the following, I list all features before discussing how each one shall be considered in the context of a learning analytics implementation. The following list is based on work by O'Brien Liam et al. (2007).

- **Interoperability:** Which platforms shall the implementation support and how can interoperability to other data sources be integrated?
- **Performance:** What are requirements for response times and how might the performance of the system influence other components?
- **Security:** How is taken care of access, authenticity, and integrity of users and data?
- **Reliability:** How can the service be integrated reliably and what might lead to an unreliable system?
- **Availability:** What is the proportion of time the system or component might be unavailable for use?
- **Modifiability:** How can new services be integrated or existing service be extended without changing software interfaces?
- **Testability:** How can the implementation be tested concerning edge or error cases? How can problems be replicated in a testing or development environment without affecting the production system?
- **Usability:** How can a good user experience be ensured?
- **Scalability:** How can the implementation scale to an increasing amount of users?

Applying these software architecture questions to the learning analytics software for Moodle yields the following requirements:

- **Interoperability:** The created software shall be compatible to the Moodle LMS. In addition, it shall be possible to extend the system to use external data sources.
- **Performance:** The system shall perform in real-time. Shown data shall contain all recent events and be available at any time. The whole system should perform within milliseconds. Users should not need to wait for longer than a second to see requested visualizations or reports. Possible data filters shall work similarly. All data shall be

visualized within normal request times (similar to any other Moodle page). If, in exceptional cases (e.g. complex filters), visualizations cannot be provided within the desired time, appropriate loading indicators should indicate to the users that data is currently being loaded. Regarding the logging operation, the routines need to make sure that the performance of the whole Moodle platform is not affected by the integration of the learning analytics software.

- **Security:** All data needs to be stored in a secure way. This does not necessary imply that special routines need to be implemented as the Moodle database itself is assumed to be safe. User interfaces need to be secured so that only the corresponding users are allowed to have access. For example, lecturers shall only be able to look at visualizations of their own course and not view data of other courses. Similarly, only administrators of the platform shall be able to view other data. The administration pages shall also only be accessible to administrators.
- **Reliability and Availability:** The system should be accessible and reliable at all times. Created reports and visualizations must be built on the underlying data. Otherwise, e.g. in case of missing or incomplete data, this shall be communicated to the user. As the software shall integrate into the Moodle LMS, it shall become part of it and only be unavailable if the Moodle platform itself is unavailable.
- **Modifiability:** Other developers and researchers shall be able to use the created software and change its functionality in a simple and anticipated way. The implementation shall offer appropriate software interfaces to support the modification and extensibility of it. New capabilities shall not require service interface changes.
- **Testability:** The software shall be developed supporting the operation in different scenarios, including development environments. Therefore, environment variables or similar approaches shall be used to make the software easy to configure and test in local testing environments.
- **Usability:** The software shall be easy to use for beginners as well as advanced users. That means the software shall be easy to discover (especially in comparison to existing Moodle reports) and shall be intuitive to use. Some lecturers might have already used similar systems or have the time to read guides for the software while other users might use the software for the first time without the intention of reading guides. Both user groups need to be considered when implementing the user interface. It shall be easy to discover, easy to use, but still offer detailed reports and visualizations for power users.
- **Scalability:** The software shall scale to ten thousands of users (as there are currently about 45,000 students enrolled at RWTH Aachen). Therefore, the software shall be able to perform logging operations with little overhead compared to the Moodle system itself. In addition, courses might have multiple thousand users (as the biggest course at the university has about 3,000 students enrolled). The software shall therefore scale vertically concerning the scalability of the Moodle system itself. IT operations, like the addition of computing power or storage, should directly help to scale up the system. As the software is implemented as part of the Moodle system, horizontal scaling (scaling across multiple servers) is not necessary.

The above list contains the most important requirements. In the following, additional non-functional requirements regarding the operation and adoption, which resulted from talks with IT operators are given.

Operation and Adoption

To allow adoption for other educational institutions, I spoke to the IT department to setup requirements regarding the operation and integration. The aim of the requirements is to make it as easy as possible adopt the software.

The tool shall be installed, operated, and administrated via the Moodle administration user interface. That means administrators should be able to change configuration and integration with little effort. The system administrators shall be supported in their task of administrating the software by providing documentation for use and maintenance. This also includes best practices regarding software engineering like using proper readme files, changelogs, and other suitable documentation files.

The logged data shall be stored in the Moodle database itself. As the development of previous prototypes has shown, there are data protections issues as well as administrative efforts necessary to maintain an external server solely for the purpose of learning analytics. By default, the software shall therefore use the Moodle database to store all information. This implicates the creation of proper installation, uninstallation, and update routines that take care of modifying the database accordingly.

All created software shall be released as open source under the *GNU General Public License* (GPL). This is required by the Moodle license itself, as Moodle is also licensed under the GPL. The software shall be published online to make it easy for other institutions, as well as users, to check the used algorithms and code base.

Based on these formulated software requirements, the following chapter presents the practical implementation of the learning analytics software.

6 Implementation

The final implementation of my learning analytics software focuses on the adoption of learning analytics, based on the idea of respecting user data while at the same time creating a software that can easily be adopted by other universities. In the first section, the overall software architecture is presented. The created software is split into two plugins: One component is responsible for the logging of data; the second component generates reports and visualizations. Both components are implemented as plugins for the Moodle LMS. The second section explains the implementation of the data logging plugin. Afterwards, the third section presents the user interface of the software, including example visualizations, usability aspects, and the overall integration into Moodle. The operation of the software in practice is explained in the following section. As the software was used in production, software engineering best practices regarding documentation and administration are covered. Afterwards, the implemented visualizations are presented in the fifth section. Finally, the development of the created learning analytics software is summed up.

6.1 Architecture

The implemented software is split into two Moodle plugins. The logging plugin receives events and writes them to the database. The user interface presents the reports and visualizations to the user. The reason for using two plugins instead of a single one is that Moodle is restrictive in what plugin type has access to which internal API. The log plugin uses Moodle's API to log events into the database. In addition, the plugins come with subplugin types that are used to extend the functionality. Prior to its first use, the software has been iterated multiple times based on feedback from lecturers, the data privacy officer of RWTH Aachen, as well as feedback from technical experts in the field (Dondorf et al., 2019).

As the software was developed for Moodle, the primary development language was PHP. As PHP is not well suitable for data processing tasks, many optimizations were integrated to scale the system to thousands of users. The generated HTML output for the frontend is enhanced by JavaScript and the visualization library Plotly.js is used for graphical output.

The overall architecture of the software is illustrated in Figure 35. Components and data of my implementation are visualized using an orange gradient as background while the Moodle system itself and Moodle core functionality is depicted as white. In the following, the flow of data is explained as presented in the figure.

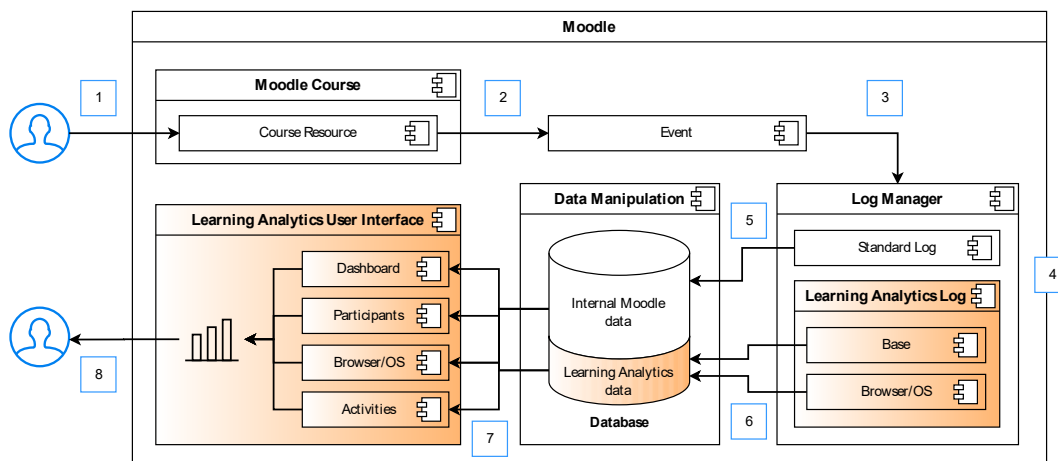


Figure 35: Overall architecture and integration into Moodle (as published in Dondorf et al., 2021)

The process starts with a user requesting a course resource inside a course room. This action triggers an event (1). Internally, Moodle then loads the file and sends it to the user (to keep the diagram simple this part is excluded). By requesting a resource, Moodle internally fires an event that contains information about what action was executed, who executed the action and further information like time, related courses, and objects (2). Moodle’s internal log manager that manages installed logstores then handles the event and passes the event to the installed logstores (4). By default, Moodle comes with the standard logstore that writes all information of the event into the Moodle database (5). The learning analytics logstore then receives the event and removes user-related and redundant data in it. It then passes the event further to its internal loggers, which inserts the data into its own table in the Moodle database (6). To generate the user interface internal Moodle data (e.g. information about the course like the number of enrolled students) as well as the collected logs are queried from the database (7). The queried data is then used to generate resulting plots, tabular data, or plain text information that is shown to the user (8).

The following Section 6.2 describes the log plugin in detail. Afterwards, the user interface is presented in Section 6.3.

6.2 Data Logging

The logging of events is implemented in the logging plugin (*logstore_lanalytics*). The *logstore* plugin type provides an API to be called whenever an event is triggered in Moodle. By default, Moodle comes with a standard logstore plugin that logs all events happening inside of Moodle. As previously explained the standard logstore is not data privacy-friendly and logs sensible data on each request. This includes information like the IP address and ID of the user that triggered the event. For this to be conform to data privacy, the institution has either to get consent from the user or to limit data storage or just store the data for a limited amount of time. As an example, at RWTH Aachen, this data is logged for only 10 days.

On installation, the plugin creates two tables inside the Moodle database containing all log records and their respective event names. To log them, the plugin uses the logstore API of Moodle. It extends Moodle’s internal class `\tool_log\log\writer` that contains the required functions to implement the logging functionality. A simplified implementation is shown in

Listing 5. The function *write* is called for every event triggered inside of Moodle. Then a buffered writer is used to buffer events before writing them to the database in the *insert_event_entries* function, which is called by the Moodle event manager. The first function (*write*) receives the triggered event as argument and adds metadata to the log entry. This contains all information about the event (when it was triggered, the context, what happened, etc.). Compared to the Moodle standard log, additional information about the device of the user is added based on the *User-Agent* HTTP header of the browser. This information is not logged by the Moodle system itself, but it was decided that this was an interesting metric worth logging. As the evaluation in Chapter 7 will reveal, this was not actually the case and the corresponding report was removed in the first iteration.

The operating system and the browser information is determined based on a list of regular expressions (Boda et al., 2011). It differentiates between seven browsers (*Chrome*, *Edge*, *Firefox*, *Internet Explorer*, *Opera*, *Safari*, and a generic *Mobile* category for other mobile browsers). The operating systems are clustered into *macOS*, *Linux*, and multiple *Microsoft Windows* versions (from *XP* to *Windows 10*). In case a mobile device is used, the operating system is reported as *iOS*, *Android*, or simply *Mobile* (in case the algorithm detects a mobile system but is unsure which exact operating system is used). Additionally, two more values are possible for the browser and the operating system: *Unknown* in case the algorithm is not able to detect the OS or browser, *Moodle API* is used in cases in which the event is triggered by an internal Moodle event. It is important to note, that this information is added as it was originally considered to be of interest for later analysis (i.e. how many users used a mobile device) and this is the only logged information that is not provided by the Moodle logging routes. All other pieces of information (like the related course or the time) are provided by the Moodle event system itself.

```
class store implements \tool_log\log\writer {  
  
    protected function is_event_ignored(event_base $event) {  
        // Checks if the event should be ignored (due to a test call, etc.)  
    }  
  
    public function write(\core\event\base $event) {  
        // creates a log entry and add its to the buffer of events  
    }  
  
    protected function insert_event_entries(array $events) {  
        // Writes events to database, respecting all settings  
    }  
}
```

Listing 5: Implementation of the log writer class (simplified)

The created log entry is then buffered so that multiple events can be written to the database at once, putting less pressure on the database. When Moodle decides to write the events to the database (as configured in the settings), Moodle calls the *insert_event_entries* function with all created logging records. That function then reads the configuration of the plugin to filter the logged events before writing them to the database. For example, the plugin might be configured to log only events triggered by students. In that case, events by lecturers would be excluded at this point. Afterwards, the events are written to the database.

6.2.1 Data Model

In contrast to the previous prototypes, the chosen data model was based on the Moodle logging model but removes a lot of redundancy before storing the data. For this, I analyzed the database schema of the *logstore_standard_log*. Table 12 shows the full database schema of the standard logstore table. As explained in the development of the second prototype in Section 4.1.1, Moodle is very wasteful regarding the needed storage space. My analysis of the database table shows which rows in the *logstore_standard_log* table are redundant. Some data is stored multiple times in the same record. For example, the values of the columns *component*, *action*, and *target* are repeated in the column *eventname*. In addition, there is redundant information that can be acquired by making a table join or using a static list. An example is the column *objecttable* that depends on the eventname. Every event with the event name `\mod_quiz\event\attempt_viewed` uses the objecttable *quiz_attempts* making the data redundant as it can be replaced by a simple static list. For each table column, I list whether the information is redundant and give the column that can be used to get the value.

The analysis of the Moodle logging table shows that of the 21 columns in total, 9 columns contain redundant information. Of the remaining 12 columns, some columns are of interest for learning analytics reports. These are *eventname* (what kind of event was triggered), the context or course (*contextid*) in which the event happened, *userid* (who triggered the event) and *timecreated* (when the event happened). Also of interest might be the *objectid*, *relateduserid*, *anonymous* and the IP address depending on the goal of possible reports. The remaining columns only serve a technical purpose or are not of interest for learning analytics.

As explained in the analysis in Chapter 5, I decided to log only anonymous data, removing the *userid* from the stored data. All in all, the following data is logged by the software:

- The type of action (*eventname*), e.g. quiz attempt started
- Time (*timecreated*), exact to the second
- Corresponding course in which the action happened (*courseid*)
- Corresponding context (*contextid*), e.g. the ID of the quiz that was started
- Operating system and browser (not taken from Moodle data), e.g. *Windows 10* and *Firefox*

The *courseid* information could also be gathered based on the *contextid* column, but this would make database queries very slow. Therefore, the value is stored separately. For storage efficiency, all data is stored as integer values. The string identifier of the type of action is being stored in a separate table for space efficiency. Therefore, when an event is triggered, the software first checks if the type of action is already present in the event table. If that is not the case, it adds the name of the event and references the ID of the added row. This technique to reduce storage space and speed up necessary SQL queries is called database normalization (Codd, 2002).

The plugins come with two subplugin types to enhance their functionality. One subplugin type can be used to enhance the logging plugin (*lalog*); the second type is used to provide more visualizations (*lareport*). The first subplugin type allows aggregation strategies (Kimball & Ross, 2011) for optimized database queries to be used. I used this technique to store the browser and operating system for each course in a separate table in the database. That way the database query to build the user interface can use the preprocessed data.

That means the plugin executes two or more database queries for each triggered event and most events are triggered for each request in Moodle. The first database query receives the ID of the event; the second query inserts the related information into a new log row in the logging table. Afterwards, subplugins might create additional requests to aggregate the data.

Table 12: Moodle standard logging database table

Column	Example	Redundant	Meaning
id	1234567	No	Primary Key
eventname	\mod_quiz\event\ attempt_viewed	No	Event name
component	mod_quiz	Yes (eventname)	Related component
action	viewed	Yes (eventname)	Corresponding action in verb form
target	attempt	Yes (eventname)	Target of action
objecttable	quiz_attempts	Yes (eventname)	Table containing data related to this event
objectid	94698	No	ID of the corresponding row in the table
crud	r	Yes (eventname)	Database action (“CRUD”)
edulevel	2	Yes (eventname)	“Educational value”
contextid	10879	No	Context ID, often refers to the related course
contextlevel	70	Yes (contextid)	Context level constant
contextinstanceid	5644	Yes (contextid)	Instance that the context is referencing
userid	20054	No	ID of the user who triggered the event
courseid	74	Yes (contextid)	ID of course in which the event happened
relateduserid	NULL	No	ID of second user, in case another user was involved (like sending a message to someone)
anonymous	0	No	If the action was executed anonymously
other	a:1:{s:6:"quizid";s:3:"557";}	No	Additional meta data
timecreated	1589189158	No	Unix timestamp
origin	web	No	How the event was triggered.
ip	127.0.0.1	No	IP address of the client
realuserid	NULL	No	In case an admin switches role to another user, this contains the user ID of the admin

6.2.2 Aggregation of Data

The logging plugin is extensible by a subplugin called *lalog*. The subplugin is technically a subplugin of the *local* (User Interface) plugin. The decision to make it a subplugin of the local plugin was made to simplify the experience for Moodle administrators. Having all subplugins in one plugin makes it easier to administrate. In practice, the logging plugin uses the *lalog* subplugins.

The logging database tables makes use of database indices. An index is a data structure that can significantly improve the speed of database queries (Lightstone et al., 2010). It uses presorted copies of existing data in combination with pointers to the actual data to provide the base for fast lookups and efficient data access. While an index can speed up specific queries, the use of indices requires more storage space. Therefore, the benefit of a specific index needs to be weighed against the created storage overhead.

The created database tables utilized two database indices. The first index on the columns *courseid* and *timecreated* is important to speed up queries used for historical queries or queries that incorporate a time filter or a grouping clause. The second database index on the columns *courseid* and *contextid* is needed to speed up queries that concern information regarding a specific resource of a course. It is to note that thanks to this multi-column index, queries on the first column (*courseid*) are also speed up.

In general, all queries were optimized to make sure that database indices are hit making it possible to deliver the results of database queries in few milliseconds even for thousands of data rows. Still, in some cases it might be necessary to aggregate the data before making computationally intensive database queries. In this case, developers can rely on the *lalog* subplugin to aggregate the data using aggregation strategies.

The subplugins of type *lalog* are able to create aggregation tables when installed and need to implement one class (*lalog/logger.php*). The *log* function of the class is called for each triggered event and is able to aggregate the event data. A use case for this was the aggregation of browser and OS data in the summer semester 2020. For the use case, the subplugin *lalog_browser_os* was developed.

Table 13: Aggregation table of subplugin *lalog_browser_os*

courseid	platform_desktop	platform_mobile	platform_api	...
1	1500	300	200	...
2	1000	2000	100	...

The needed data for reports and visualizations is already present in the log table. However, the problem is that each time the corresponding report is shown all log entries for the complete course would have been scanned by the database. As the underlying data is not indexed, this is a computationally intense database query. Therefore, a separate database table to aggregate the requests for the used platform was used. An example for clarification is shown in Table 13. In the example, the course with the ID *1* has been accessed 1500 times via a desktop computer, 300 times via a mobile device, 200 times via an API. Accessing the information in this way only

needs one simple database query reading one row instead of a database query aggregating the results of all 2000 log rows.

In general, this aggregation strategy can be seen as a trade-off between the implementation of a full map reduce algorithm and solely relying on the (not aggregated) log data.

6.2.3 Import of Data

During the development of the software, I was in contact to IT operators from my university as well as other universities. Most universities are using the standard logstore to log data. In most cases, I was told that, similar to RWTH Aachen, the data was only logged for a limited amount of time, in some cases only a few days but in other cases even months.

To help with testing and onboarding, I developed a script that is able to import the data from the standard logstore from Moodle into the log table of the plugin based on an ETL (*extract, transform, load*) process. The implemented *Command Line Interface* (CLI) script can be executed by a system operator. It transfers the existing log data into my database table format. Listing 6 shows the output provided to the operator when the help screen is called via “php cli/import.php -help”.

```
Imports data from table `logstore_standard_log` into table
`logstore_lanalytics_log`.

Options:
  -h --help          Print this help.
  --clean            Clean the `logstore_lanalytics_log` table before running.
                    Be aware, that this options deletes all data from the
                    table `logstore_lanalytics_log`. This option should only
                    be used before activating the logstore in the settings.
  --startid=<value> First ID to be imported, leave empty to import all events.
  --pastweeks=<value> Instead of using startid you can use past-weeks to set how
                    much weeks from the pasts should be imported. The importer
                    will ignore all events that are older. Example: Set this
                    value to 3 to import only the logs from the last 3 weeks.
  --batch=<value>    How many logs to be handled in one batch. Defaults to 10000
  --limit=<value>    For testing/development purposes only. This set the max. ID
                    of the row to limit the number of rows to be imported.

Example:
php cli/import.php
```

Listing 6: Output of the import script

The script comes with multiple options allowing limiting the import to only some weeks in the past or to the amount of log records that are imported. The script has been used many times in development for the use of testing data. In addition, multiple universities used the script to import data from the Moodle log table.

6.3 User Interface

The user interface plugin is the larger part of the implementation, containing the visualizations and the integration into the Moodle interface. It is a *local* plugin, as explained in Section 2.2.3 about plugins and plugin types, allowing a deep integration into the Moodle system. The general

layout was based on related research using a similar dashboard design (Kuzilek et al., 2015). The latest version of the user interface is shown in Figure 36.

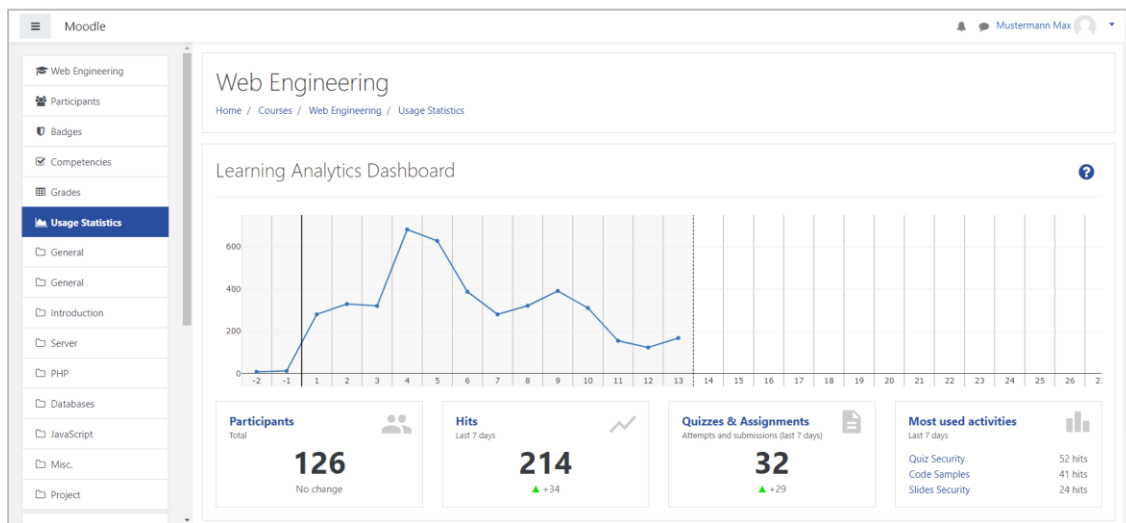


Figure 36: Start page of the user interface

The entry page, the dashboard, contains a plot at the top showing the aggregated access numbers by week. Below, a summarizing number for each remaining subplugin is shown with a link leading to a page containing more information.

The user interface is split into multiple components. Output renderers support the visualizations of tables, plots, and split views. Split views are simple containers that allow grouping of other visualizations horizontally. Figure 48 in Section 6.5.1 shows how the split component is used to divide the page into two parts.

Tables were implemented using simple HTML tables. However, as simple tables are not a user-friendly form of presentation, I introduced *Micro Charts* to enhance the tabular visualization. Micro charts add miniature bar charts below the exact numerical value allowing the presentation of the exact value in tabular form in addition to its visualization. This makes it easy to compare values visually. In addition, colored bars can be used to show the relation of values. The approach to integrate charts into table has been used in the past by other UI frameworks for the web (SAP, 2020). An example is shown in Figure 37. The first column shows the name of a Moodle activity. The remaining columns list multiple metrics using micro charts. The used bars make it possible to intuitively spot differences in the data.

Quiz	Participants	Attempts	Overall average of all attempts	Overall average of first attempts
Quiz Introduction	104	423	89.5%	79.7%
Quiz Server	95	339	84.4%	76.6%
Quiz PHP	101	384	87.3%	83.6%
Quiz Databases	92	301	82.9%	74.0%
Quiz JavaScript	87	287	76.2%	63.9%
Quiz Security	80	273	87.9%	79.0%

Figure 37: Micro Charts

For more complex visualizations the library *Plotly.js* (Plotly Technologies Inc., 2015) was used, which was already part of the second prototype. Plotly.js supports a variety of visualizations including line charts, bar charts, scatter plots, and heatmaps. As Plotly.js is a JavaScript library, an interface was implemented allowing developers to write PHP code to generate the visualizations. The data is then printed to the HTML code and interpreted by a bridge implementation. It reads the output from the PHP code and creates the according visualizations with Plotly.js on the client-side in the browser.

As Plotly.js supports many different types of visualizations, these two output types cover most needed use cases in learning analytics. For more flexibility, I added another output type that allows normal HTML to be printed. Therefore, even complex HTML constructs can be used.

6.3.1 Visualizations

Individual visualizations were implemented based on a second subplugin type *lareport*. Each subplugin automatically receives a page handle corresponding to the name of the subplugin. For example, the subplugin *coursedashboard* is accessible via the following website:

```
MOODLE_PATH/local/learning_analytics/index.php/reports/coursedashboard?course=123
```

That way, developers implementing new visualization pages, do not need to setup a page for the frontend. However, the developer needs to take care of several usability and data protection requirements. For example, the threshold needs to be applied before showing visualizations and reports.

A minimal UI plugin can consist of only two files:

- **version.php**: This is required by Moodle and contains information about the plugin like the name, version, and possible requirements.
- **lareport_[pluginname].php**: This is the actual plugin file extending another PHP class to generate the visualizations.

Below I give an example for both files to show how a visualization plugin can be created. The created sample shows the total number of times a course was accessed. In addition, the plugin uses the plot visualization to create a bar chart for demonstration purposes. I exclude the start of the file that contains the PHP opening tag (*<?php*) and the license information required by Moodle (and the GPL) at the start of the file for simplicity.

An example for the *version.php* is shown in Listing 7.

```
$plugin->component = 'lareport_hits';  
$plugin->version = '2021050600';  
$plugin->requires = '2017111302';
```

Listing 7: File *version.php* of the demo plugin

The file requires only few information. This is the version of the subplugin, the required Moodle version that is needed to run the plugin, and the name of the component, consisting of the type of the plugin (*lareport*) and the name of the subplugin (*hits*).

An example for the implementation part of the subplugin is shown in Listing 8. The file consists of one class that extends the *report_base* class of the UI plugin. This class requires two functions

to be implemented: The *params* function defines which values the plugin expects as input. This is then parsed and validated before being handled to the main function *run*. That function receives the parameter (in this case the ID of the course). It then executes an SQL query to read the number of total hits of the course. For demonstration purposes, the demo plugin creates a bar chart with two values. Finally, the function returns the number of hits as part of a string followed by the generated chart. The resulting page is shown in Figure 38.

```
class lareport_hits extends local_learning_analytics\report_base {
    public function run(array $params): array {
        global $DB;

        $courseid = $params['course'];

        // Read the total number of hits from the log table
        $query = "SELECT COUNT(*) hits FROM {logstore_lanalytics_log}
                WHERE courseid = ?";
        $rows = $DB->get_records_sql($query, [$courseid]);
        $hits = reset($rows)->hits;

        // Add a demo bar chart with two bars
        $plot = new plot();
        $plot->add_series([
            'y' => ['Demo Bar 1'],
            'x' => [30],
            'orientation' => 'h',
            'type' => 'bar',
        ]);
        $plot->add_series([
            'y' => ['Demo Bar 2'],
            'x' => [100],
            'orientation' => 'h',
            'type' => 'bar',
        ]);

        // Return the string and plot
        return ["There have been {$hits} hits in total in this course!", $plot];
    }

    public function params(): array {
        return [
            'course' => required_param('course', PARAM_INT)
        ];
    }
}
```

Listing 8: File lareport_hits.php of the demo plugin

The sample code is kept to the bare minimum. It shows how easy it is to integrate textual as well as visual output. Thanks to the use of Plotly.js, visualizations that are more complex are possible as well. In the example, an SQL query to receive the number of total hits was used. In an actual use case, more complex database queries would be used. Thanks to the integration into Moodle, it is also possible to link data from within Moodle (like the number of students or the course activities) to the learning analytics log data.

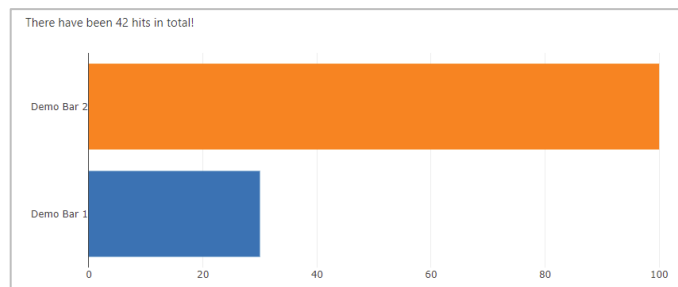


Figure 38: Output of the demo plugin

In addition to the page itself, each report can optionally create a preview box, which is shown on the course dashboard. To show a preview, a *preview.php* file has to be placed in the *classes* directory of the plugin. This file is then automatically used by the course dashboard. Listing 9 shows the file content of the preview file.

```
class preview extends report_preview {
    public static function content(array $params): array {
        $courseid = $params['course'];
        return [
            report_preview::box(
                'previewname', 'Demo', self::icon(),
                'More info text...', 100, 30,
                $courseid, 'linkedreport'
            )
        ];
    }

    private static function icon() { // vector image of a circle
        return '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
            <path d="M12 2C6.47 2 2 6.47 2 12s4.47 10 10 10 10-4.47 10-
            10S17.53 2 12 2z"/>
            </svg>';
    }
}
```

Listing 9: File *preview.php* of the demo plugin

For demonstration purposes, a simple circle was used as SVG icon and the numbers are hardcoded into the code. In a real scenario, database queries would be performed instead. In this example, the helper function *report_preview::box* is used to create a preview. It expects multiple parameters (like shown numbers and linked reports) and returns the corresponding HTML code. The resulting preview as it is shown in the course dashboard is shown in Figure 39.

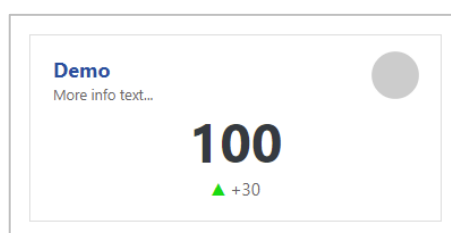


Figure 39: Preview box in the course dashboard based on *preview.php*

This subsection presented the implementation of visualizations as Moodle plugin. The actual visualizations and pages that were used in the field in this dissertation will be presented in Section 6.5.

6.3.2 Usability

A lot of work was put into usability aspects of the learning analytics software. A dedicated help page was introduced before the start of the second case study. The page contained the most asked questions from lecturers and feedback from the first case study. The help page is linked from the question mark on the dashboard page in the top right, as shown in the Figure 36. A screenshot of the help page is shown in Figure 40.

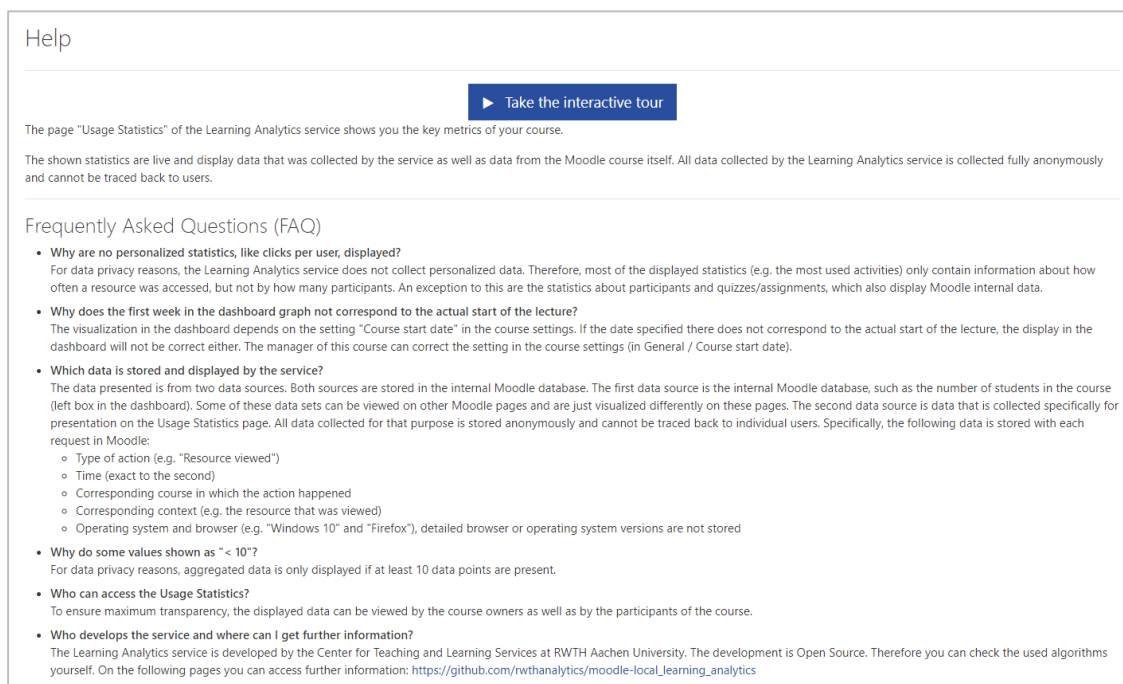


Figure 40: Help page

At the top, the help page contains a big button that uses a Moodle tour to introduce the learning analytics features interactively. Clicking the button takes the user back to the dashboard page and starts a *Moodle User Tour*. User tours are normally utilized by system administrators to introduce users to new features. Moodle contains multiple tours by default for this purpose. During the installation of the local plugin, a user tour for Moodle is installed, which is then shown to the user after pressing the button.

A single step of the tour is shown in Figure 41. The tour uses a step-by-step approach to show multiple different features to the user by highlighting the corresponding parts of the page and showing related information.

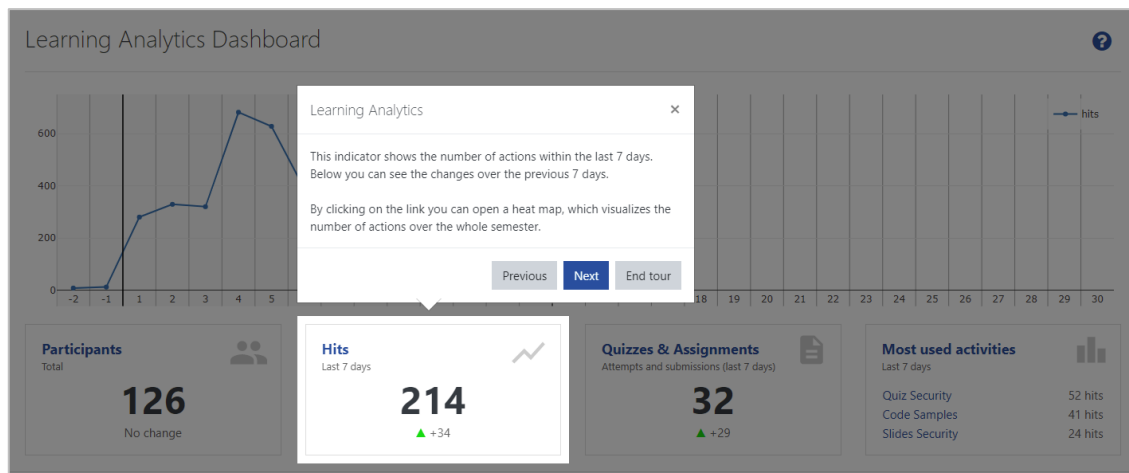


Figure 41: Moodle user tour on the dashboard

Another usability feature was the usage of icons and symbols to support the text where possible. This is visible in the boxes at the bottom on the dashboard where each box contains a separate symbol in the top right corner to make them easy to distinguish visually. As some of the boxes show the difference to the previous week, a green upwards pointing arrow is used to show an increase of value, a red downwards pointing one to show a decrease.

Another possible usability problem was detected during the development: the slow loading of Plotly.js charts. Although, this was mostly only noticeable during the first second of the page load, a loading indicator was added to show that that part of the page is still loading. Therefore, all visualizations that include a plot show a visual loading indicator while the plot is being prepared to show.

6.3.3 Data Privacy and Ethics

As explained in the analysis in Chapter 5, the software logs no personal data. While this makes it impossible to spot actions by individual users, it is still possible to see if an action was performed. In theory, that makes it in courses with few users possible to estimate whether a single user has performed a task (if the number of actions was zero before). Therefore, the user interface implements a data privacy measure to limit what data is visible: Only if at least ten data points are available in any given data set, the data is shown. This value is configurable (set to 10 by default), as data privacy officers of other institutions might find other values reasonable to use. Values below that threshold are labeled as < 10 and shown as zero in plots. On pages that contain aggregated information (like the page *Participants*), the information is even excluded from the list when the threshold is not reached. An example of a threshold that is not reached is shown in Figure 42.

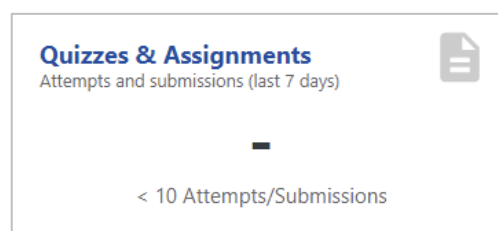


Figure 42: Necessary threshold for the indicator is not reached

One downside of the approach using no personal data is that my solution is not able to show any personalized feedback to students, although this is what students expect from a learning analytics system (Schumacher & Ifenthaler, 2017). To make sure students do not feel monitored when using the LMS, all pages were made available to students as well as lecturers. That means, all data in all pages is available for students. The only exception for this, are visualizations that reference resources that are hidden to students on purpose. This might be course data from a later time of the course, which the lecturer has not yet revealed to the students.

The access rights were implemented using Moodle capabilities. Capabilities decide which user is allowed to see which feature inside of Moodle. The created plugin has a single capability that decides if a user is allowed to see statistics or not for a course in which he is enrolled: *learning_analytics:view_statistics*. By default, this capability works as described above, by giving anyone having access to the course also access to the learning analytics page. Of course, the system administrators of Moodle might decide to change which user has this capability and therefore who has access to the learning analytics results.

6.3.4 Integration into Moodle

To make the learning analytics page easy to use and discover, the link to the main page was directly added to the course navigation. This goes against the recommendations of the Moodle developer documentation, which highly discourages adding navigation nodes to the navigation as it makes the navigation “harder to use” (MoodleDocs, 2020a). The recommendation by the Moodle developer docs is to use the *Reports API* instead which adds the plugin to the list of reports. As explained in the related work of Moodle plugins in Section 3.3.2, the reports are difficult to discover for lecturers. They are practically hidden behind multiple clicks. The decision not to follow the advice of the Moodle documentation was therefore on purpose. The integration into a Moodle course is shown in the Figure 36 at the start of this section. On the left, the course navigation menu is visible. A link to the dashboard page of the software is added after the last link to other activities.

The user interface also makes part of Moodle’s internal data in some cases. As an example, the entry page (dashboard) uses the course start date when displaying historical visualizations. That way, no additional settings are introduced for users. Some displayed visualizations also directly incorporate the data from Moodle database table. For example, the number of students is shown and one visualization shows the courses in which these students are enrolled. This data is stored by Moodle itself and is not collected by the logging tool. The use of the data has two advantages: First, I make sure that shown numbers (like the number of enrolled students) are not different to my own data source. Second, I can use data that is not available from my data source. Although the data is not collected by the software, it must be treated according to data privacy regulations. That means the software only shows aggregated data and applies a threshold before showing it.

6.4 Operation

To make the software easy to use and to increase its adoption, I tried to make it as easy as possible for system administrators to try out the software. The user interface supports two languages, English and German, and has the following features to increase adoption.

6.4.1 Administration

The administration of the plugins is done via the Moodle administration. Both plugins come with multiple different settings concerning the logging of data as well as the visualization. One key feature is that the logging works independent of the user interface. As an example, during the case studies the plugins were configured to log the data of all students while the user interface was only shown in participating courses.

Logging (Logstore Plugin)

The logging plugin embeds into the core of Moodle as it is called for every triggered event. After installation, it is therefore necessary to activate the plugin first. Figure 43 shows the administrative list of installed logstores in Moodle, which includes my created logstore plugin. To activate the plugin the *Enable* icon has to be pressed.

On the settings page for the plugin, the Moodle administrators have the following options to choose:

- **log_scope:** The options are *one of all, include, or exclude*. It defines the scope of the logging process. By default, everything is logged but together with the option *course_ids* it can be set to log data only in specific courses.
- **course_ids:** Together with the first option, the administrator can enter IDs of courses separated by comma to limit the collection of data to these courses (example: *10,153,102*).
- **tracking_roles:** Defines whose data should be logged (whitelist) unless specified via *nontracking_roles* below. By default, the data of all users is logged.
- **nontracking_roles:** Defines whether data from specific roles should not be logged. This is useful if the data of some roles (like managers or teachers) should not be collected.
- **buffer_size:** In case a single page fires more than one event, this is the number of events that is buffered before writing them to database in bulk.

These first two options make it possible to test the plugins in specific courses before rolling it out to all courses. The white and blacklists of roles allow excluding teachers or managers from being logged as operators might prefer to exclude their data from visualizations. It is to note, that this check is only executed once when the data is logged. After the data is logged, it is impossible to find out whether an event was triggered by a lecturer or student as all data is stored anonymously.

Available log stores							
Name	Reports supported	Version	Enable	Up/Down	Settings	Uninstall	
Standard log	Logs, Live logs, Activity report, Course participation, Statistics	2019052000		↓	Settings	Uninstall	
Learning Analytics Log	-	2020102200		↑	Settings		
External database log	Logs, Live logs	2019052000			Settings	Uninstall	
Legacy log	Logs, Live logs, Activity report, Course participation, Statistics	2019052000			Settings	Uninstall	

Figure 43: Learning analytics log in the list of Moodle logstores

User Interface (Local Plugin)

In addition to the logstore plugin, the user interface can be configured to decide which information to show and how. The following options are available in the Moodle administration:

- **status:** One of *show_if_enabled*, *show_courseids*, *show_always*, *hide_link*, *disable*, *course_customfield*. This value sets whether the user interface should be activated and whether a link is shown in the navigation. By default, the link and page are visible if logging is enabled for the course. This option makes it possible to show the user interface only in specific courses. The latest option adds a setting to the Moodle course setting page allowing lecturers to decide themselves if they want to activate the plugin (see Figure 44).
- **course_ids:** To be used together with the *status* option *show_courseids* to show the UI only in specific courses.
- **navigation_position_beforekey:** Specifies where the link to the page is added in the Moodle course navigation.
- **dataprivacy_threshold:** This value determines how many data points a data set needs to include before the data is displayed (data privacy measure).
- **student_rolenames:** In case the role(s) for students/users in a course is not simply *student*, the administrator can specify the corresponding role name(s).
- **student_enrols_groupby:** Option to allow merging of multiple courses with the same *shortname* or *fullname* in the parallel/previously heard courses visualization.
- **setting_dashboard_boxes:** Determines which boxes are displayed in the course dashboard, in which order and how big the boxes are. The specification is in the format *reportname:width*, separated by commas. A line has a maximum width of 12, based on the Bootstrap grid format⁶⁸. Example: ``learners:8,activities:4`` displays two boxes in the dashboard, where the first one is much wider than the second one. The value only needs to be changed if other subplugins are installed or if the layout of the dashboard needs to be changed. The default value is `learners:3,weekheatmap:3,quiz_assign:3,activities:3`, showing all four default reports in a single row.

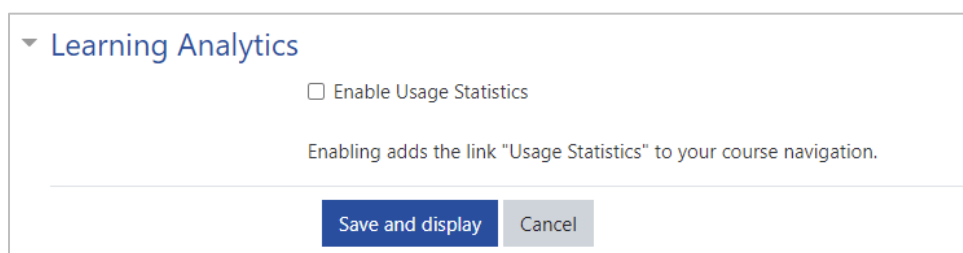


Figure 44: Option to enable the plugin in the course settings page

⁶⁸ Moodle uses the CSS framework Bootstrap. A grid row therefore consists of 12 units according to the documentation: <https://getbootstrap.com/docs/4.0/layout/grid/>

Thanks to these administration pages, both plugins are fully integrated into Moodle, not needing any configuration files or additional installation routines. The operation of the software for system administrators is therefore greatly simplified compared to similar software.

6.4.2 Documentation and Releases

The software is released as open source under the *GNU General Public License* (GPL), the same license Moodle itself uses. *Git* was used as version control system to keep track of changes and allow collaborative work on the source code. The first release of the software was published on the internal Git platform of RWTH Aachen University. Later releases were published on GitHub in the *RWTHAnalytics* software repository⁶⁹. The repository page is also linked to from within the help page of the plugin so that interested users can check algorithms and implementation themselves. This openness makes it possible to have other developers contribute source code if interested. For example, during operation the IT Center, who operated the software in production was interested in a specific feature, implemented it, and contributed the software back to the code repository. Similarly, staff of another university implemented support for a different database management system.

Regarding software releases and versions, I followed best practices regarding software engineering by documenting changes across versions. A single *CHANGELOG.md* file in the repository documents all changes in the Markdown⁷⁰ format since the previous release. The format used is specified by the *keepachangelog* initiative (keepachangelog.com, 2017). The first few lines of an update from January 2021 are shown in Listing 10. The file is much longer and contains information related to the changes of every update of the software.

```
# Changelog
All notable changes to this project will be documented in this file.

## [0.15.0] - 2021-01-19
### Changed
- Minor text changes on the FAQ page
### Fixed
- Test & Quiz pages errors in very specific edge cases for quizzes and assignments
```

Listing 10: Documentation of changes in the changelog file

A *README.md* Markdown file⁷¹ in the repository of the local plugin contains the documentation for both plugins. A single file for both plugins is used to keep all documentation in one place. The README file contains information about installation, configuration, data storage, development, and project information (license and authors).

For documentation purposes, the software comes with a demonstration mode, which can be activated by checking out the *demo* Git branch. The activation does not query data from the database, but instead uses exemplary data to illustrate the functionality. Most of the pictures in this chapter were taken with the demonstration mode activated. As IT operation is often

⁶⁹ Source code of both plugins: <https://github.com/rwthanalytics>

⁷⁰ Markdown is a simple markup language that is easy to read in its source code form.

⁷¹ README content: https://github.com/rwthanalytics/moodle-local_learning_analytics#readme

interested in screenshots of the software for the user documentation, this is a simple way to create screen captures showing the same data.

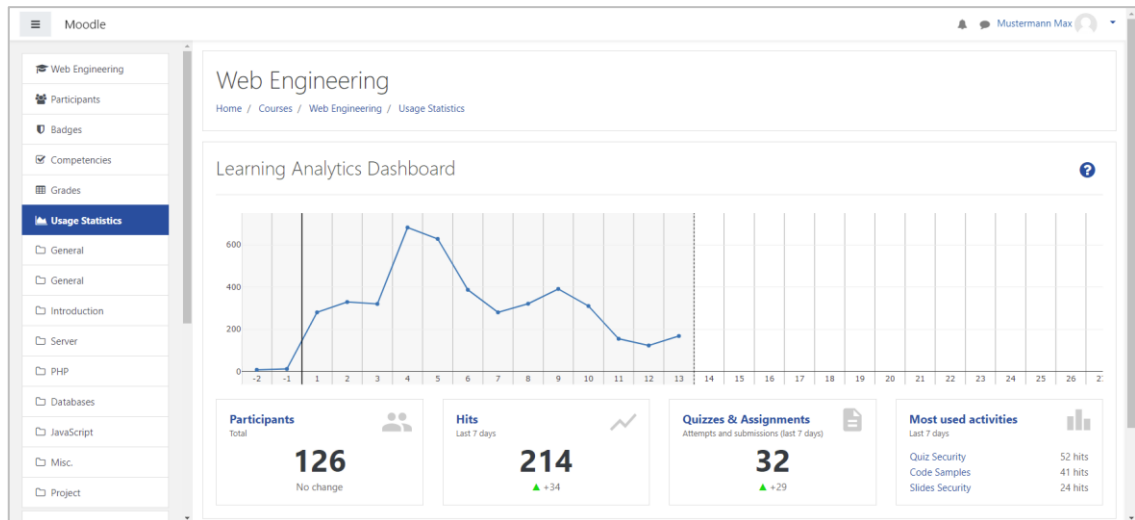
During operation of the software, I was in close contact to the Moodle operators from universities. There were some bugs during operation, which were in most cases fixed relatively soon after I was informed. One example was a wrong regular expression in the browser detection code, which led to wrong technical reports in the first semester. I fixed the regular expressions and reset the values for each device to zero in the week after being made aware of the bug. In addition, a textual information was added for one week to inform about the reset of the values. Another bug was discovered regarding the names of activities, which were cut off due to using a wrong function that was unable to handle a German umlaut (due to their UTF-8 character encoding) in strings. Another detected bug was that the heatmap that generated hourly visualizations did not include daylight saving time shifts of time zones when calculating the corresponding hour. This led to visualizations being off by one hour in some cases. Overall, no major issues were reported. There were no issues concerning the operation of Moodle itself. All bugs were related to the reports and visualizations.

6.5 Implemented Visualizations

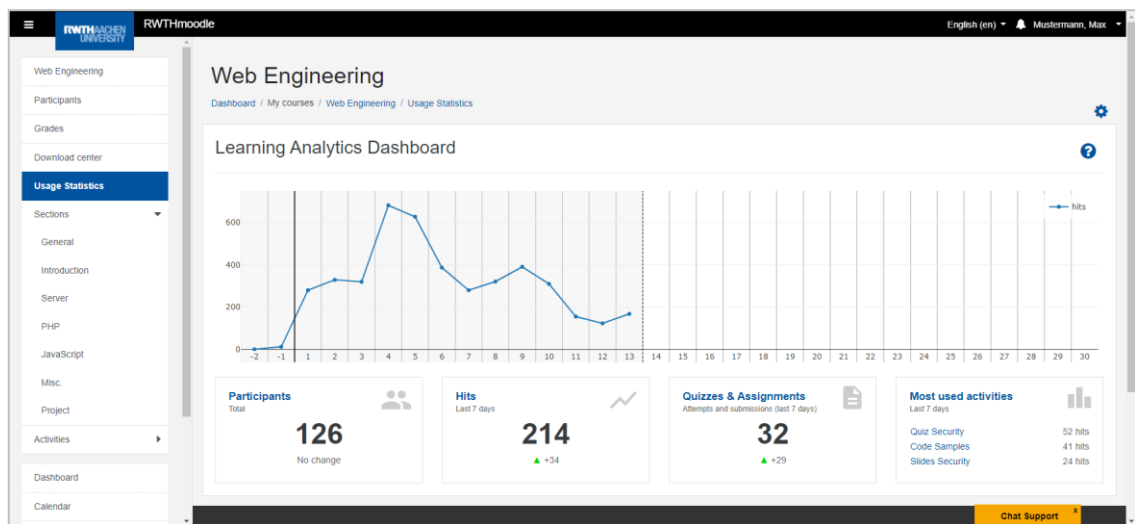
In the following, I explain which reports and visualizations were implemented in which semester. As explained in more detail in the next chapter, the evaluation was conducted using two case studies. While the overall appearance of the software remained the same, modifications regarding the supported features were implemented. In addition, multiple universities were involved in the evaluation featuring different Moodle themes. In the following, the implemented changes are explained as well as the use of different themes.

As Moodle is very customizable, participating universities used slightly different Moodle designs that also applied to the developed software. Figure 45 compares the default Moodle design with the *RWTHmoodle* design. The presented screen captures show the software in demo mode featuring the exact same content. The default Moodle design, as depicted in Figure 45a features a very light theme with much spacing between headers page content. The *RWTHmoodle* design, as shown in Figure 45b, changes the colors of headers and footers to a darker style and removes some of the spacing.

Overall, the design differences are negligible and are not considered further in the evaluation of the software. Following screen captures will use the default Moodle design, but most of the figures will only show the central part of the software, without navigation bar, heading and footer, as this is always the same.



(a) Default Moodle installation



(b) RWTHmoodle

Figure 45: Comparison of Moodle designs

6.5.1 Case Study 1: Summer Semester 2020

For the first case study, in the summer semester 2020, four pages were implemented visualizing a variety of indicators. Figure 46 shows the overall embedding of the learning analytics software in Moodle. To create the following pictures the demo mode of the software was used (as explained in Section 6.4.2), which mimics the use of the software in a real scenario, but does not utilize real data. Examples of real data are shown in the evaluation in Chapter 7.

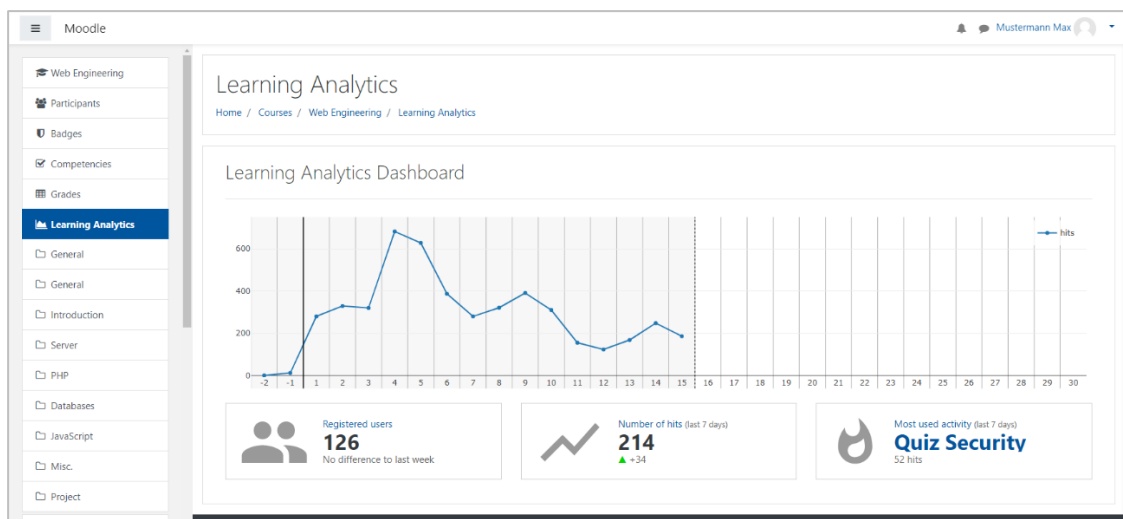


Figure 46: Learning analytics dashboard page in a Moodle course

The learning analytics page for each course was available to students as well as lecturers. A link was embedded into the Moodle navigation on the left side of the website. The surrounding layout was similar to other Moodle pages with the title of the page at the top, followed by a so-called *breadcrumb navigation* that lists the navigation hierarchy. The link *Learning Analytics* is therefore a child link of the displayed course (*Web Engineering* in Figure 46).

Figure 47 shows the view of the dashboard without navigation bar, header, and footer. At the top, the aggregated activity in the course for each week is shown. The user can hover over each row in the graph to see a tooltip of the corresponding data point with more information. The visualization starts two weeks before the start of the course (as specified in the course settings) indicated by negative week numbers. A line is shown to indicate the start of the first week. To indicate that no more data is present, a dashed line is shown at the end of the graph. The current week is excluded from the graph as otherwise there would likely be a downwards trend for the last week as it contains less underlying data points.

Below the graph at the top, three boxes are displayed, each containing a summarized preview of each of the linked pages as well as an icon to improve the user experience. The box at the left shows the total number of enrolled users. The link *Registered users* takes the user to the *Participants* page. Below the big number, the difference to the previous week is depicted. The box in the middle contains the number of hits of the last seven days. Below, the difference of this number to the previous seven days is depicted. The link *Number of hits* leads to the *Browser and Operating System* page. The last box shows the most used activity of the last seven days. The linked resource takes the user directly to that resource. Below the name of the resource, the number of hits of that resource is shown. The link *Most used activity* leads to the page *Activities*.

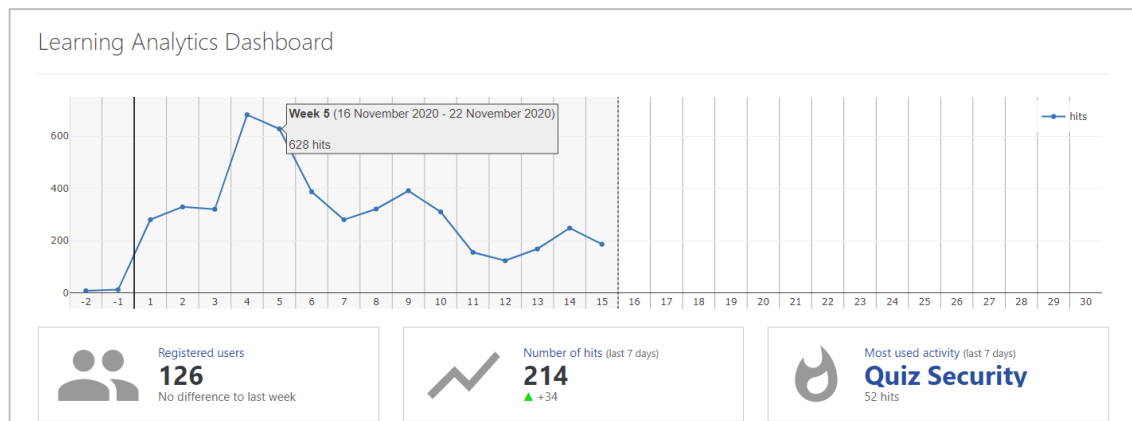


Figure 47: Dashboard (first case study)

The page *Participants* as depicted in Figure 48 contains information about the enrolled students. It shows which courses have been heard before by the students on the left and which courses students are enrolled to in the same semester on the right. In contrast to other visualizations, the underlying data is queried from Moodle database tables instead of the log data. Therefore, the shown data does not rely on the logging data and is even correct if the plugin was just installed. This is especially useful for the table labeled *Courses heard before*.

The data itself is visualized as a mix of a table and bar charts, named micro charts (see Section 6.3). The labels of the axis (course and percentage) are specified in a tabular way at the top while a bar below each value accompanies the values.

By default, only the ten courses with most enrolled students are shown. In case, there are more courses that users are or have been enrolled, a link is shown at the bottom of the table that expands the table when clicked.

When the page was developed, it was discussed if more information regarding students should be shown. Especially, the language and location would have been easy to implement as internal Moodle data could be used. Due to ethical concerns, these indicators were not added.

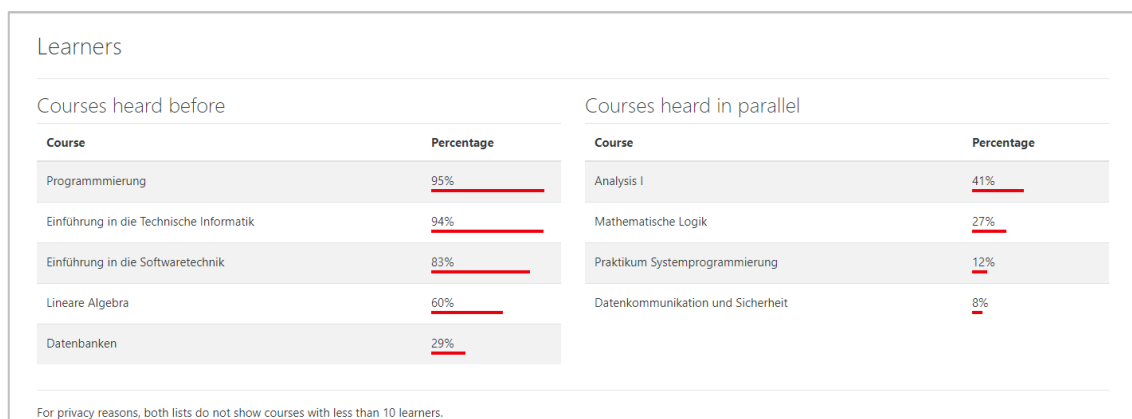


Figure 48: Page *Participants* (first case study)

Figure 49 shows a screen capture of the *Browser and Operating System* page. The page lists the used operating systems and browsers used to access Moodle. At the top three bar charts indicate

the most used platforms, operating systems, and mobile operating systems. Below, a table shows the total number of accesses for desktop browsers.

The data was generated by using a dedicated log subplugin (*lalog*). This subplugin was executed for each event and stored the data in a dedicated aggregation table. The information was only based on the user agent as all logging happened on the server. To make it impossible to identify individual users, no detailed information about browsers like the version number or operating systems were stored.

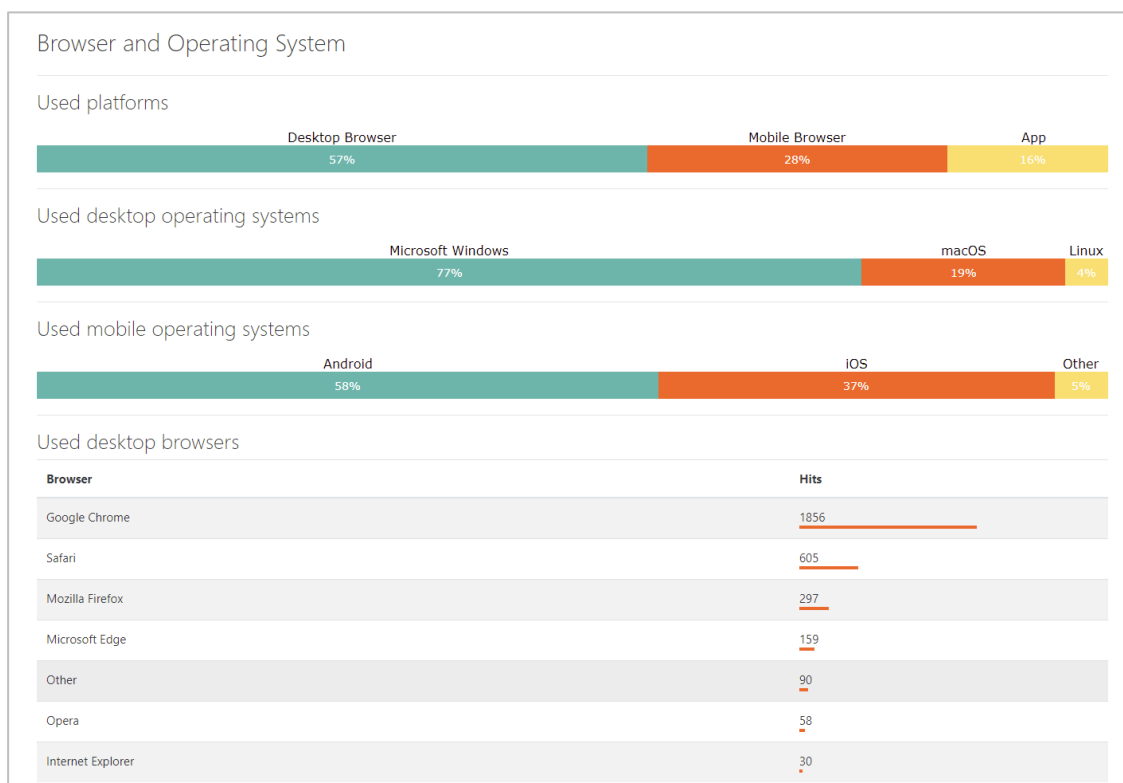


Figure 49: Page *Browser and Operating System* (first case study)

The last page (*Activities*) is shown in Figure 50. It consists of three parts and an additional information page. The top part of the page lists all course resources and activities in a horizontal bar chart. The order is based on their order in the course page. Each type of activity (e.g. quiz or file) is given a dedicated color to make it easy to identify them visually. As an example, it is easy to see that the quizzes in Figure 50 were accessed more than other resources in the course.

Below the bar chart, a table is shown listing the activity type and their aggregated number of hits. As before, micro charts are used to make it easy to compare numbers visually. The used colors correspond to the colors used in the bar chart above. In addition, the used icons are the same as used in the course page by Moodle.

The last element on the page is the table showing the most used activities. Here, the five most used activities of the course are listed. The table also shows the activity type as well as the name of the course section in which it appears. In the last column, the number of hits is listed.

At the bottom of the list, an *Expand list* link is shown that takes the user to the full list of activities. That list contains all activities displayed in tabular form as the table at the bottom of the page, but ordered by their appearance in the course.

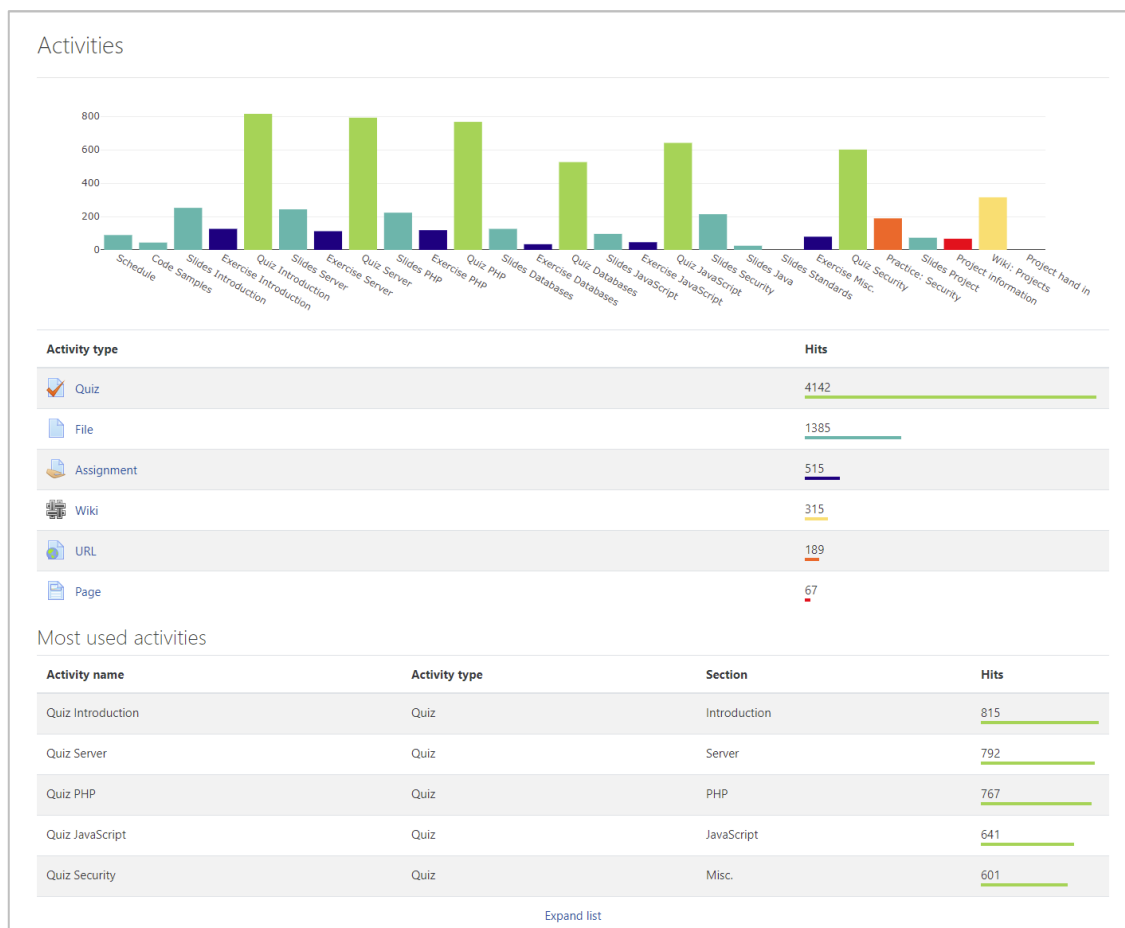


Figure 50: Page Activities (first case study)

The presented reports and visualizations were used for half a year at two universities after which an evaluation was conducted. Based on the evaluation, the software was iterated to reflect the received user feedback. The most notable changes are discussed in the next section.

6.5.2 Case Study 2: Winter Semester 2020/21

A full record of changes is available in the code repository in the changelog file⁷². In the following, the most notable changes are presented. As before, the screenshots used below use the demo mode of the software as explained in Section 6.4.2 to simplify the identification of changes.

The entry page of the user interface plugin was slightly changed compared to the previous semester. The most notable change was the introduction of a fourth box (*Quizzes & Assignments*) below the plot at the top leading to an additional report. In addition, the link *Hits* did not lead to the page containing technical information anymore but instead linked to a heatmap (explained further below).

⁷² Available online: https://github.com/rwthanalytics/moodle-local_learning_analytics/blob/master/CHANGELOG.md

A more subtle change was that the numbers of the boxes were put into focus. While in the first case study, the box icons were relatively big in comparison to the numbers, this time, the size of the icons was significantly reduced and they were moved into the top right corner of the boxes, putting the numbers into the visual center.

The latest box (*Most used activities*) was changed from showing only the most used activity to listing the three most used ones next to the number of hits in a tabular format. Thanks to this change, the names of the listed activities were also smaller making it possible to show the full name of most activities. Previously, the names of activities were often cut off to fit into the box.

Other changes included the addition of a help icon at the top right of the dashboard that linked to the help page which shows user documentation and answers some frequently asked questions. An interactive tour was introduced to explain the software to users interactively (see Section 6.3.2). In addition, the layout of the page was optimized for mobile devices making the boxes in the dashboard move below each other vertically for smaller screen widths. A screen capture of the dashboard is shown in Figure 51.



Figure 51: Page Dashboard (second case study)

The page *Participants* (Figure 52) did receive two subtle changes: First, I changed the style of the micro charts. Previously, the bars were red which was not fitting well to the overall Moodle and RWTHmoodle style. Second, I added the ability to merge courses with the same name together. This had to be configured by Moodle administrators, due to technical requirements. When activated, courses with identical names were merged and displayed as single course. Previously, Moodle courses with the same name would show up as two separate courses. This led to confusion, as sometimes two courses with the same name from different semesters would be listed. This change solved the problem.

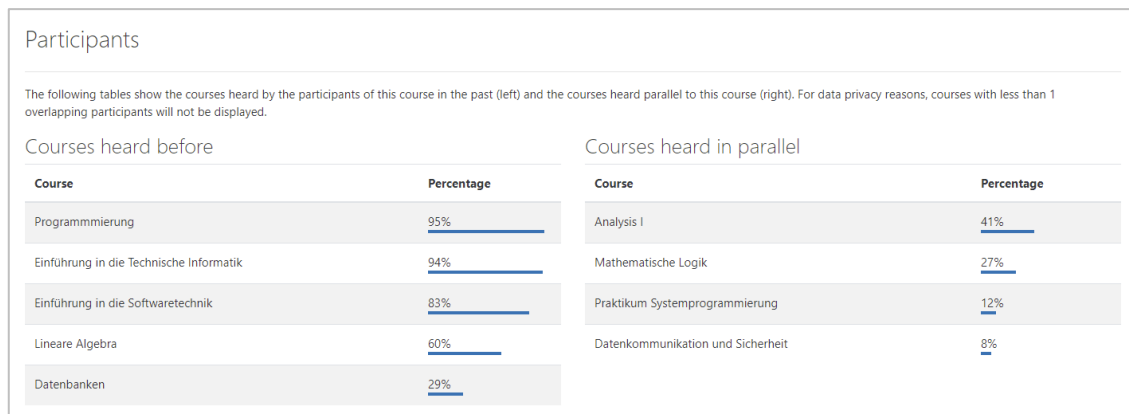


Figure 52: Page *Participants* (second case study)

The page *Browser and Operating System* was removed. As the evaluation of the features from the first case study revealed (see Chapter 7), the provided technical information was deemed as rather not useful by most lecturers. It was replaced by a heatmap showing the weekly access pattern of users. A screen capture is shown in Figure 53. The new visualization contains all events from users and lists them across weekdays (y-axis) and hours (x-axis). This makes it easy for lecturers to see at a glance at which time of the days students are most active in the system.

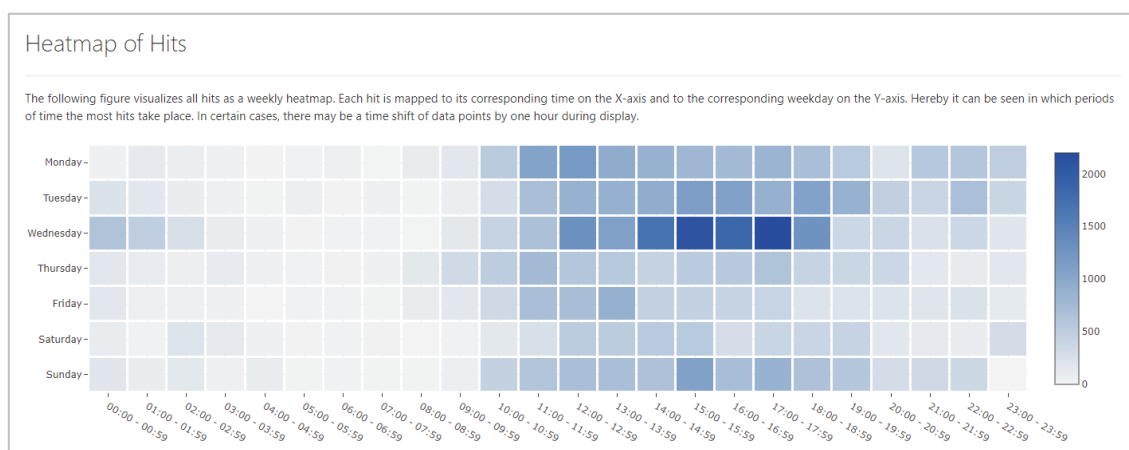


Figure 53: Page *Heatmap* (second case study)

The newly added page *Quizzes and Assignments* was developed as response for lecturers asking for personalized reports. One of the drawbacks from the approach of logging no personal information is that reports are unable to present data related to individual participants based on the logged data. Using this approach, I tried to compensate for this drawback by using internal Moodle data to create visualizations. The page makes use of the quiz results data stored in other Moodle database tables. The data is already visible to lecturers by clicking on individual quizzes or by using the course gradebook, which displays all grades and outcomes of quizzes for students. I decided to use the same data but instead show all data in a single page to improve upon the reports of Moodle. To be in line with the remaining parts of the plugin, I still apply the ethical and transparency guidelines that require a threshold before displaying data and require making the data available to students.

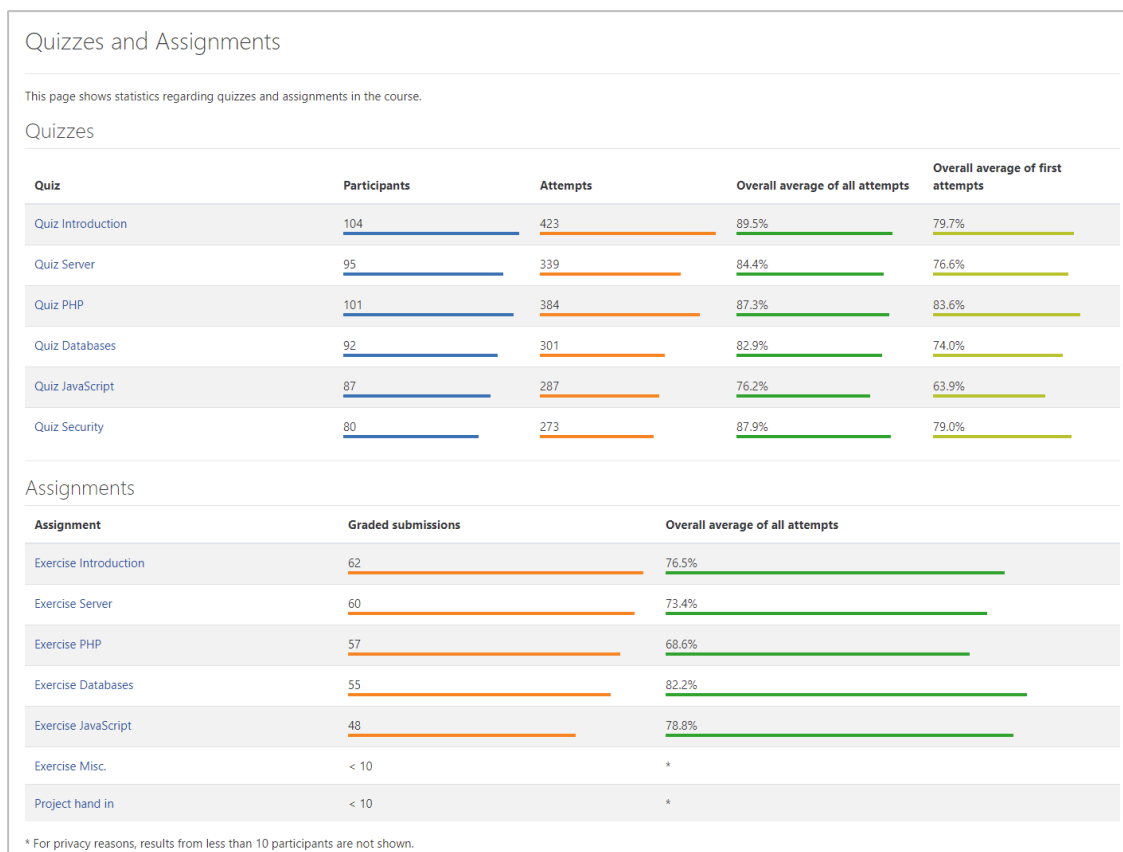


Figure 54: Page *Quizzes and Assignments* (second case study)

Figure 54 shows the resulting page. It shows the number of participants, attempts and the reached scores for each quiz of the course. Below, the number of submissions and the average scores are shown for assignments. In case a course does not contain any quizzes or assignments, the page remains empty and shows a note saying that there are no such activities in the course.

The page showing activities received some minor changes. A search bar was added at the top right which allow to filter the report below for activities containing the search term. In addition, the plot at the top of the page was changed to group activities into their course sections. This change was made based on feedback from lecturers that explained that they perceived the bar chart as chaotic. They were unaware that the order of activities in the plot was based on the order of activities in the course. Therefore, the visual grouping into sections was introduced. Figure 55 shows the result.

In addition, multiple minor changes, mostly related to the user experience, were introduced before the start of the second case study. One change was the addition of loading indicators to visualizations as in some cases it might take up to a second before a chart was shown. Another change was the renaming of the navigation item *Learning Analytics* to *Usage Statistics*, as the term was easier to understand for users.

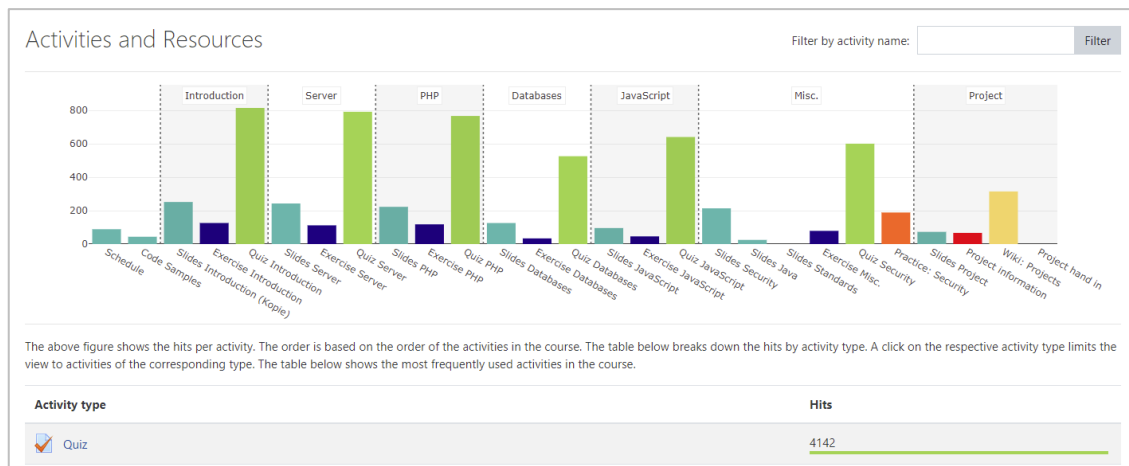


Figure 55: Page Activities (second case study)

Several changes in the backend of the software were implemented which I do not cover in detail. One major change regarding the user experience was that lecturers were able to activate the software themselves via the course settings (see Figure 44 in Section 6.4.1). This allowed them to activate the user interface of the software without the need of contacting the ServiceDesk and led to an increase of users after the feature was announced to lecturers.

6.6 Summary

In this chapter, I presented the implementation of my learning analytics software. It is split into two Moodle plugins. One serves as data logging tool, one serves as user interface for lecturers and students. The software was implemented with the requirement to respect data privacy to the point that no logged data can be traced back to individual users. Therefore, the logging software stores the data anonymously in a database table. The downside of this approach is that the user interface is not able to show personalized reports based on the log data. The software implementation currently focuses on lecturers as individual reports for users are impossible with this approach. An exception to this are reports like *Quizzes & Assignments*, which rely on internal Moodle data for visualization.

Overall, several thousand lines of code were written as part of this dissertation, mostly by myself but also by undergraduates, apprentices, and other developers, even from external universities. In the period from March 2020 (first public release) to June 2021, over 28 versions of the local plugin have been published and 8 versions of the logstore plugin.

To give an impression of the scale of the application, the following tables lists the lines of code of the local and the logstore plugin. More lines have been written during the prototype development as well as during the development of the software. This is the summary of the current state of the software.

Table 14 lists the number of files, lines of code, lines of comments, blank lines, and total number of lines for each file type of the local plugin. It is to note that the large number of comments is caused by the license requirements of the GPL, which requires multiples lines of comments at the start of each file. In addition, the Moodle coding style requires further comments in each file related to the content of the file. Therefore, most PHP files contain about 20 lines of comments at the start.

Table 14: File statistics of the Moodle plugin *local_learning_analytics*

Language	Files	Code Lines	Comment Lines	Blank Lines	Total Lines
PHP	64	3,245	1,556	782	5,583
Documentation (Markdown)	5	795	0	214	1,009
CSS	4	280	2	21	303
JSON	3	106	0	0	106
HTML	3	60	0	5	65
JavaScript	3	19	9	5	33
XML	1	10	0	0	10
Total	83	4,515	1,567	1,027	7,109

Table 15 lists the statistics related to the logstore plugin.

Table 15: File statistics of the Moodle plugin *logstore_lanalytics*

Language	Files	Code Lines	Comment Lines	Blank Lines	Total Lines
PHP	10	688	172	156	1,016
Documentation (Markdown)	2	42	0	16	58
XML	1	37	0	0	37
Total	13	767	172	172	1,111

The code has been published under an open-source license on GitHub⁷³ where anyone can view the code and the used algorithms. The software has been used at multiple German universities as well as universities from outside of Europe at the point of writing. Details related to its usage and adoption will be evaluated in the next chapter.

⁷³ Repositories on Github: https://github.com/rwthanalytics/moodle-local_learning_analytics (local plugin) and https://github.com/rwthanalytics/moodle-logstore_lanalytics (logstore plugin)

7 Evaluation

This chapter presents the evaluation of the developed learning analytics software based on the research questions of this dissertation. The evaluation is split into two parts: A technical and a user evaluation, of which the latter is again split into two case studies.

The first section outlines the used methodology for the evaluation and explains how the research questions were answered. Afterwards, the collected data at RWTH Aachen University is presented to put the evaluation into context. In total, over 115 million log records were collected over the course of two semesters. Additionally, the results are compared to data from another German university, *Hochschule Ruhr West*, and a university from outside of Europe, the *German University of Technology in Oman*.

After presenting the data, the technical evaluation is described in the third section. Results from the scalability, extensibility, and transferability evaluation are presented and discussed. In the following two sections, the user evaluation is presented. It was conducted using two longitudinal case studies, which ran over the course of two consecutive semesters. Both times a quantitative approach was chosen to evaluate the implemented software based on questionnaires sent to participants. In the first case study, conducted in the summer semester 2020, lecturers from two universities used the software in the field. It focused on the overall usage of the software and its usability. After the first case study, results were gathered and based on the feedback the software was iterated before the start of the next semester. The learning analytics software was centrally announced by the vice-rector for teaching during the second case study leading to a large increase of users at RWTH Aachen University. In addition, a third university started using the software in the winter semester 2020/21 leading to three German universities who used the software in production. Due to the large number of users, this part of the evaluation also addressed scalability aspects.

The last section concludes the chapter by summarizing the evaluation and discussing results.

7.1 Methodology

The research questions can be split into two categories: Questions related to how a learning analytics software can be practically implemented and foundational questions.

The research questions related to the technical implementations are how a learning analytics software can be integrated successfully into an LMS. Based on the collected data, which is presented, the developed software is evaluated regarding scalability, extensibility, and transferability. To evaluate the scalability of the software, the needed data storage of the log

database table was analyzed. Afterwards, the runtime performance for individual courses was simulated on a development machine. The runtime behavior of the slowest SQL queries for the biggest courses was simulated to estimate how well the software performs in real scenarios.

Afterwards, the evaluation covers the practical evaluation of the created software, which also answers foundational questions related to whether the approach using only anonymous data satisfied the need of lecturers for learning analytics. Before the COVID-19 pandemic, the original plan was to evaluate the software by conducting qualitative interviews with lecturers and experts. Using thinking-aloud tests, the goal was to identify issues within the software and discuss possible changes and the incorporation into day-to-day teaching activities. Due to the pandemic, RWTH Aachen University operated the software from the start for the whole university on the central Moodle platform and logged data for all courses. Therefore, more users than originally anticipated joined the case studies and the evaluation was switched to questionnaires, which were sent to all participants at the end of the semester. In addition, talks to individual lecturers were used to gather qualitative feedback.

The questionnaire was in large parts based on related research regarding learning analytics development as well as usability and user experience evaluations. The questionnaire was divided into multiple parts, which are summarized by the following questions:

- In which courses was the software used? (size and kind of course)
- How was the usability of the software rated? (based on the *System Usability Scale*)
- At what point in the semester was the software used?
- How do lecturers assess each feature?
- Did the software support the lecturers in their teaching?
- What kind of features would lecturers like to have in the future?

The second case study used the same base structure but due to the increased number of users, questions regarding the participants (like their profession) were added. In addition, they were asked whether they had participated in the first case study to test for significant differences in their responses.

7.2 Data Analysis

Millions of records were collected over the course of two semesters. The data contains all events inside Moodle that were triggered by students. Events from lecturers (in Moodle so called course managers) were not logged. The data reflects how Moodle was used. In the following, I present the analysis of the data to give a perspective on the actual use of Moodle and to put the later parts of the evaluation into context. In total, about 115 million log entries were generated over the course of two semesters at RWTH Aachen. Due to the large amount of data, my analysis can also serve as a general indicator regarding when and how Moodle is being used. Due to the COVID-19 pandemic, most learning activities happened online. The presented numbers can therefore also be interpreted as a general indicator regarding the learning activity of students.

Thanks to the cooperation with *Hochschule Ruhr West* as well as the *German University of Technology in Oman*, I was also able to analyze data from other universities and compare the data. In the following, the analysis from the data of RWTH Aachen is presented before showing the comparison to other universities.

7.2.1 RWTH Aachen University

Figure 56 shows the average weekly trend of LMS activity. For each weekday, a bar shows the average number of events. As an example, on Tuesdays there are on average about 400,000 events triggered in Moodle. The visualization clearly indicates that students use the LMS the most on Mondays with a decreasing trend over the week, hitting a low on Saturday.

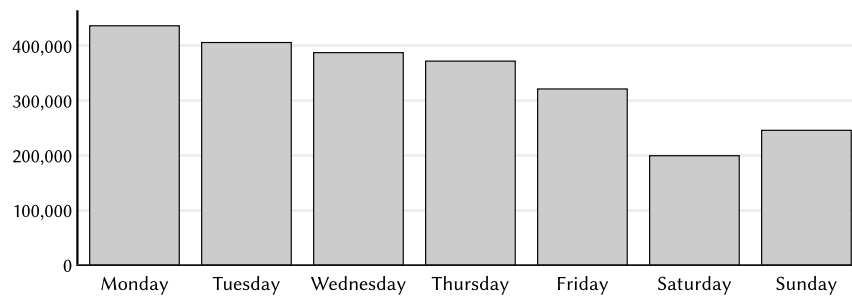


Figure 56: Number of events in Moodle by weekday

Figure 57 shows the daily student activity in Moodle. The numbers show the average amount of views for the corresponding period. For example, there are on average about 25,000 events in the hour from 16:00 to 17:00 (labeled 16). There is a visible peak between 10:00 and 17:00 of about 25,000 events per hour. After 17:00, this number steadily decreases hitting a low around 5:00 in the morning. Still, there are roughly 1,200 views between 5:00 and 6:00.

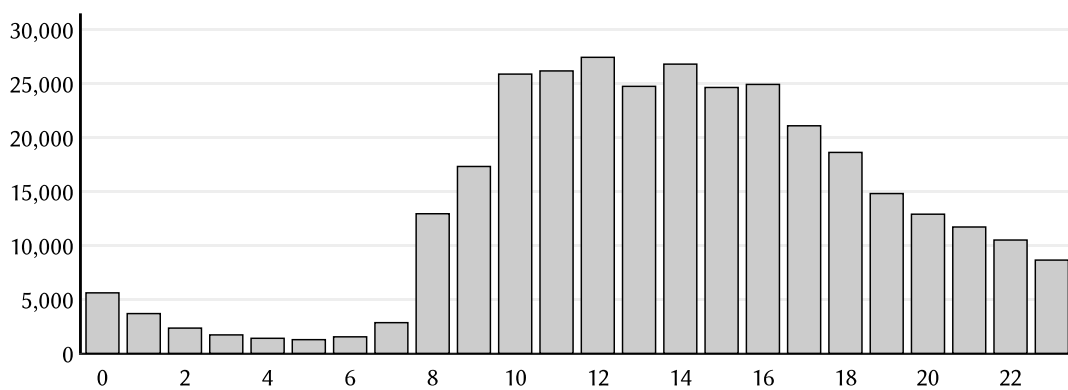


Figure 57: Number of events in Moodle by hour

As a different form of visualization, I used a heatmap to show the weekly average distribution. Figure 58 shows the values relative their absolute peak of the week, which is Mondays between 12:00 and 13:00 with 5,100 views on average. Darker squares indicate a high value in relation to the peak, light squares a lower value. As in previous visualizations, the x-axis indicates the interval of the corresponding hour (e.g. the field labeled 8 refers to the period from 8:00 to 9:00). On the y-axis, the weekdays are labeled.

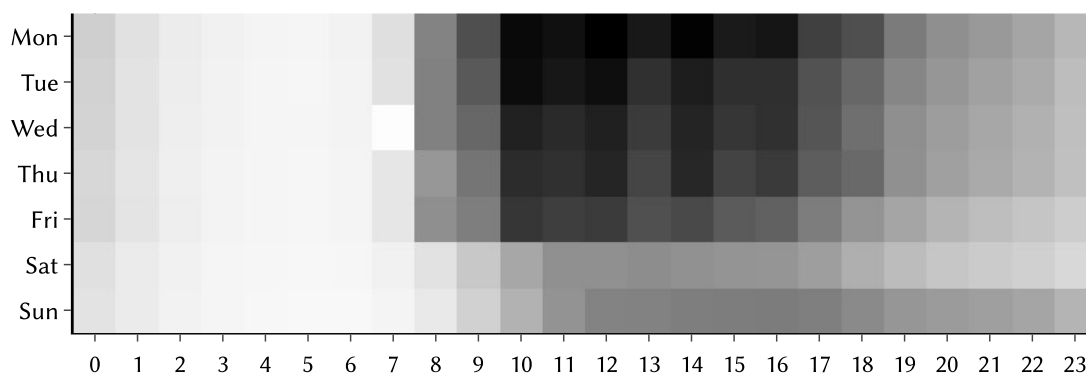


Figure 58: Heatmap of Moodle activity of students

As already indicated in Figure 56 and 57, most events are triggered on workdays in the time from 10:00 to 17:00. The heatmap also reveals local maxima for each day in the periods starting at 10:00, 12:00, 14:00, and 16:00. These local peaks result from the lecture slots at RWTH Aachen University starting at 8:30, 10:30, 12:30, and so on. Therefore, students access the LMS more often in the earlier parts of the lecture or shortly before the start of lectures. In the next analysis (Figure 59), this correlation is investigated in more detail.

One outlier in the heatmap is visible on Wednesdays between 7:00 and 8:00. During this time, the system operators regularly maintained or updated the Moodle platform resulting in close to zero events, as nobody was able to use the system at that point in time.

As already indicated by Figure 56, Moodle was used less on weekends. Another fact regarding the weekly use is that Sundays the overall usage of the system remains nearly constant throughout the afternoon. This might be due to homework or other deadlines that must be submitted by the end of the week.

One hypothesis was that the student activity correlates with the lecture slots. To test this, I obtained a list from the administration of the university containing all registered teaching activities, which includes lectures, tutoring sessions, seminars, and other activities. For each activity, I received the start and end time as well as the number of registered students. While the actual number of students who participated is likely lower than the number of registered students, this gives a good estimate of the size of each course. The list contained 69,200 activities from the summer semester 2020 and 82,500 records from the winter semester 2020/21. The following tests were conducted for each semester but no significant difference between both semesters was found. Therefore, the data was merged resulting in 151,700 records in total, which the following report is built on.

Based on the list of activities, I divided the records into slot from Monday to Sunday and 0:00 to 23:59. Over 91% of all teaching activities started in the slots from Mondays to Fridays and in the hourly slots from 8:00 to 9:00, 10:00 to 11:00, and so on. This was expected because of the fixed lecture slots. These time slots are visualized in the following figure. The remaining 9% of other teaching activities were excluded from the visualization as these were divided over the week and close to not being visible in graphical comparisons.

Figure 59a shows the results. Each field reflects the aggregated number of registered students for the designated slot. Darker fields represent higher values. The peak of registered users is Wednesdays at 10:30 with around 16,000 registered students, the low on Fridays at 18:30 with

6,400 students. As a reminder: This number should be interpreted with caution as the number of registered students for a lecture does not necessarily reflect the number of present students. Students might even register for multiple courses given in parallel.

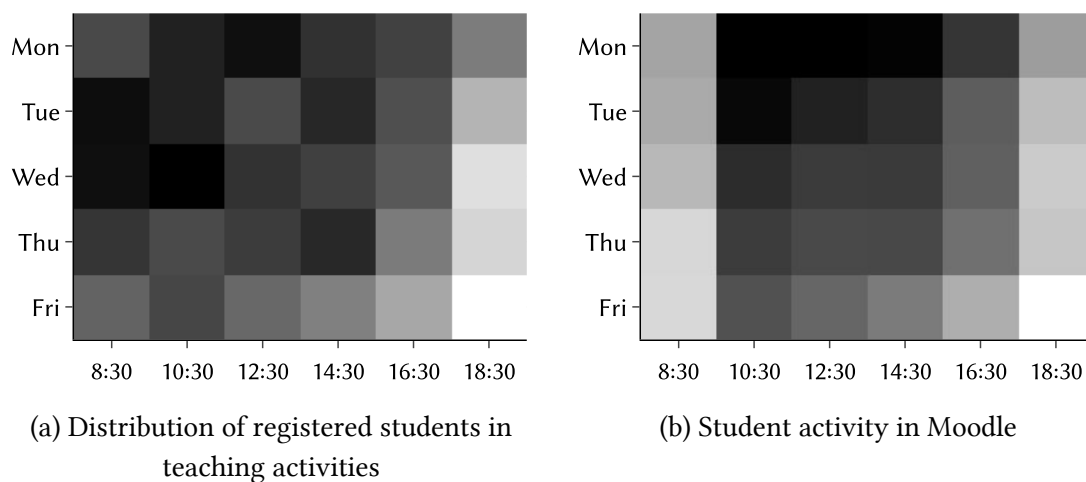


Figure 59: Weekly distribution of teaching activities by the number of registered students and the weekly access pattern of students

The resulting heatmap in Figure 59a shows two things. First, the slot at 18:30 is the least used one, the slot at 10:30 the most used one. For obvious reasons, lecturers and teaching assistants tend to prefer slots that are not in the evening. Second, only on Fridays a trend towards earlier lecture slots exists. The peak on Fridays at 10:30 is followed by a constant downwards trends.

For easier comparison, I reused the numbers visualized in Figure 58 and limited them to the time of the teaching activity slots. The result is shown in Figure 59b. As an example, the slot at 8:30 contains the aggregated number of accesses from thirty minutes before the start of the teaching activity (8:00) to 10:00. There is no direct correlation between both data sets. While many of the teaching activities start at 8:30, the majority of students does not use the LMS as actively during that time. The only correlation that can be spotted is on Fridays: Similar to the lecture slots, the activity peaks at 10:30 before constantly decreasing for the remaining day. As there is otherwise no correlation between both data sets, it is unlikely that this result can be attributed to the allocation of teaching activities. Overall, the allocation of teaching activities appears to be mostly random with exceptions on evenings and Fridays, while the activity of students follows a clear weekly trend as discussed above.

The next analysis in Figure 60 shows the number of events for each week for the full data set of one year starting in April 2020. The x-axis labels the start date of the week. The y-axis reports the total number of events from that week. Unfortunately, the first two weeks of April are missing as the university-wide logging was only activated two weeks after the start of the semester (after the rectorate made the decision to log data for all courses). The data contains the summer semester 2020 and all records for the winter semester 2020/21. In Figure 60, the lecture period as well as the lecture-free/examination period are marked. In the summer semester a downwards trends can be observed. In the first recorded week, close to 3 million events were triggered. At the end of the lecture period, this value goes down to about 2 million. The value further decreases to a low of about 600,000 events at the end of the lecture-free period.

A local minimum can be spotted in the seventh week, which can be attributed to a German holiday (Pentecost Monday on June 1, 2020).

At RWTH Aachen University, the start of the winter semester was split: For students in the second semester and later, the courses started on October 26. For first-year students, the semester began at November 2. As most German study programs regularly start in the winter semester, most students start their studies in the winter semester. This trend is confirmed by the data. The absolute weekly peak in my data is reached in the week in which the first-year students start with 5 million events in one week. The daily peak was reached on the Monday of that week with 930,000 events in a single day. Similar to the summer semester, there are local minima due to Christmas holidays at the end of the year.

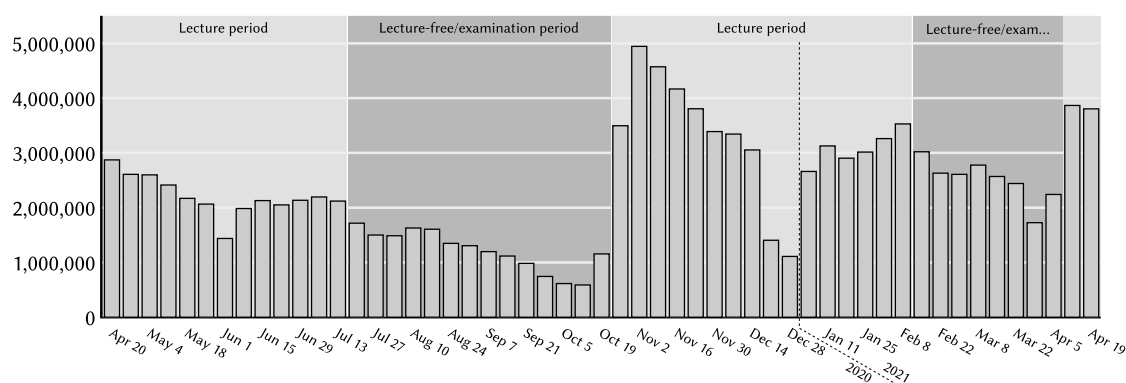


Figure 60: Weekly aggregated number of events in Moodle over two semesters

In addition, to the time component, I analyzed what kind of events were fired most often in Moodle. Table 16 lists the four most common events from the log database table, based on 115 million events in total. By far the most common event is the *course_viewed* event, which is triggered by accessing the course page. The second most common event is the viewing of a resource, which is the download or view of a file (like a PDF). The third most common event is triggered by students looking at results of a finished quiz. The fourth most common event lists how often a page was accessed. This event might be triggered more often in comparison to previous (non-pandemic) semesters as many lecturers used pages to include videos. All remaining events account to less than 5% of the total events.

Table 16: Most common events in Moodle

Event name	Component	Frequency	
		in millions	relative
<code>\core\event\course_viewed</code>	Core	42.3	37%
<code>\mod_resource\event\course_module_viewed</code>	Resource	14.9	13%
<code>\mod_quiz\event\attempt_viewed</code>	Quiz	8.5	7%
<code>\mod_page\event\course_module_viewed</code>	Page	6.5	6%

In addition to the name of the event, the corresponding components that triggered the events were analyzed. Similarly, the results above, about 43% of events are triggered from Moodle Core, 13% from resources, 13% from quizzes, 9% from assignments, 6% from pages, 6% from forums, and 3% from folders. All other components relate to less than 1% of all events.

The statistics from RWTH Aachen University have shown which data the logging plugin recorded and how Moodle was used by students. The data includes data from the summer semester 2020 and the winter semester 2020/21. In addition, I was able to obtain data from two other universities presented in the following.

7.2.2 Comparison with other Universities

Based on the data from RWTH Aachen University, the question remains whether the analyzed access patterns of students are unique to the university or whether students' access patterns at other universities are similar. Thanks to projects and cooperations with two other universities, I was able to compare the data from RWTH Aachen to another German university and to a university from outside of Europe.

Hochschule Ruhr West

Hochschule Ruhr West (HRW) started using the software in the summer semester 2020 and provided me the anonymous log data for further analysis. Therefore, I was able to analyze what kind of student behavior was similar to students from RWTH Aachen. It is to note, that both universities have a strong focus on Engineering and technical fields of study.

First, I analyzed the weekly and hourly access patterns. Figure 61 shows the relative number of events for each hour normalized to their absolute peak. I normalized the values as the RWTH peak is at about 27,000 events while the HRW peak is at 5,000. As in previous figures, the x-axis represents a one-hour period, so that the label 2 references the period from 2:00 to 3:00.

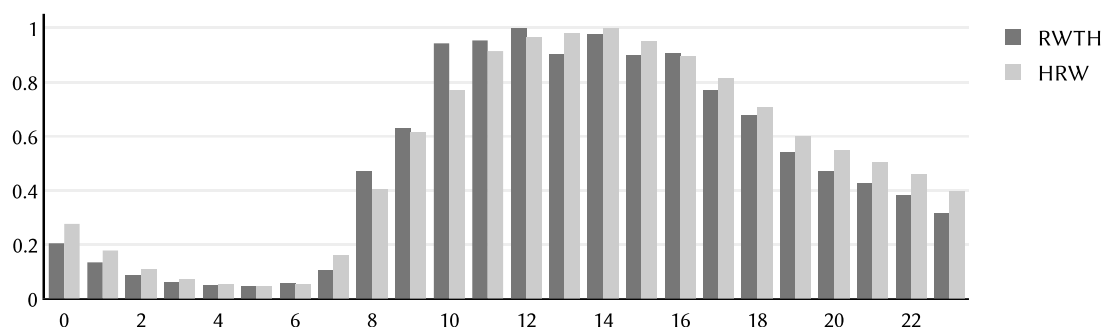


Figure 61: Comparison of RWTH Aachen University and Hochschule Ruhr West regarding events per hour relative to their absolute daily peak

The visualization shows that the overall access patterns of students is similar. There are slight differences regarding the peaks. The LMS activity at RWTH increases earlier at the start of the day, with the peak being between 12:00 and 13:00. The activity at HRW reaches the peak between 14:00 and 15:00, but stays higher for the remaining day. As a reminder: The values are scaled to their absolute peak allowing no direct comparison of absolute values.

The weekly distribution of LMS activity for HRW is nearly identical to RWTH, with the average difference being less than 4% (normalized to their absolute peak). The created visualization is

therefore almost identical to the one from RWTH (see Figure 56). Similarly, the heatmap contains no big differences, except for the difference of fixed lecture slots at RWTH (at 10:30, 12:30, etc.) leading to multiple local maxima, which was not the case at HRW.

German University of Technology in Oman

The *German University of Technology in Oman* (GUtech) is a private university in Oman, which was established in collaboration with RWTH Aachen University in 2007. Due to a collaborating project, the learning analytics software has been in use at GUtech since September 2020. To gain further insights, the university shared their log data with me. This log data was used to compare access patterns from RWTH Aachen to GUtech.

In summary, most of the discovered access patterns at RWTH can also be discovered at GUtech. Figure 62 show the comparison of hourly accesses adapted to their respective time zone and normalized to their absolute peak. Similar to RWTH, the students at GUtech tend to use the LMS the most in the periods from 10:00 to 17:00. However, while overall peaks and trends are similar, the difference between maxima and minima is much smaller. At RWTH, the students access the system between 5:00 and 6:00 95% less compared to the peak at 12:00. In comparison to that, the students at GUtech only use the system 45% less between 6:00 and 7:00 compared to the absolute peak at 15:00. I can conclude from this, that the LMS activity at GUtech is much more spread over the day than at RWTH.

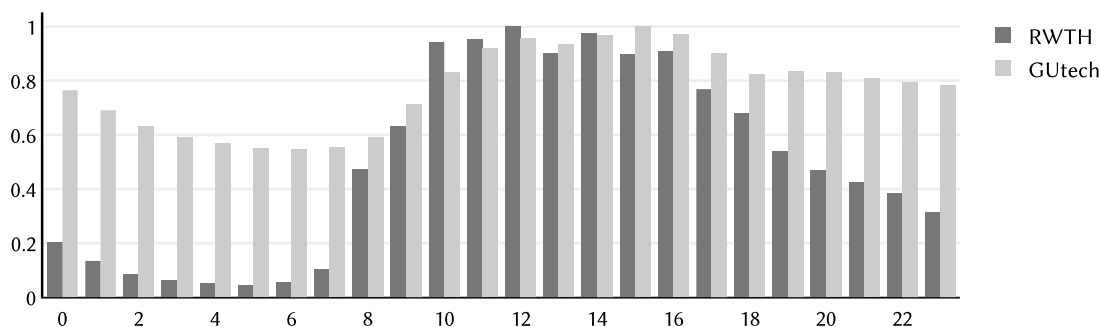


Figure 62: Comparison of RWTH Aachen University and GUtech regarding events per hour relative to their absolute peak

The analysis was also conducted for weekdays. Figure 63 shows the result. While students tend to use the LMS the most at the start of the workweek⁷⁴, the activity declines until the minimum is reached on the first day of the weekend. However, similar to the previous analysis, the difference between maxima and minima is noticeable. While at RWTH, the minimum reaches 46% of the absolute peak, at GUtech the minimum is at 76% of the first day of the workweek.

⁷⁴ The weekend in Oman, as in most Arab countries, is on Friday and Saturday and the start of the workweek is on Sunday. In my analysis of data of the GUtech, I therefore compared the Monday in Germany to the Sunday in Oman. The names of the days on the x-axis in Figure 63 are therefore not individually labeled but instead clustered into workweek and weekend.

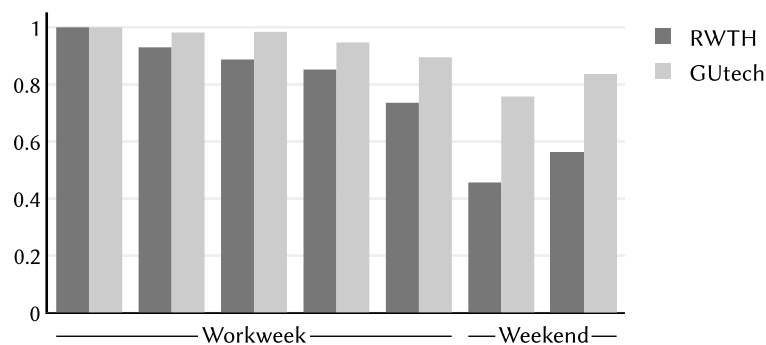


Figure 63: Comparison of activity of students by weekday between RWTH Aachen University and GÜtech relative to their absolute peak

A conclusion that can be drawn from the analysis is that the LMS activity in the GÜtech is much more distributed over the day and week compared to Germany. Data from both German universities shows that compared to the GÜtech, the weekend is used a lot less for digital learning activities. I talked to a professor of GÜtech and to people who are familiar with the country to find out where these differences come from. The main reason lies in cultural differences. Many people tend to use the cool of the night for activities while resting in the heat of the day. Similarly, the weekend is often not seen as a strict resting day. While in Germany shops are closed on Sunday, many Omanis use the weekend for shopping or other activities.

Overall, the presented data from RWTH Aachen as well as its comparison to other universities shows trends in students' learning activities and use of Moodle. In the following, the collected data from RWTH Aachen is analyzed to conduct the technical evaluation of the implementation.

7.3 Technical Evaluation

In the following, the developed learning analytics software is evaluated regarding scalability, extensibility, and transferability. The evaluation is based on the collected data, as presented in the previous section and feedback from the IT operators of Moodle.

7.3.1 Scalability

The scalability was evaluated regarding two parts of the software: The logging plugin, which writes data of events into the database and the user interface, which runs the SQL queries to generate the reports. In the following, the logging plugin is evaluated regarding its storage space and load on the database. Afterwards, the runtime behavior and impact of the user interface were evaluated.

Logging and Data Storage

To analyze how much data was stored over the course of both semesters, I analyzed the SQL data from the logging plugin at RWTH Aachen. For this, I only looked at the logging database table (*logstore_analytics_log*) that contains the logging records. I ignored the helper database table (*logstore_analytics_evtname*) as the space was only a few kilobytes. Compared to the logging data, the table is therefore negligible regarding the storage and space requirements. In the summer semester, I also used another helper table to store browser and operating system data, but as this part of the plugin was later removed, I ignore it in the evaluation.

In the summer semester 2020, over 44 million log entries were generated. In the following winter semester, over 72 million records were collected. Table 17 shows the number of rows and the required database space of the rows, split into space needed for data and indices. I did not generate the numbers on the production system but on a separate MySQL server using the *InnoDB* storage engine. A single row needs about 60 bytes of storage for index and data space. That space is used to about 60% for index storage, allowing a fast access of the data for individual courses and to about 40% for the actual data.

Table 17: Database data storage

Semester	Rows in millions	Data Space in GiB	Index Space in GiB	Total Space in GiB
Summer 2020	44.0	1.1	1.5	2.6
Winter 2020/21	71.5	1.8	2.5	4.3
Total	115.5	2.9	4.0	6.9

As RWTH Aachen switched to online-based lectures, one can assume that the number of traffic was higher than in previous semesters. Therefore, it can be assumed that the number of logged entries might go down for future semesters after the pandemic. Overall, 6.9 GiB for all logging rows of two semesters is a low value compared to alternative solutions using dedicated learning record stores.

In addition to the amount of data that was stored, I evaluated the impact of the write requests to the database in my local development environment. The MySQL database profiler *Neor Profile SQL*⁷⁵ was used which acts as a proxy server for the database. After activation, all SQL statements were executed against the profiler, which reported the executed SQL queries and their runtime. To evaluate the impact of the logging service, the number of database queries as well as the query runtime for several Moodle course pages was recorded. To my surprise, the number of queries that Moodle executes varies greatly for different courses. For some courses, Moodle executed up to 200 database queries for a single course page. Other courses required only 40 queries. The tests were executed with Moodle version 3.7. The database queries of Moodle include queries for user information, course information, session locks, and other course related information. Through operational optimization of the system, this number can likely be reduced, for example by using Moodle's caching system. My local testing environment did not do that. Therefore, my results would likely differ on production systems. Nevertheless, the following numbers can serve as an estimate of the produced load on a server.

The logging software executes two database queries when a Moodle event is triggered. The first query selects the *ID* of the event from the helper table containing the event names. The second query then executes the actual *INSERT* statement in the logging table. I ran the queries by loading the course pages of five different courses with a varying number of activities. The first

⁷⁵ <https://www.profilesql.com/>

query (*SELECT*) had a mean runtime of 2.4 milliseconds. The second query (*INSERT*) had a mean runtime of 4.5 milliseconds. Depending on the size of the courses, this runtime accounted for about 2-6% of the total runtime of database queries executed when the course page was loaded. That means the database queries executed by my logging plugin might increase the load on the database by up to 6%.

Regarding the actual operation of Moodle at RWTH Aachen, I was in close contact to the system operators of the IT Center. They raised an issue regarding the logging system, as previously the subplugin aggregating the technical statistics would fire *UPDATE* statements that would raise warnings on their logging server. This issue was fixed later. In addition, the system operators have raised concerns that long-term the logstore database table might become too big either requiring the deletion of data or moving the data to an external database server. The system operators had great interest in the collected data and its analysis, as it allows drawing conclusions regarding the scalability of the Moodle platform as a whole. At the end of the semesters, I supported the operators by generating tailored statistics for the operation of Moodle.

Overall, the logging software stored few gigabytes of data per semester and caused no noticeable increase in load on the database itself. The logging part of the learning analytics software is therefore well suitable for high-load scenarios.

Database Queries of the User Interface

To evaluate the user interface and its database queries, I looked at the courses that produced most log entries and the tables with most enrolled users. For comparison, courses with high number of users but fewer log records were added as well as courses with only few enrolled students. For each course, I evaluated the runtime of three database queries that are executed by the user interface. The *Dashboard* database query in Listing 11 is executed to create the plot at the top in the dashboard. It aggregates the number of hits by week. The query in Listing 12 generates the data for the *Heatmap* page by returning the number of Moodle events by hour and weekday. Both queries are the most computational heavy database queries used by the application. They have similar runtime behavior as both are primarily using the time of the logged event. The last evaluated query was an adapted⁷⁶ version of the database query run to generate the page *Activities* (Listing 13). In addition to the query, the Moodle API is used to get the name of each activity. This part was excluded from the evaluation as this is not part of my development and depends on various internal Moodle factors like caching of modules.

For the evaluation of the database queries, I ran each query five times and reported the mean. The results for the evaluated courses are listed in Table 18. All database queries were run with SQL statements disabling any database caches⁷⁷ that might speed up the query in real scenarios but also make it impossible to estimate the actual runtime reliably. The full SQL queries are

⁷⁶ An adapted version was used as the original includes database JOIN statements to Moodle tables. As I ran the queries in my local database environment, I had no access to the production data of the Moodle system and therefore was only able to run that part of the query.

⁷⁷ The *SQL_NO_CACHE* statement was used inside the *SELECT* statement, which disables database caches that a server might use in a real scenario. Therefore, the executed runtimes reflect the values in case the statement would be executed for the first time. Subsequent statements might even be faster in reality.

listed in Listing 11 to 13. The queries were executed in my local testing environment and not on the actual database system. For comparison: The IT Center of RWTH Aachen uses an InnoDB database cluster based on five dedicated database nodes with 14 CPU cores and 64GB memory each. In my local testing environment, I used a machine with 8 CPU cores and 16 GB of memory. This machine is also running Windows with a graphical user interface as well as the full development environment. Therefore, my test must be interpreted as a rough estimate rather than an exact measure of query time. It is to assume that most dedicated database systems would perform at least similarly well if not better than my local test.

```
SELECT SQL_NO_CACHE
  (FLOOR(l.timecreated / (7 * 60 * 60 * 24)) + 1) AS week,
  COUNT(*) hits
FROM mdl_logstore_lanalytics_log l
  WHERE l.courseid = ?
GROUP BY week
ORDER BY week
```

Listing 11: Database query to evaluate the performance of the page *Dashboard*

```
SELECT SQL_NO_CACHE
  FLOOR((l.timecreated % (60*60*24*7)) / (60*60)) AS heatpoint,
  COUNT(1) AS value
FROM mdl_logstore_lanalytics_log AS l
  WHERE l.courseid = ?
GROUP BY heatpoint
ORDER BY heatpoint
```

Listing 12: Database query to evaluate the performance of the page *Heatmap*

```
SELECT SQL_NO_CACHE
  COUNT(l.id) hits,
  l.contextid
FROM mdl_logstore_lanalytics_log l
  WHERE l.courseid = ?
GROUP BY l.contextid
ORDER BY hits DESC
```

Listing 13: Database query to evaluate the performance of the page *Activities*

Table 18 shows the results. Each row contains one course with the number of rows in the log database table and the enrolled users. I included courses from both case studies. The number of table rows was rounded to thousands, the number of enrolled students to hundreds. On the right, I show the results of the database queries. The table is sorted by the number of logging entries (first column).

Table 18: Runtimes of database queries for selected courses

Log records of course	Enrolled students	Dashboard query runtime in milliseconds	Heatmap query runtime in milliseconds	Activities query runtime in milliseconds
1,232,000	1,300	828	838	272
984,000	2,700	665	653	216
757,000	1,000	482	494	166
556,000	2,500	369	366	125
468,000	2,200	303	341	103
241,000	1,500	153	175	59
139,000	500	91	94	31
34,000	100	25	25	15

There is a linear correlation between the number of table rows and the corresponding query times of the investigated queries making it possible to estimate the load on the database even before the use of the database in production. From the results, one can see that on my testing machine, all database queries were executed in less than one second. Even the queries for the course with more than one million table rows took less than a second to execute. The software has therefore proven to be scalable even in scenarios with thousands of students.

7.3.2 Extensibility

As explained previously, the resulting software of this dissertation is released as open source under the GNU General Public License. Other universities can therefore build upon the code and even contribute back code to the repository. This has already happened in the past, with another university contributing the code needed to support another SQL database management system.

The software comes with two subplugin types (see Chapter 6) that allow extending the logging functionality and the user interface. While the logging functionality supports the use case of aggregating data from the log, the user interface can be used to show any kind of data. At the end of the winter semester 2020/21, the IT Center of RWTH Aachen discussed the possibility to integrate insights regarding the use of their video hosting platform with me. As the user evaluation later will show, this was of great interest and one of the most requested features (see feedback from lecturers of the first case study in Section 7.4.2). I had a short video conference in which I presented the plugin system and the subplugin type *lareport*. Afterwards, staff of the IT Center used my learning analytics software and the subplugin type to provide video insights to lecturers. It is currently being used in the running summer semester 2021. Their plugin uses an external logging server (Matomo) to log accesses to the provided videos. The implemented subplugin then fetches the data from the server and presents it as part of the dashboard. In addition, further information can be displayed on a separate page. Figure 64 shows the integration.

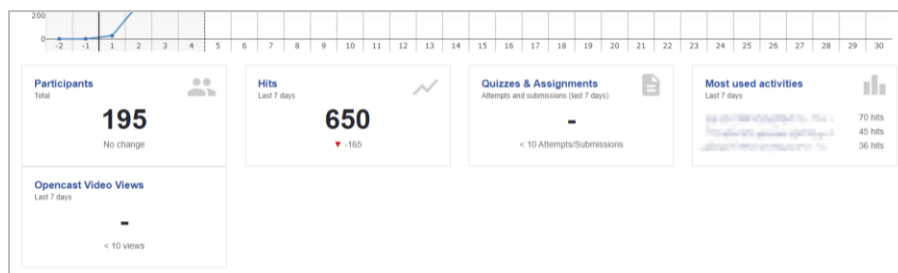


Figure 64: Integration of a custom subplugin for video views in the bottom left

It is to note that the integration of the custom plugin developed by the IT Center happened before the introduction of the *preview.php*, as explained in Chapter 6. The *preview.php* file was introduced based in parts on their feedback. Previously, the integration was lacking some features. In addition, the styling of the added box was not optimal as the necessary space between the boxes in the next row is missing. This was fixed in a newer version of the software. Nevertheless, regarding the functionality itself, the integration worked.

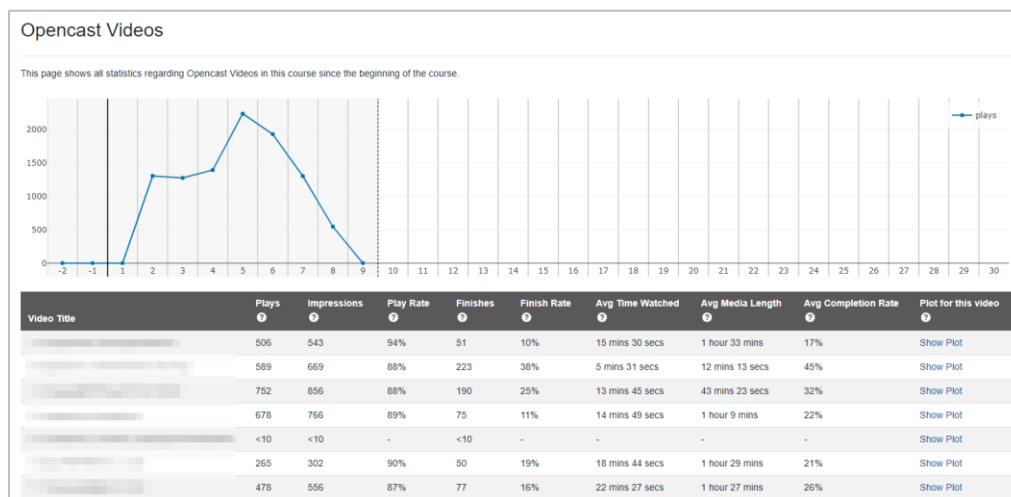


Figure 65: Plugin page for videos developed by the IT Center

In addition to the box on the dashboard, the IT Center developed a page with more detailed visualizations for each video of the course. A screen capture of an actual course (hence pixilated) is shown in Figure 65. The IT Center did the development on their own (apart from the initial talk).

After the IT Center developed and deployed their plugin, I interviewed the developer to evaluate the use of the subplugins. I asked for possible improvements and for an estimate how easy the integration was. The integration into the dashboard was problematic (due to the missing *preview.php* at that time as explained above). That issue was fixed. The overall integration was described as “very easy” by the developer. The availability of many helper functions and components like the plot component was praised. This shows that the extensibility of the plugin by the use of subplugins is not just a theoretical construct but already used in practice at RWTH Aachen.

7.3.3 Transferability

To check how well the created software could potentially be used by other universities, I analyzed how many other German universities use Moodle.

Table 19: Use of learning management systems at German universities

Learning Management System	Universities		Students	
	absolute	relative	absolute	relative
Moodle	162	38%	1,470,000	53%
ILIAS	70	16%	590,000	21%
OpenOLAT	17	4%	190,000	7%
OPAL	9	2%	90,000	3%
Stud.IP	8	2%	60,000	2%
Other	11	3%	120,000	4%
None/Unknown	153	36%	220,000	8%

In total, over 426 German universities were explored in a semi-automated way. As basis, the Wikipedia list of German universities⁷⁸ was used. A crawler was developed to automatically follow the links and check the homepages of each university for texts like *learning management system* and LMS names like *Moodle* or *Blackboard*. In more than 200 cases, this approach allowed an automated detection of the used LMS. After that, the list was manually verified and it was checked if more information could be found out by manually searching for the LMS on the page of each university. In addition, I reported the number of students for each university based on the value in the Wikipedia list. While these values are, in some cases up to ten years old, this still serves as good estimate.

Of 426 investigated university websites, the LMSs of 277 universities were identified. Table 19 shows the results. For each detected LMS, it shows the number of universities that use it. I found four universities that used multiple LMS, which is why the total number of universities is higher than the number of investigated universities. The table also reports the aggregated number of students for the universities. This shows that only small universities do not use an LMS or use an LMS unable to detect: While 36% of German universities use no or an unknown LMS, this accounts for only 8% of students. This is not surprising as 173 universities on the list have less than thousand students with some smaller universities not even having a proper online presence with information about their LMS (if they even have one).

Of the 277 LMS that were identified, 97% were either *Moodle*, *ILIAS*, *OpenOLAT*, *OPAL* or *Stud.IP*. All other learning management systems were used by less than three universities. Two LMS

⁷⁸ https://de.wikipedia.org/wiki/Liste_der_Hochschulen_in_Deutschland

that are prominent outside of Europe are *Canvas* (used twice) and *Blackboard* (used once). Other systems were often custom built, sometimes based on systems like *Microsoft SharePoint* (like *Emma+* from *Hochschule für nachhaltige Entwicklung Eberswalde*).

The results show that Moodle is, by far, the most used LMS in Germany. This means that the practical results of this dissertation are transferrable to a majority of universities in Germany. Although, it was not possible to check which Moodle version was used by each universities, the software has been tested even with older version of Moodle (back to version 3.2 released in 2016). Therefore, even universities using older Moodle version are able to use the software. In theory, that means that over 38% of all German universities could potentially install the developed learning analytics software.

In practice, there are already multiple universities using the software. The following four universities in Germany are using the software:

- *RWTH Aachen University* since the summer semester 2020
- *Hochschule Ruhr West* since the summer semester 2020
- *Hochschule für angewandte Wissenschaften Neu-Ulm* since the winter semester 2020/21
- *Paderborn University* since the summer semester 2021

In addition, there are two universities from outside of Germany, which are using the software:

- *German University of Technology in Oman* since September 2020
- *National University of Cuyo* (Argentina) since November 2020

The *GUtech* is using the software due a project (*Step2Future*) funded by the *German Academic Exchange Service* (German: *Deutscher Akademischer Austauschdienst, DAAD*). The project enabled me contact to lecturers and researchers from *GUtech* for cooperation to foster the digitalization of teaching at *GUtech*. The Argentine university was informed about the software due to a cooperation with *Hochschule für angewandte Wissenschaften Neu-Ulm*, who was already using the software.

Before and during the operation of the software, I had direct contact to the Moodle operators of most universities. In semi-structured interviews, I asked for feedback regarding installation and operation. Most operators praised the easy installation as well as the nice integration into Moodle. In addition, the configuration via Moodle settings was mentioned as positive. Overall, the feedback from IT operators was mainly positive.

Three more universities have reached out to me asking for more information or signaled that they are currently evaluating the use of the software in coming semesters. As the project is available online, without the need to contact me before using, there might even be more institutions using the software already.

After the technical evaluation, the following sections describe two longitudinal case studies, which were conducted to gather feedback from actual users.

7.4 Case Study 1: Summer Semester 2020

The first case study was conducted in the summer semester 2020. The goal of the case study was to evaluate the usage and usefulness of the implemented software and to gather feedback for an iteration of the software. My hypothesis was that the learning analytics service was easy

to use by professors and teaching assistants and that they would incorporate it into their teaching activities. In total, 63 lecturers with 169 courses used the software at two universities.

7.4.1 Setting

Two universities participated in the case study, *RWTH Aachen University* and *Hochschule Ruhr West*. At both universities, the software was deployed to log the data of all students, not just the students of the courses who participated in the case study. This allowed lecturers to join the case study even after the start of the semester. While the majority of lecturers joined the case study during the beginning of the semester, a small number of users joined during the running term. Originally, the first case study was supposed to include a small number of lecturers from RWTH Aachen to gather qualitative feedback based on interview sessions. Due to the COVID-19 pandemic, the plan was changed shortly before the start of the summer semester 2020 as more and more lecturers asked to join.

Original Plan and Changes due to COVID-19

Before the start of the summer semester in April 2020, a meeting with the central IT department of RWTH Aachen, the IT Center, was held to discuss the start of the summer semester and the test of the learning analytics software. The original plan was to install both Moodle plugins in the central Moodle system of RWTH Aachen but set it up in a way that only data of the courses of the participating lecturers was logged. The meeting was only a few days before it became apparent that the COVID-19 pandemic was inevitable with major implications for teaching and learning scenarios. One week after the meeting (still in the lecture-free period), the administration of RWTH Aachen announced all employees should work from home whenever possible. In addition, the university announced that the summer semester would be held fully digitally without face-to-face teaching with only few exceptions.

Due to this shift to digital teaching, the IT Center and Center for Learning Services (CLS) planned to make my learning analytics software available to all courses. Therefore, the setting was changed. The software was configured to log the data of all courses (instead of only the participating ones). This was done to offer analytics to lecturers as my software was the only active logging system in Moodle.

The decision was confirmed by the rectorate of RWTH Aachen and the setting was changed in mid-April 2020. By default, the learning analytics software was not visible to lecturers and students. They first had to approach the IT Center, CLS, or me so that their courses were added to the list of participating courses via the administrative settings. I was in close contact to the IT Center and CLS and was informed about new lecturers joining the case study. An instruction sheet⁷⁹ was created by me that showed how the software could be used and configured in courses. It was sent to all participating lecturers via mail.

At RWTH Aachen, no central announcement was made during the first case study regarding the software. As the user evaluation will later show, most lecturers actively asked for analytics

⁷⁹ The instruction sheet consisted of seven PDF pages. Most of them showed the screenshots, as presented in Section 6.5.1, with information on how each feature could be used and accessed. In addition, the sheet contained information regarding data privacy and contact information for questions.

or were made aware of the functionality by their colleagues. In addition, colleagues from the CLS mentioned the software in Moodle training sessions.

Although, the implemented plugins were not published yet or announced at conferences, staff of another university, *Hochschule Ruhr West – University of Applied Sciences (HRW)* reached out to me. Because of a previous talk regarding the second prototype at a German conference (*MoodleMoot* in March 2018), they were aware that learning analytics development for Moodle was happening at RWTH Aachen. They asked if there was a software to be used. Similar to RWTH Aachen, their lecturers also switched to digital lectures due to the pandemic, increasing the need for basic descriptive analytics. I provided them the source code, helped regarding installation, and provided documentation. In turn, they supported me in the evaluation by sending the evaluation questionnaire to their lecturers.

Participants

During the summer semester 2020, the software was used at two German universities, *RWTH Aachen University (RWTH)* and *Hochschule Ruhr West – University of Applied Sciences (HRW)*. The semester started in April 2020 and ended in September. Table 20 lists the number of participating lecturers and courses. In addition, the number of enrollments in the participating course as well as the number of total students is given. The latter is provided as both universities operated the software in a setting that logged the data of all students. The user interface and the link to it in the course was unlocked only for the participating courses as explained above. At first, the IT Center decided to make the software only available in a few courses fearing that the software might not scale well enough. During the semester, it turned out that this was not the case and therefore the limit was lifted. Across both universities, 63 lecturers with 169 courses participated.

Table 20: Participants of the first case study

University	Participating lecturers	Participating courses	Enrollments in courses ⁸⁰	Total students at university
RWTH	31	59	13,106	45,628
HRW	32	110	10,516	23,622
Total	63	169	23,622	69,250

To evaluate the software and its usage, a survey based on a digital questionnaire was conducted. It was sent all participating lecturers at the end of the semester via mail.

⁸⁰ The total number of enrollments for all participating courses was counted, not the unique number of students. For that reason, this number can be higher than the total of students.

Survey

The questionnaire was split into five pages, each covering one of the following aspects:

- The participating lecturer (university, faculty, etc.)
- The participating course(s)
- Usability
- Usage and feature usefulness
- Reflection, impact, and future development

The first two pages asked the participant metadata regarding the participating lecturer and course(s). As the same questionnaire was used at RWTH and HRW, the first two questions asked for the associated university of the lecturer as well as the faculty association. The last question on the first page asked how many courses of the lecturer participated in the case study. Following, the second page asked how the lecturer became aware of the learning analytics service and how many students were roughly enrolled in the courses.

To assess the usability of the software, the lecturers were asked to complete the 10-item System Usability Scale (SUS) by Brooke (Brooke, 1996). The System Usability Scale (SUS) was chosen as usability test, as it is well-researched and widely used for assessing the usability of software applications (Bangor et al., 2009; Kortum & Bangor, 2013; Lewis, 2018) as explained in Chapter 3. In addition, the simple 10-question format worked well together with the remaining part of the questionnaire.

The next page asked how often the software was roughly used and at which point during the semester. Unfortunately, I was not able to use the log data to confirm these findings, as the logging system was setup in a way to log only events triggered by students. In addition, lecturers were asked to grade the usefulness of each page.

The last page asked whether the software supported the lecturer in his teaching activities and whether the lecturer was overall satisfied with the use of learning analytics. In addition, two questions were asked to identify the need for personalized learning analytics. The first question asked if the lecturer would like to see this kind of insights. The second question asked whether the lecturers would like to compare students directly to each other. The questions were based on a Likert scale (1-5). The last question asked for open feedback.

Since my learning analytics software focuses on lecturers, it was unclear whether soliciting feedback from students would benefit the evaluation. As the only way to received feedback from students at RWTH would have been to ask lecturers to send a questionnaire to their students, it was decided not to do it. Even if I had convinced lecturers to submit the questionnaire, it likely would have resulted in only a few responses as the semester was already over. However, HRW had a central questionnaire for students where they were able to ask questions regarding learning analytics. Therefore, at HRW questions regarding the software were integrated into the university-wide questionnaire for students.

The questionnaire consisted of seven questions based on a 5-scale Likert scale. The first two questions asked when the tool was used. The next three questions asked whether they liked having access to the learning analytics page, whether it supported their learning, and whether the statistics were easy to understand. Two more questions asked if they were interested in personalized visualizations and if they would like to see comparisons to fellow students. A last question asked for open feedback.

7.4.2 Survey Results and Discussion

The questionnaire for lecturers was sent to all participants via mail at the end of the summer semester, in the last week of September. It is to note that the questionnaire used was German; the items state in the following are translated into English. Of the 63 lecturers who used the learning analytics software in the summer semester 2020, 43 completed the questionnaire (68%). All participants were of the group of professors and teaching assistants.

Participating Lecturers and Courses

Table 21 shows to which faculties the lecturers belonged. Due to the technical focus of both universities, most participants were from technical and engineering related study fields. It is to note that RWTH and HRW have different names for their faculties as well as a different number of faculties. I therefore merged them into four reasonable fields of study: Engineering, natural sciences (including mathematics and computer science), business and economics, and others.

Table 21: Faculties of participating lecturers

Faculty	RWTH	HRW	Total
Engineering	8	8	16
Mathematics, Computer Science, and Natural Sciences	7	8	15
Business and Economics	1	7	8
Other	4	0	4
Sum	20	23	43

Regarding the number of courses, the majority of lecturers only asked for learning analytics to be activated in a single course: 22 of 42 lecturers participated with one course. Most of the remaining lecturers participated with two courses (12) while 4 lecturers participated with three and four courses each. A single lecturer participated with five courses.

To find out whether learning analytics was mainly used in small or big courses, I asked for a rough estimate on how many students were enrolled in their courses. The question allowed multiple answers in case they had specified that they participated with more than one course. Table 22 shows the results. There is no clear trend. Overall, the learning analytics software was used in 2 courses with more than 500 students as well as in 4 small courses with less than 20 students enrolled. In general, the answers seem to reflect the distribution of students at both universities. RWTH, which has more students in general, has a higher average than HRW. For both universities, the number of enrolled students follows a normal distribution.

Table 22: Course sizes of participants

Number of enrolled students	RWTH	HRW	Total
< 20	2	2	4
20 - 50	4	5	9
51 - 100	5	10	15
101 - 200	7	7	14
201 - 500	7	3	10
> 500	2	0	2

As lecturers had to contact the service teams of each university to participate, they were asked how they became aware of the learning analytics service. At HRW, the software was centrally announced via mail to all lecturers. This was reflected in the answers. At RWTH, about half of the lecturers stated, that they actively asked for such a service, either by contacting the CLS or the ServiceDesk of the IT Center. 4 lecturers answered that other lecturers informed them about the learning analytics service and 2 stated that they were informed about learning analytics as part of a Moodle training. All responses are shown in Table 23. The high number of lecturers actively asking for learning analytics can be explained by the COVID-19 pandemic. This is reflected again in the feedback of the free text answers. In total, seven lecturers reported that they were very thankful for having some kind of statistics. Some of them even stated that they felt otherwise clueless on how active the students were following their lectures.

Table 23: Activation of learning analytics service

How did you become aware of the Learning Analytics service?	RWTH	HRW	Total
Through Moodle training	2	0	2
Another lecturer told me about it	4	2	6
I actively asked for such an offer	8	2	10
I was sent an email telling me about the pilot phase	1	10	11
Other	2	0	2

Features and Usability

To evaluate the usability of the software the SUS questionnaire, as presented in the related work in Section 3.4.2, was used. The scores were obtained according to the scoring of the system usability scale (*Strongly disagree* to *Strongly agree*). As every second item has a negative connotation, the ratings for these items have been reversed in the following table to make it easier to compare. Therefore, a rating of 5 reflects the best possible response. The scores in

Table 24 show the comprehensive assessment on the system usability and lists for each item option how many times which value was selected. The results can be used for comparison among different versions of the system and give a good first estimate of the overall usability of the software.

Table 24: System Usability Scale results

ID	Item	1	2	3	4	5	Mean	SD
1	I think that I would like to use this system frequently.	0	2	5	12	23	4.3	0.9
2	I found the system unnecessarily complex.*	0	2	2	18	21	4.3	0.8
3	I thought the system was easy to use.	1	1	7	16	18	4.1	0.9
4	I think that I would need the support of a technical person to be able to use this system.*	0	1	3	10	29	4.6	0.7
5	I found the various functions in this system were well integrated.	0	4	17	12	9	3.6	0.9
6	I thought there was too much inconsistency in this system.*	0	6	10	14	12	3.8	1.0
7	I would imagine that most people would learn to use this system very quickly.	0	1	2	21	19	4.3	0.7
8	I found the system very cumbersome to use.*	0	2	6	12	23	4.3	0.9
9	I felt very confident using the system.	1	1	7	19	13	4.0	0.9
10	I needed to learn a lot of things before I could get going with this system.*	0	0	3	10	30	4.6	0.6
Sum / Average of mean and SD		2	20	62	144	197	4.2	0.8

* Ratings reversed

The total score, ranging from 0 to 100 points, was obtained according to the scoring of SUS by subtracting 1 from the average points given in Table 24 and then multiplying the aggregated score with 2.5. The resulting score was 80.2 showing that respondents were satisfied with the system usability. Using the acceptability ranges described by Bangor and Kortum (Bangor et al., 2009), the software ranks in the best possible range (*Acceptable*), the equivalent of a *B* on a standard school grading scale (starting at 80). Compared to past learning analytics evaluations in literature that used the SUS, the software ranks amongst the best (compare Section 3.4.3).

The questions regarding when and how often the software was used received mixed responses. The answers to the (single-choice) question how often the software was used indicated that

about a third of the lecturers used the software once a week or even more often. Another third used the software approximately once per month. The last third used it a few times throughout the semester. No lecturer stated that the software was only used once or not at all, which is not surprising, as lecturers had to approach the IT Center, the CLS, or me to activate the software for their course room. Table 25 lists the responses.

Table 25: Usage frequency

How often did you use the Learning Analytics module over the semester approximately?	Frequency
Once a week or more often	12
Once a month	10
Few times throughout the semester	13
In total once	0
Not at all	0

Regarding the time of use, a multiple-choice item was used. As before, the time of use was also mixed. Table 26 shows the responses. No particular point can be concluded at which the software was used most often.

Table 26: Time of use (multiple selections possible)

At what times during the semester did you use the Learning Analytics module?	Frequency
After activation for my course	24
During the semester	32
At the end of the lecture period (before the start of the exam period)	25
At the beginning of the exam phase	13
Others	4

To rate the usefulness of each feature, lecturers were asked to rate each page on a Likert scale from 1 to 5. In addition, the option *I do not know this feature* was presented. The results are shown in Table 27. Overall, the usefulness of the dashboard and page *Activities* were rated very well, both reaching an average score of 4.5.

The page *Participants* received mixed ratings. The page showing technical information (browser and operating system) received overly negative scores. While one lecturer even stated in mail correspondence with me, that he liked this page a lot as it showed his “students are using Linux”, the overall usefulness was perceived as rather poor.

Table 27: Rating of usefulness for each feature

Feature	1	2	3	4	5	Mean	SD
Dashboard	0	0	6	7	23	4.5	0.8
Participants	3	8	7	9	6	3.2	1.2
Browser / Operating System	16	12	7	4	3	2.2	1.2
Activities	0	2	0	13	27	4.5	0.7
Total (in percentage) / Average	12%	14%	13%	22%	39%	3.6	1.0

For future development purposes, the lecturers were asked if they were interested in personalized statistics, which a great majority was. The responses are shown in Table 28.

Table 28: Responses regarding personalized analytics

Item	1	2	3	4	5	Mean	SD
I would like to see more personalized statistics (e.g. not only the number of clicks per action, but also how many users performed an action).	0	0	1	8	34	4.8	0.5
I would like to see statistics that allow students to be directly compared to each other.	6	4	9	3	20	3.6	1.5

Regarding the desire for personalized analytics, only one person gave a neutral rating while 79% strongly agreed. In addition, the lecturers were asked they would like to compare students to each other, which is more privacy invading. The response was mixed with 24% disagreeing, 21% taking a neutral stance, and 55% agreeing. These responses show that future research directives need to focus on providing personalized statistics while still respecting data privacy of individual users.

Overall Perception

To get an overall impression of users, two questions regarding the general perception of the software were asked. Table 29 shows the results. The first question asked lecturers to reflect on their learning activities to estimate if the tool supported them in their teaching. Over 60% of the lecturers agreed or strongly agreed with the statement. The second question, asking if they are overall satisfied with the tool, received a slightly better response with 70% agreeing or strongly agreeing.

Table 29: Overall rating

Item	1	2	3	4	5	Mean	SD
The tool has supported me in my teaching activities.	4	4	9	16	10	3.8	1.2
Overall, I am satisfied with the use of the Learning Analytics tool.	1	3	9	17	13	3.9	1.0

An additional free text field at the end asked for open feedback. Twenty lecturers made use of it. I manually clustered the responses into suitable categories. Eight lecturers asked for personalized statistics. Several of them stated that it would help them a lot if they were able to distinguish between the number of clicks a resource receives and the number of students that actually accessed the resource. This response again confirms the finding from the question regarding personalized statistics. Seven lecturers expressed their gratitude for providing the software, some even stating that they have been missing this feature in Moodle. Another common response was that lecturers would have liked to see more reports on video-based resources: Three lecturers asked for this feature. Other textual responses included individual feedback like the proposal of new features or the explanation on how the software was used. In addition to the free-text feedback, I received multiple mails of lecturers during the course of the semester. One lecturer complained that he did not like the fact that students were able to access the learning analytics page. I explained to him the stance on transparency and ethics. Multiple lecturers expressed their gratitude for the software while three lecturers confirmed the interest from the free-text answers regarding personalized statistics.

To check for significant differences, I performed multiple t-tests regarding the size of courses and university affiliation compared to the last two questions regarding the overall rating of the software. No significant differences were determined.

After the survey, I conducted semi-structured interviews with individual lecturers to reveal issues and possible ideas for optimizations. Based on the ideas of the cognitive walkthrough (Wharton et al., 1994), I asked the interviewees to perform a set of tasks. A lot of feedback included that the plot in the activities page was actually not helpful, mainly because the order of the activities in the plot was unclear. In addition, multiple lecturers explained that they did not find the browser and operating system report useful. This was also revealed by the quantitative evaluation of features. Lastly, lecturers explained that they would like to move the plot graph in the dashboard as the first week in the plot did not correspond to the actual first week of their course. The last issue was actually covered by the help document provided to lecturers. The start date of the plot was based on the course settings. The interviews revealed that this feature was not easy to discover for lecturers.

Student Evaluation

The survey for students conducted at HRW contained a branching question in the general student evaluation, showing a screen capture of the learning analytics tool and asking if the tool was used in the course. If the students answered the branching question positively, they were presented the following questions to evaluate the software. In general, the questionnaire was much simpler than the one used for lecturers, as this was part of the semester-wide evaluation of HRW and not a dedicated questionnaire for the tool. Therefore, I had to simplify

some questions, change the wording to be in style with the remaining questionnaire or even remove some items entirely. In total, 30 students participated in the survey.

The first question asked how often the students used the tool. Over 68% of the students reported that they used the tool less than once per week. The remaining ones used it once per week or even multiple times per week. Nobody used it more than once per day.

All remaining items used a 5-point Likert scale from *disagree* (1) to *agree* (5). Results are shown in Table 30. The first item asked if students liked the fact that they had access to the tool. I argue that it is important to show maximal transparency when presenting data. However, the results of the items do not clearly indicate this. Only 55% of the students agree or rather agree with the statement. A third of the students had a neutral stance on this.

Table 30: Results of student evaluation

Item	1	2	3	4	5	Mean	SD
I like that I had access to the course statistics.	1	2	10	9	7	3.7	1,0
Being aware of the course statistics has supported my learning.	10	4	10	3	1	2.3	1.2
The course statistics were easy for me to understand.	0	0	6	10	13	4.2	0.7
I would like to see personalized course statistics about my course usage.	3	2	4	9	11	3.8	1.3
I would like to see comparisons of my personal course statistics to those of my fellow students.	6	1	3	8	11	3.6	1.5

Only few students reported that the software support their learning (rated 2.3). This was not surprising, as the tool is primarily aimed at lecturers and only shown to students for transparency. The third item asking whether the reports were easy to understand scored 4.2, confirming the well-received usability aspects from the lecturers' evaluation.

The last two items regarding personalized analytics scored 3.8 for seeing personalized statistics in general and 3.6 for the comparison with fellow students. Therefore, the majority of students (69%) agreed or rather agreed that they would like to see personal statistics. Slightly less students still agreed with the items asking for comparisons with fellow students. This is on par with current research indicating comparable preferences (Schumacher & Ifenthaler, 2017). Still, the score of only 3.8 and 3.6 as well as the high standard deviation indicate that not all students like to see an implementation of this feature.

An additional free-text field asking for feedback and suggestions was used by two students. One student explained how he used the LMS and that he thinks that the software would not accurately present his learning style as he downloads all materials at the start of the semester once. A second student explained that he felt that the tool had a negative impact on the students as professors would notify students and tell them that they would not learn enough referencing

the learning analytics software. The student perceived the use of the software as monitoring and described the software as not useful.

7.4.3 Findings

The goal of the first case study was to evaluate the usage of the software and the usefulness of its features. In addition, I wanted to know whether lecturers would incorporate learning analytics into their daily teaching activities. The case study was conducted in the field with lecturers using the software as an additional feature of the LMS. In total, 43 of the 63 participating lecturers filled out the questionnaire, which itself already shows great interest in learning analytics.

Overall, a high interest in learning analytics observed. Although, the software was not centrally announced at RWTH Aachen, over 31 lecturers participated in the first case study. About half of the lecturers claimed that they actively asked for such an offer. Another quarter reported that they were told of it by other lecturers. I argue that this high demand for analytics can be traced back to the COVID-19 pandemic, which has led to a digitalization of teaching and learning processes. I argue that further steps concerning the digitalization of teaching will even increase the demand for simple descriptive statistics. Before the implementation of my software, lecturers had no way of telling how often their learning materials were downloaded making it hard to estimate how actively students were following their lecture.

The active use of learning analytics was also reflected in the answers to the question asking how often and when the software was used. Lecturers used the software over the whole semester, during the beginning of the exam phase as well as during the lecture period. About a third of lecturers reported that they used the software once a week or even more often showing that many lecturers incorporated learning analytics into their regular teaching activities.

Regarding the usability of the software, a score of 80 in the SUS questionnaire was reached showing a great reception compared to similar learning analytics tools (see Section 3.4.3). I argue that the dashboard approach had a big impact on this outcome. Hiding detailed statistics behind pages while showing the most important key metrics on the entry page must be considered the correct approach for learning analytics software. In addition, I argue that the use of micro charts as presented in Section 6.3 in Figure 37 further improved the overall accessibility of the software.

The individual ratings of the features show that two of the pages, the dashboard and activities, were perceived as very useful. The report showing technical information (browser and operating system) was considered not useful by most lecturers. Regarding the page about participants, I received mixed feedback. As a reminder: The page shows in which other courses students were enrolled in the past or are enrolled currently (in parallel to the course). Based on the free-text responses, I argue that this feature is subjective to the kind of course. As an example, this feature might be useful for lecturers giving the course *Mathematics 2* as it is of interest to know whether students have taken the course *Mathematics 1* before. On the other hand, a lecturer that gives a course for first semester students might know very well what other courses they are enrolled in and that they have not heard any other courses before.

The software does not log any personal data making it impossible to trace back database records to individual users. While this respects the data privacy of users as much as possible, it comes with the downside of not being able to generate personalized reports. In contrast to other learning analytics tools, the software is not able to tell how many students accessed a resource.

This downside is reflected by the survey, as 98% of lecturers agree or fully agree that they would like to see personalized statistics. Similarly, the student evaluation reveals the interest in personalized statistics for students, but also shows that a high number of students would rather not like to see this feature. One student even expressed his concerns regarding monitoring. Therefore, personalized statistics need to be implemented carefully and in agreement with all relevant stakeholders.

On a technical level, the user evaluation supports my claims from the technical evaluation regarding scalability. Two lecturers who gave courses with more than 500 students participated in the survey. None of them reported any kind of problem regarding scalability. A check for correlations between the course size and ratings revealed no significant differences between answers from lecturers with bigger courses. Therefore, I argue that my learning analytics software is a suitable tool for all kind of course sizes.

Overall, I consider the first case study a great success. Over 70% of the lecturers agreed or strongly agreed that they were satisfied with the use of the learning analytics tool. Negative responses mostly mentioned the missing personalized statistics. I argue that there is a high demand for basic descriptive analytics that my software provides.

7.5 Case Study 2: Winter Semester 2020/21

The second case study was conducted in the winter semester 2020/21, running from October 2020 to April 2021. Based on the results from the first evaluation, the software was iterated to incorporate feedback from lecturers. The browser and operating system visualization was removed. It was replaced by a heatmap visualization showing at which time the students accessed the system. Another page was added to show visualizations regarding quiz and assessment data. In addition, many small changes regarding usability were made. The changes are explained in more detail in Section 6.5.2.

The goal of the second case study was to confirm findings from the first case study, evaluate the iteration of the software, and to test whether the software scales with more users. My hypothesis was that the overall usability increased compared to the last iteration, that the software would be able to scale in big courses, and that the offered features would be perceived as more useful compared to the first case study.

7.5.1 Setting

In addition to the two universities from the first case study (RWTH and HRW), a third university, *Hochschule für angewandte Wissenschaften Neu-Ulm* (HNU), joined the second case study. Due to the ongoing pandemic, all three universities still held most of their lectures in blended learning scenarios or as video-based lectures. The latest version of the software was deployed before the beginning of the semester at three universities.

At RWTH Aachen, the software was not only available to lecturers contacting the ServiceDesk as in the previous semester. Instead, lecturers were able to activate the user interface by themselves in the course settings in Moodle. As before, lecturers were able to activate the software while the semester was running. In January, the vice-rector for teaching announced the learning analytics software to all professors and teaching assistances at RWTH generating an influx of new users.

At Hochschule Ruhr West, the system was ran in the same scenario as before but with different lecturers. The number of lecturers as wells as the number of courses was roughly the same as before. Hochschule Neu-Ulm ran the software the same way as HRW with system administrators manually activating the user interface for individual courses. In total, six lecturers participated with ten courses at HNU. One difference from HNU compared to RWTH and HRW was that the university did not disable the access to Moodle reports for lecturers. At RWTH and HRW, the standard logstore was disabled due to privacy concerns. At HNU, lecturers had access to Moodle reports as well as to my software. One lecturer therefore compared both systems in the free-text answers. Another difference was that the evaluation was conducted a few weeks before the evaluation of RWTH and HRW, as this was part of the semester evaluation that was conducted earlier into the semester. In addition, the HNU questionnaire contained a few more questions, most of them being free-text fields, asking questions related to learning analytics in general requested by staff of the HNU. The responses were used by the staff of the HNU internally and are not addressed in the following.

Table 31 shows the number of participants, courses, and enrollments for each university. It is to note that the exact number of participating users for RWTH could not be determined. This was the case as lecturers were able to activate the software themselves. The given number reflects all users who were enrolled in a course (with activated learning analytics) with the *manager* role. However, it was unclear who exactly activated the software. The mail to the questionnaire was therefore sent to all these course managers, although a branching question in the questionnaire did exclude all users who were not aware that the software was activated.

Table 31: Participants of the second case study

University	Participating lecturers	Participating courses	Total students at university
RWTH	~2,000*	~650	45,628
HRW	33	97	23,622
HNU	6	10	~3,800
Total	~2,000	~760	~73,000

* Number of managers in the participating courses

The used questionnaire was nearly identical with only few adaption related to the changed group of users and the iteration. As at RWTH Aachen, all course managers received the mail linking to the questionnaire, it was unclear if many student helpers or secretaries would receive the mail. As this would have changed the user group compared to the first survey, a question asking for the role of the participant was added.

To limit the findings to previous users, I also added a question asking whether the user was part of the case study in the previous semester. I also asked if the user activated the software himself via the course settings. One additional change was that the questionnaire was made available in English and German, opposed to the first questionnaire, which was conducted solely in German.

7.5.2 Survey Results and Discussion

The questionnaire was sent to all course managers of Moodle courses at RWTH with activated learning analytics as well as all lecturers participating at HRW. The six users at HNU were asked earlier due to different semester cycles. In total, the questionnaire was sent to roughly 2,000 Moodle managers at RWTH, 33 lecturers at HRW and 6 lecturers at HNU. I received 213 completed questionnaires.

Due to the high number of addressed Moodle managers at RWTH, many participants that were asked for feedback might not have actually used the software during the semester. They were possibly only made aware of it because of the mail. For that reason, I applied a strict filter to the submitted records to include only participants who had actually used the tool.

I therefore removed records that met the following criteria:

- Participants who selected *I do not know this feature* for four or more features (of five possible features). This excluded records from participants who likely only visited the dashboard page but did not actually use the software during the course of the semester. This excluded 27 records.
- Participants who stated in the free-text form that it was unclear to them what was being evaluated. This excluded records from three participants. Two of them actually provided scores for SUS as well as features regardless but stated in the free-text form that they were unsure what exactly they were evaluating.
- Participants who stated in the free-text form that they only used the software after they were asked to evaluate it (one record).

Applying the filter, I was left with 182 responses in total: 163 responses from RWTH, 15 from HRW, and 4 from HNU. These are covered in the following.

Participating Lecturers and Courses

Due to the change from a closed case study in the previous semester to an open study in which lecturers were free to activate the software on their own, a lot more lecturers from RWTH were participating. Similarly, to the previous semester, I clustered the faculties of the universities into four categories. Table 32 shows the results. As in the previous case study, the majority of lecturers belonged to faculties of the field of engineering and natural sciences.

Table 32: Faculties of participating lecturers at RWTH Aachen University and Hochschule Ruhr West in the second case study

Faculty	RWTH	HRW	Total
Engineering	78	7	85
Mathematic, Computer Science, and Natural Sciences	51	4	55
Business and Economics	7	4	11
Other	27	0	27
Sum	163	15	178

It is to note that the questionnaire for lecturers at HNU did not ask for faculty association, as the small number of participants would have likely revealed individual respondents.

The concern that a large proportion of respondents might come from the group of student helpers turned out to be unjustified: 23% of the participants were professors, 71% were researchers or research assistants. The remaining 6% were technical and administrative staff, student assistants, or others. The group of participants is therefore very similar to the group of the first case study in which only professors and research assistants participated.

Asking for an estimate in how many courses the software was activated, revealed that the majority of participants (51%) used the software in a single course. Over 23% used it in 2 courses and 14% of the participants in 3 courses. A single lecturer activated the software in 8 courses.

The number of students enrolled in the courses is shown in Table 33. As in the previous case study, the participants could select multiple answers for cases in which the software was operated in more than one course. In contrast to the previous case study, the software was used at RWTH in all kind of course sizes. Over 26 participants stated that the software was operated in a course with more than 500 enrolled students. It is to note that there might be an overlap, as some participants might reference the same course. Still, the results confirm the findings from the technical evaluation showing that the software was able to perform in high load scenarios with hundreds of students in a single course.

Table 33: Course sizes of participants

Number of enrolled students	RWTH	HRW	HNU	Total
< 20	23	3	1	27
20 - 50	48	7	1	56
51 - 100	38	6	1	45
101 - 200	31	9	3	43
201 - 500	39	0	0	39
> 500	26	0	0	26

Although the learning analytics software had been centrally announced, I asked how the participants became aware of the software in cases in which they stated that they activated the software themselves (at RWTH via the Moodle course settings). As expected, a majority of participants (51%) answered that they were informed about the software via mail. Over 14% answered that other lecturers made them aware of it, 7% said they actively asked for such a service. The remaining lecturers were informed in a Moodle training, already knew that the service existed from the previous semester, or answered *other*. A dedicated question asking whether they had been part of the case study from the previous semester received a positive answer in 19% of the answers.

Features & Usability

To evaluate the usability, the SUS questionnaire was used, as in the previous semester. The results are shown in Table 34. The resulting score was 79.0. To determine whether this was a significant change from the previous semester (80.2) I conducted a t-test, which showed that the slight decrease was actually non-significant ($p=0.28$). If any, I would have expected an increase, as multiple usability features like a separate help page were introduced before the start of the semester. But even though the score decreased insignificantly, it needs to be considered that the software is overall scoring very well compared to other software (Bangor et al., 2009).

Table 34: System Usability Scale results

ID	Item	1	2	3	4	5	Mean	SD
1	I think that I would like to use this system frequently.	0	7	30	54	89	4.3	0.9
2	I found the system unnecessarily complex.*	2	4	31	68	74	4.2	0.9
3	I thought the system was easy to use.	1	4	29	73	69	4.2	0.8
4	I think that I would need the support of a technical person to be able to use this system.*	1	7	11	37	124	4.5	0.8
5	I found the various functions in this system were well integrated.	1	13	64	70	30	3.6	0.9
6	I thought there was too much inconsistency in this system.*	4	13	39	59	59	3.9	1.0
7	I would imagine that most people would learn to use this system very quickly.	1	4	22	89	64	4.2	0.8
8	I found the system very cumbersome to use.*	2	7	24	63	83	4.2	0.9
9	I felt very confident using the system.	2	10	30	77	60	4.0	0.9
10	I needed to learn a lot of things before I could get going with this system.*	1	1	15	45	118	4.5	0.7
Sum / Average of sums		15	70	295	635	770	4.2	0.9

* Ratings reversed

The question asking how often lecturers used the software received similar results to the first study. Over 32% answered they used the software once per week or more often, 29% used it at least once per month, 24% a few times, and 15% answered they used it only once.

Similarly, there was no single point in time at which learning analytics was used most often. Lecturers used it during the lecture period, as well as at the end of it and at the start of the exam phase. Table 35 shows the responses.

Table 35: Time of use (multiple selection possible)

At what times during the semester did you use the usage statistics?	Frequency
After activation for my course	96
During the semester	102
At the end of the lecture period (before the start of the exam period)	106
At the beginning of the exam phase	65
Others	12

The provided features were slightly changed compared to the previous semester (see Section 6.5.2 for details). The page *Heatmap* and *Quizzes & Assignments* were introduced in this semester. The heatmap, which replaced the *Browser and Operating System* page from the previous semester received a much better score (3.7 compared to 2.2). The results and the comparison to the first case study are shown in Table 36.

Table 36: Rating of usefulness for each feature

Feature	1	2	3	4	5	Mean	SD	Change
Dashboard	4	9	22	57	78	4.2	1.0	-0.3
Participants	11	25	26	39	32	3.4	1.3	+0.2
Heatmap	11	20	25	38	54	3.7	1.3	+1.5*
Quizzes & Assignments	7	17	18	39	47	3.8	1.2	-**
Activities	0	7	14	50	91	4.4	0.8	-0.1
Total (in percentage) / Average	4%	11%	14%	30%	41%	3.9	1.1	+0.3

* Compared to the *Browser and Operating system* page, which it replaced

** Introduced in this semester

Both, the *Heatmap* and the *Quizzes & Assignments* page turned out to increase the overall reception of the software. Surprisingly, the rating of the dashboard decreased to 4.2 (from 4.5). The activities page stayed nearly constant at 4.4 (from 4.5). Although, the page *Participants* received no significant changes, it received a slightly better score (3.2 to 3.4). Overall, the average score of the features was increased from 3.6 in the previous case study to 3.9.

The attitude of participants towards the use of personal data and the presentation of personalized visualizations slightly changed in this case study. The question asking whether lecturers are interested in personalized statistics changed from 4.8 in the first case study to 4.2. Similarly, the question asking whether lecturers are interested in directly comparing students to each other received a score of 3.2 (previously 3.6).

Overall Perception

The results of the items regarding the overall reception of the software are shown in Table 37. The item whether the tool supported the lecturers in their teaching activities received a score of 3.4 (previously 3.8). The question whether the users were overall satisfied with the software stayed nearly constant at 3.8 (from 3.9). While only few people disagreed with the second statement, the question regarding the support of teaching activities divided participants (also indicated by the standard deviation).

Table 37: Overall rating

Item	1	2	3	4	5	Mean	SD	Change
The tool has supported me in my teaching activities.	14	26	51	51	37	3.4	1.2	-0.4
Overall, I am satisfied with the use of the Learning Analytics tool.	2	12	45	80	39	3.8	0.9	-0.1

As in the previous semester, I asked for open feedback at the end of the questionnaire. I clustered the answers into suitable categories. Again, the most requested feature was to include personalized statistics to not only show absolute click values but also show how many students accessed each resource. In total, ten lecturers mentioned this. Five lecturers asked for analytics for videos. Multiple participants explained how they used the tool in more detail while others expressed their gratitude for the service. Multiple features were proposed: More filter options for the activities, the option to filter for specific periods of time, and a history view for the activities and heatmap page. A proposed feature, which I consider out of the scope for this project was to implement an evaluation system for resources, which lets the students vote on uploaded material (“thumbs up or down”).

7.5.3 Findings

The goal of the second case study was to evaluate the iteration of the software, confirm the findings from the first case study, and test the scalability by approaching a larger user group. Due to the announcement by the vice-rector, the software was activated in over 650 courses at RWTH Aachen in the winter semester. In total, there are about 3,000 courses in Moodle each semester⁸¹ meaning that the software was activated in a quarter of all Moodle courses at RWTH Aachen. Compared to the previous case study, the exact group of participants was not known

⁸¹ There are more courses in Moodle than actual lectures as some lectures use multiple Moodle courses to divide students into groups. The numbers of lectures per semester is estimated to be around 1,500.

before the evaluation, as the mail with a link to the questionnaire was sent to all Moodle managers that had the software activated in their courses at RWTH. This meant two things: the group of lecturers at RWTH that was asked for feedback was much larger (about 2,000 against 60 in the previous semester). This means that many of the participants who filled out the questionnaire might have a lower interest in learning analytics compared to the participants from the previous semester. An additional factor was the missing personal contact. In the first case study, I had contact via mail to all participating lecturers and sent a mail with a short information sheet. In this case study, the information sheet was replaced by documentation of the IT Center and a built-in information page. This means that the participants had no personal contact to me, which may result in values that are more realistic since no personal factors were involved.

Over 650 courses, about a quarter of the courses at RWTH Aachen used the provided learning analytics software showing a high adoption throughout all faculties and courses. As in the last case study, I argue that the COVID-19 pandemic supported this drive to digitalization as most lectures were held digitally.

The use of learning analytics (how and when) stayed nearly constant compared to the first case study. As previously, there was no specific point in time at which learning analytics was used the most. The overall use of the software was divided into lecturers using the software weekly, monthly, or a few times over the course of the semester. I can therefore conclude that many lecturers incorporated learning analytics into their regular teaching activities. The scalability of the software was again confirmed with many of the participants using the software in courses with more than 500 students enrolled.

The usability of the software was perceived as non-significantly worse (from 80.2 to 79.0). Features that had only minor changes (like the *Activities or Participants* page) received a similar feedback as in the first evaluation. Surprisingly, the dashboard received an average score of 4.2 (from 4.5). The newly introduced features (*Heatmap* and *Quizzes and Assignments*) scored above average (3.7 and 3.8). They were considered more useful than the technical information provided in the previous semester.

Table 38: Comparison of scores of features and their data sources

Feature	SS 20	WS 20/21	Moodle Data	Log Data
Dashboard	4.5	4.2	●	●
Participants	3.2	3.4	●	○
Browser / Operating System	2.2	-	○	●
Heatmap	-	3.7	○	●
Quizzes & Assignments	-	3.8	●	○
Activities	4.5	4.4	○	●
Average	3.6	3.9		

Table 38 shows the average scores for both case studies. In the last two columns, I specify which kind of data source was used as basis for the reports and visualizations. As an example, the page *Quizzes & Assignments* is solely based on Moodle's internal database data. It does not use the log data collected by my plugin. Overall, the mix of Moodle data seems to be a good way to enhance purely based anonymized log statistics by data that is already present in Moodle. The comparison of the average rating of features from this case study compared to the previous one shows an increase from 3.6 to 3.9.

The free-text feedback shows that personalized statistics are the most-requested feedback. Future activities need to find a way to incorporate data privacy-respecting approaches when implementing these insights. Surprisingly, no lecturer complained that students had access to the learning analytics software, although this was stated at multiple occasions. Therefore, I conclude that the ethical considerations regarding transparency were justified and received well by lecturers.

The overall perception of the software remained nearly constant at 3.8 (from 3.9). I argue that this is due to the software being the only way for lecturers to estimate the activity of their students expect for directly asking them for feedback. Especially during the pandemic where lectures were held digitally, basic descriptive analytics are an important tool to support teaching.

7.6 Summary

The technical evaluation has shown that the developed learning analytics software is well scalable and extendible. Over the course of two semesters, more than 120 million data records were logged and evaluated at RWTH Aachen. My analysis showed that the software was used in courses with thousands of users showing that the software is well scalable even for large courses. Still, the logged data only needed a few gigabytes of database storage capacity. The collected data was evaluated to show how the Moodle platform of RWTH Aachen was used. Several access patterns of students were discovered in the process. The IT Center of RWTH Aachen has already extended the software to show statistics regarding views of their video platform. The provided software interfaces to extend the software are therefore already used in production today. The research questions how a learning analytics system can be successfully developed and integrated into an LMS are therefore answered. The research question how changing requirements and future use cases can be supported is answered by providing a flexible plugin system for the software.

Over the course of two semesters, the software was used in over 700 courses at RWTH Aachen and in more courses at other universities. The follow-up evaluations showed that the software was well received by lecturers. Table 39 summarizes both case studies and participants.

Table 39: Overview of case studies and participating universities

University	Participated lecturers	Participated courses	Evaluated questionnaires	Logged data in million records
Summer semester 2020				
RWTH	31	59	20	44.0
HRW	32	93	23	10.6
Winter semester 2020/21				
RWTH	~2,000*	~650	163	71.5
HRW	33	97	15	12.0
HNU	6	10	4	*

* Exact data unknown

The created learning analytics software does not log any personalized data. However, while the feedback indicated great interest in personalized reports and individual visualizations, overall the provided features were well perceived and rated as rather useful. In addition, the usability was rated highly above average.

Talks to other universities and IT operators revealed that data privacy is still one of the main issues for further adoption of learning analytics. My solution, which does not rely on personal data, was therefore well received. The research questions on how the barriers for a greater adoption of learning analytics can be lowered can therefore be answered: It is important that learning analytics software incorporates data privacy and protection as a key feature. In my case, this issue was handled by not logging personal data at all resulting in a fast adoption of the software at multiple universities.

Overall, the evaluation has shown that a learning analytics application, which is integrated into the LMS and based on anonymous log data, can support lecturers and benefit teaching. The developed software and its features were well received by lecturers. Feedback from IT operators showed that the software was easy to integrate and operate. I therefore conclude that this approach towards learning analytics can benefit the overall adoption of learning analytics as currently basic descriptive analytics features are missing in many German universities.

Due to the COVID-19 pandemic, participating universities switched from face-to-face lectures to blended learning scenarios or even full video lectures. This has accelerated the process of digitalization in higher education. It remains to be seen if this trend will continue after a possible step back to classic face-to-face lectures. As of now, learning analytics must be considered an essential tool to support teaching.

8 Conclusion

In this dissertation, I presented a data privacy-friendly approach for providing learning analytics for the Moodle learning management system. This chapter summarizes the key findings, their assessments, and provides future research and development directions.

8.1 Summary

In the first chapter, the motivation for this dissertation was explained together with the corresponding research questions.

The second chapter presented the fundamentals for the thesis. The term learning analytics was defined and related research fields were presented, as well as networks, organizations, and initiatives that supported or implemented learning analytics in the past. As it is a central part of this dissertation, the learning management system Moodle was presented as well as its use in higher education. Lastly, data privacy and ethical considerations as well as applicable regulations were outlined.

In the third chapter, the related work was presented. Software architectures for learning analytics were discussed. Only few existing initiatives are providing a usable software implementation that is published as open source. Other architectures rely on third parties to provide essential services. Afterwards, popular learning analytics tools, general-purpose analytics solutions, and learning records stores were presented. In conclusion, there are many implementations mentioned in literature, often presented in detail and even evaluated. Unfortunately, only rarely the corresponding source code is published. The majority of software implementations that was examined in this dissertation was developed for Moodle. Over thirty tools were investigated. However, a large part of the tools was either outdated or not aimed at providing analytics for lecturers or students. The only exception were commercial solutions. Lastly, the chapter discussed evaluation methods and presented how other learning analytics implementations were evaluated.

The fourth chapter described the development of two prototypes created prior to the final software implementation. The first approach used data science methods to analyze Moodle log data combined with data from outside of the LMS like student performance data. Several examples were presented showing the potential of personalized learning analytics. The second prototype explored the integration into Moodle. It used an external server to store the data combined with a Moodle plugin for visualization. Evaluations showed that the external server was in fact not necessary as computations could be optimized to perform them inside of Moodle.

Based on the experience from the prior development as well as the related work, the analysis was presented in the fifth chapter. It summarized the findings from the preliminary work and discussed common issues including personal data collection and its privacy implications. It was outlined how learning analytics can be provided as a university-wide service by explaining the overall process as well as defining the software requirements necessary for the implementation.

In Chapter 6, the actual software implementation was presented starting from the big picture of the software architecture down to the implementation of the Moodle plugins. The created software is split into two Moodle plugins: one for logging and one for the user interface. Implementation and operational details were discussed. Afterwards, the used visualizations for two case studies were presented.

The seventh chapter describes the conducted evaluation. After explaining the methodology, the collected log data was presented to provide a context for the following evaluation. The data analysis was based on over 115 million log records collected over two semesters at RWTH Aachen University. It revealed typical access patterns of student activity inside the LMS. Comparisons to two other universities were presented to show similarities and differences. Afterwards, the technical evaluation was conducted verifying the scalability, extensibility, and transferability of the software. Overall, the software performed well in small courses as well as large courses with thousands of students. In addition, the software can be extended and installed easily by other interested parties. After the technical evaluation, two longitudinal case studies were presented. The case studies incorporated user feedback based on questionnaires containing items related to the usability, usage, and usefulness of individual features. After conducting the first case study, the results were applied to the software and the case study was repeated with more lecturers. Overall, the implementation received good feedback. The evaluation showed that many lecturers were incorporating the learning analytics software into their daily teaching activities.

8.2 Adoption

My software, the surrounding research, and parts of the results have been presented at conferences and workshops in the past. The software was presented in November 2020 at the *E-Prüfungs-Symposium* (ePS), a local conference in Germany, which was focused on learning analytics in that year. In addition, the software was presented to an international audience at the pre-conference workshop *Responsible Learning Analytics* of the *LAK21* conference, the *11th International Conference on Learning Analytics and Knowledge* (Dondorf et al., 2021). Both times, the software and its evaluation received good feedback from the audience and interested users contacted me afterwards for more information. Based on the feedback I received, there seems to be a great interest in ready-to-use data privacy-friendly learning analytics.

The software was published as open source on GitHub⁸². Multiple universities and interested parties have approached me. Currently, four universities in Germany are using the software in their production Moodle system. Two more universities from outside of Germany have deployed the learning analytics software, and three more universities have contacted me and stated that they are currently testing the software or considering its use for future semesters.

⁸² <https://github.com/rwthanalytics>

The created learning analytics software was published to the Moodle plugin store in June 2021. It is currently waiting for approval by Moodle developers. The release of the software in the plugin store will further lowers the barriers to adopt the software for interested users and educational institutions.

8.3 Future Research and Development Directions

This dissertation provides an evaluated, production-ready, and easy-to-use learning analytics solution to higher education institutions. Nevertheless, future research work is necessary to improve upon this work.

First, there are multiple possible extensions of the developed software. Currently, the implementation only comes with logging utilities that store anonymous records of LMS activity. To provide personalized learning analytics a new logging software is necessary that includes personalized data like the user identification. However, this challenge itself is not a technical one but a legal and ethical one as a process is necessary to receive and manage the consent of users regarding the data collection.

The implementation and embedding of personalized statistics would also allow providing personalized learning analytics for students. With one exception in the evaluation, this thesis only focused on learning analytics for teachers. Future research and development should include students, especially when designing dashboards and coming up with new use cases.

Supporting personalized learning analytics also means that it is possible to use data from different systems. That makes it possible to merge data from the Moodle LMS with performance data from the student information system for individual users. This would allow detailed visualizations and even correlations similar to what was achieved in the first prototype.

After the implementation of personalized learning analytics, the main challenge is to connect the learning analytics results to the learning and learner models. Right now, the evaluations are based on feedback from lecturers and students but further research is necessary to prove that the use of learning analytics can significantly improve the teaching and learning processes. For this, theories from educational sciences need to be applied to learner models and verified based on data from learning analytics. The anonymous data I presented in this dissertation can serve as a first step in that direction but in the long term, learning analytics reports and visualizations will benefit from personalized data.

Another future research direction might be the incorporation of other fields of research like data mining or machine learning. Tools like *Apache Hadoop* allow the processing of big data to analyze billions of records. These tools could be integrated as external solution to provide classical data science approaches. Additionally, the use of machine learning and artificial intelligence has become a hot topic in the field of learning analytics with tools being able to predict performances of students or recommend suitable courses based on past activity.

On the implementation side, some enhancements to the software interface should be implemented in the future. One enhancement is the option to filter the visualizations to specific periods as requested by lecturers in the evaluation. That would make it possible to check how the course was used before the exam period and compare this to the results from the exam period. One purely technical improvement to the software is the aggregation of log data for old semesters. This would remove the options to evaluate the data but would also drive down the

required database storage. Right now, the amount of stored data is still manageable by the IT department. Nevertheless, in the long term this issue needs to be taken care of.

Another improvement would be the addition of administrative reports and visualizations. Similar to the data analysis conducted in Chapter 7, this could allow administrators to optimize the Moodle platform and researchers to gain insights on the LMS activity. These kinds of reports could be implemented in the future by using cron jobs running on a regular basis (e.g. daily) to produce computationally intensive evaluations.

In conclusion, this work has shown how a learning analytics software can be implemented, deployed, and provided to other universities. My implementation has been used in thousands of courses across multiple universities. The feedback I personally received has been positive and supportive. Members of the learning analytics community as well as members of the Moodle community have shown great interest in the developed software. It can serve as a first introduction into the topic for universities struggling with the adoption of learning analytics. However, the created work in this dissertation is not the final, but the first step towards a greater adoption of learning analytics in higher education.

Bibliography

- ADL. (n.d.). *Advanced Distributed Learning Initiative. Sharable Content Object Reference Model (SCORM®)*. Retrieved May 12, 2021, from <https://adlnet.gov/projects/scorm/>
- ADL. (2017). *Experience API: xAPI-Spec*. Advanced Distributed Learning (ADL) Initiative, U.S. Department of Defense. Retrieved April 1, 2021, from <https://github.com/adlnet/xAPI-Spec>
- ADL. (2021). *Conformant LRSs*. Advanced Distributed Learning (ADL) Initiative, U.S. Department of Defense. Retrieved May 12, 2021, from <https://adopters.adlnet.gov/products/all/0>
- Aguilar, D. A. G., Therón, R., & Peñalvo, F. G. (2008). Understanding Educational Relationships in Moodle with ViMoodle. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*.
- Aguilar, S., Lonn, S., & Teasley, S. D. (2014). Perceptions and use of an early warning system during a higher education transition program. In *Proceedings of the fourth international conference on learning analytics and knowledge*.
- Akçapınar, G., & Bayazıt, A. (2019). MoodleMiner: Data Mining Analysis Tool for Moodle Learning Management System. *Elementary Education Online*, 18(1).
- Aladwani, A. M., & Palvia, P. C. (2002). Developing and validating an instrument for measuring user-perceived web quality. *Information & Management*, 39(6), 467–476. [https://doi.org/10.1016/S0378-7206\(01\)00113-6](https://doi.org/10.1016/S0378-7206(01)00113-6)
- Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Sharifzadeh, S., Tresp, V., & Lehmann, J. (2020). Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. *ArXiv Preprint ArXiv:2007.14175*.
- Alonso-Fernández, C., Martínez-Ortiz, I., Caballero, R., Freire, M., & Fernández-Manjón, B. (2020). Predicting students' knowledge after playing a serious game based on learning analytics data: A case study. *Journal of Computer Assisted Learning*, 36(3), 350–358.
- Alstrup, S., Hansen, C., Hansen, C., Hjuler, N., Lorenzen, S., & Pham, N. (2017). DABAI: A data driven project for e-Learning in Denmark. In A. Mesquita (Ed.), *European Conference on e-Learning* (pp. 18–24). Academic Conferences International Limited.
- Alves, G. R., Viegas, M. C., Marques, M. A., Costa-Lobo, M. C., Silva, A. A., Formanski, F., & Silva, J. B. (2012). Student performance analysis under different Moodle course designs. In *2012 15th International Conference on Interactive Collaborative Learning (ICL)* (pp. 1–5). IEEE.

- Anderson, D., & Seck, R. von (2020). The GDPR and its impact on the web. *Network Architectures and Services*, 1.
- Apereo. (2015). *Learning Analytics Initiative*. Retrieved January 15, 2021, from <https://www.apereo.org/communities/learning-analytics-initiative>
- Apereo. (2016). *Learning Analytics Initiative*. Retrieved May 12, 2021, from <http://web.archive.org/web/20181021054525/https://www.apereo.org/communities/learning-analytics-initiative>
- Apereo. (2020a). *Apereo Membership*. Retrieved May 11, 2021, from <https://www.apereo.org/content/apereo-membership>
- Apereo. (2020b). *Projects & Communities*. Retrieved May 10, 2021, from <https://www.apereo.org/content/projects-communities>
- Appelbaum, D., Kogan, A., Vasarhelyi, M., & Yan, Z. (2017). Impact of business analytics and enterprise systems on managerial accounting. *International Journal of Accounting Information Systems*, 25, 29–44.
- Arnold, K. E. (2010). Signals: Applying academic analytics. *Educause Quarterly*, 33(1).
- Arnold, K. E., & Pistilli, M. D. (2012). Course signals at Purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- Atif, A., Bilgin, A., & Richards, D. (2015). Student preferences and attitudes to the use of early alerts.
- Avouris, N., Komis, V., Fiotakis, G., Margaritis, M., & Voyiatzaki, E. (2005). Logging of fingertip actions is not enough for analysis of learning activities. In *12th International Conference on Artificial Intelligence in Education, AIED 05 Workshop 1: Usage analysis in learning systems*.
- Ayyagari, R. (2012). An exploratory analysis of data breaches from 2005-2011: Trends and insights. *Journal of Information Privacy and Security*, 8(2), 33–56.
- Baepler, P., & Murdoch, C. J. (2010). Academic analytics and data mining in higher education. *International Journal for the Scholarship of Teaching and Learning*, 4(2), 1–9.
- Baker, B. M. (2007). *A conceptual framework for making knowledge actionable through capital formation*. University of Maryland University College.
- Baker, R. S. d., Duval, E., Stamper, J., Wiley, D., & Shum, S. B. (2012). Educational data mining meets learning analytics. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- Bakharia, A., & Dawson, S. (2011). SNAPP: a bird's-eye view of temporal participant interaction. In *Proceedings of the 1st international conference on learning analytics and knowledge*.
- Banerjee, A., Bandyopadhyay, T., & Acharya, P. (2013). Data analytics: Hyped up aspirations or true potential? *Vikalpa*, 38(4), 1–12.
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114–123.
- Bargas-Avila, J. A. (2001). *Evaluationmethoden der Mensch-Maschine-Interaktion im eCommerce. Lizentiatsarbeit. Institut für Psychologie - Universität Basel*. Referenced on www.soscisurvey.de

- Behrmann, M., Pflug, H.-J., & Dondorf, T. (2017). *Erstellung eines Moodle Plugins zur Darstellung von Statistiken und Grafiken eines Learning Analytics Servers: Bachelorarbeit im Studiengang "Scientific Programming" Fachbereich Medizintechnik und Technomathematik der Fachhochschule Aachen, Campus Jülich.*
- Bekavac, I., & Garbin Praničević, D. (2015). Web analytics tools and web metrics tools: An overview and comparative analysis. *Croatian Operational Research Review*, 6(2), 373–386.
- Bellman, S., Johnson, E. J., Kobrin, S. J., & Lohse, G. L. (2004). International differences in information privacy concerns: A global survey of consumers. *The Information Society*, 20(5), 313–324.
- Beri, B., & Singh, P. (2013). Web analytics: Increasing website's usability and conversion rate. *International Journal of Computer Applications*, 72(6).
- Berking, P. (2016). *Choosing a learning record store (LRS)*. Advanced Distributed Learning (ADL) Initiative, U.S. Department of Defense. Retrieved April 1, 2021, from <https://adlnet.gov/assets/uploads/ChoosingAnLRS.pdf>
- Berkman, M. I., & Karahoca, D. (2016). Re-assessing the usability metric for user experience (UMUX) scale. *Journal of Usability Studies*, 11(3), 89–109.
- Betts, B. (2015). *Jisc Selects Learning Locker to Power Open Learning Analytics Service*. Retrieved May 11, 2021, from <https://learningpool.com/jisc-selects-learning-locker-to-power-open-learning-analytics-service/>
- Bias, R. G. (1994). The pluralistic usability walkthrough: coordinated empathies. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 63–76). John Wiley & Sons, Inc.
- Boda, K., Földes, Á. M., Gulyás, G. G., & Imre, S. (2011). User tracking on the web via cross-browser fingerprinting. In *Nordic conference on secure it systems* (pp. 31–46). Springer.
- Borbon, P. (2020). *X-Ray Learning Analytics - OpenLMS*. Retrieved May 21, 2021, from <https://help.openlms.net/en/instructor/reports/x-ray-learning-analytics/>
- Bramucci, R., & Gaston, J. (2012). Sherpa: Increasing student success with a recommendation engine. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- Brath, R., & Peters, M. (2004). Dashboard design: Why design is important. *DM Direct*, 85, 1011285-1.
- Bray, T. (2017). *The JavaScript Object Notation (JSON) Data Interchange Format*. Retrieved June 21, 2021, from <https://datatracker.ietf.org/doc/html/rfc8259>
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.
- Brooke, J. (1996). SUS: a “quick and dirty” usability scale. *Usability Evaluation in Industry*, 189-194.
- Brown, M., McCormack, M., Reeves, J., Brooks, C. D., Grajek, S., Alexander, B., Bali, M., Bulger, S., Dark, S., Engelbert, N., Gannon, K., Gauthier, A., Gibson, D., Gibson, R., Lundin, B., Veletsianos, G., & Weber, N. (2020). *EDUCAUSE Horizon Report: Teaching and Learning Edition*. Educause.
- Bundesdatenschutzgesetz. (2018). *Bundesdatenschutzgesetz*. Bundesministerium für Justiz und für Verbraucherschutz.

- Caeiro-Rodríguez, M., Conde, M. Á., Guenaga, M., Hernández-García, Á., Larrañaga, M., Martínez-Monés, A., Muñoz-Merino, P. J., Pastor-Vargas, R., Perillos-Ruiz, A., & Rodríguez-Conde, M.-J. (2016). SNOLA: Spanish network of learning analytics. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*.
- Campbell, J. P., DeBlois, P. B., & Oblinger, D. G. (2007). Academic analytics: A new tool for a new era. *EDUCAUSE Review*, 42(4), 40.
- Campbell, J. P., & Oblinger, D. G. (2007). Academic analytics. *Educause Quarterly*.
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.
- Cavoukian, A. (2009). Privacy by Design: The 7 Foundational Principles. *Information & Privacy Commissioner, Ontario, Canada*.
- Chaffey, D., & Patron, M. (2012). From web analytics to digital marketing optimization: Increasing the commercial value of digital analytics. *Journal of Direct, Data and Digital Marketing Practice*, 14(1), 30–45.
- Chassang, G. (2017). The impact of the EU general data protection regulation on scientific research. *Ecancermedalscience*, 11.
- Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, 4(5-6), 318–331.
- Chatti, M. A., Muslim, A., & Schroeder, U. (2017). Toward an open learning analytics ecosystem. In *Big data and learning analytics in higher education* (pp. 195–219). Springer.
- Chin, J. P., Diehl, V. A., & Norman, L. K. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In J. J. O'Hare (Ed.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 213–218). ACM. <https://doi.org/10.1145/57167.57203>
- Cho, Y.-S., Hoel, T., & Chen, W. (2016). Mapping a Privacy Framework to a Reference Model of Learning Analytics. Retrieved June 4, 2019, from https://lace.apps.slate.uib.no/wp-content/uploads/2015/12/ep4la2016_paper_4.pdf
- Clark, D. J., Nicholas, D., & Jamali, H. R. (2014). Evaluating information seeking and use in the changing virtual world: the emerging role of Google Analytics. *Learned Publishing*, 27(3), 185–194.
- Clow, D. (2012). The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- Codd, E. F. (2002). A relational model of data for large shared data banks. In *Software pioneers* (pp. 263–294). Springer.
- Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. *IEEE Transactions on Learning Technologies*, 10(1), 17–29. <https://doi.org/10.1109/TLT.2016.2616312>
- Croll, A. (2012). *Big data is our generation's civil rights issue, and we don't know it*. O'Reilly Radar. Retrieved April 15, 2021, from <http://radar.oreilly.com/2012/08/big-data-is-our-generations-civil-rights-issue-and-we-dont-know-it.html>
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *PSYCHOMETRIKA*, 16(3), 297–334.

- Daniel, J. (2020). Education and the COVID-19 pandemic. *Prospects*, 49(1), 91–96.
- Davenport, T., & Harris, J. (2017). *Competing on analytics: Updated, with a new introduction: The new science of winning*. Harvard Business Press.
- Decker, B., Politze, M., & Grzemski, S. (2018). From a SharePoint-Based E-Learning System to Moodle. In *SIGUCCS'18*, Orlando, FL, USA.
- Delen, D., & Demirkan, H. (2013). *Data, information and analytics as services*. Elsevier.
- DER SPIEGEL (1971). EDV im Odenwald, Article DER SPIEGEL 20/1971, 88. Retrieved June 16, 2021, from <https://www.spiegel.de/politik/edv-im-odenwald-a-0746bb53-0002-0001-0000-000043176393>
- Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human-Computer Interaction*. Prentice Hall.
- Dondorf, T., Breuer, R., & Nacken, H. (2016). Nutzung und Erweiterung von Moodle für täglichen Hochschulbetrieb. In *Die offene Hochschule: Vernetztes Lehren und Lernen* (pp. 322–323). Waxmann.
- Dondorf, T., Persike, M., & Nacken, H. (2021). Learning Analytics without personal data? It's possible! In M. Scheffel, N. Dowell, S. Joksimovic, & G. Siemens (Eds.), *Companion Proceedings of the 11th International Conference on Learning Analytics & Knowledge* (pp. 351–356). Society for Learning Analytics Research (SoLAR).
- Dondorf, T., Pyka, C., Gramlich, R., Sewilam, H., & Nacken, H. (2019). Learning analytics software implementation for the moodle learning management system. In *Proceedings of the 12th annual International Conference of Education, Research and Innovation*, Seville, Spain. <http://dx.doi.org/10.21125/iceri.2019.1655>
- DR. (2011). *Learning Registry Paradata Specification*. Retrieved March 24, 2021, from https://docs.google.com/document/d/1lrOYXd3S0FUwNozaEG5tM7Ki4_AZPrBn-pbyVUz-Bh0/edit
- Drachsler, H., & Greller, W. (2016). Privacy and Analytics: It's a DELICATE Issue a Checklist for Trusted Learning Analytics. In *LAK '16, Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (pp. 89–98). Association for Computing Machinery. <https://doi.org/10.1145/2883851.2883893>
- Drachsler, H., & Scheffel, M. (2016). LACE Project Exit Action Plan. Retrieved June 2, 2021, from <https://core.ac.uk/download/pdf/74509477.pdf>
- Dragulescu, B., Bucos, M., & Vasiu, R. (2015). CVLA: integrating multiple analytics techniques in a custom moodle report. In *International conference on information and software technologies* (pp. 115–126). Springer.
- Dron, J., & Anderson, T. (2009). On the design of collective applications. In *2009 International Conference on Computational Science and Engineering* (Vol. 4, pp. 368–374). IEEE.
- DSG NRW. (2018). *Datenschutzgesetz Nordrhein-Westfalen (DSG NRW)*. Retrieved April 16, 2021, from https://recht.nrw.de/lmi/owa/pl_text_anzeigen?v_id=3520071121100436275
- Duda, R. O., Hart, P. E., & others. (1973). *Pattern classification and scene analysis* (Vol. 3). Wiley New York.
- Dyckhoff, A. L. (2014). *Action Research and Learning Analytics in Higher Education. Dissertation*. RWTH Aachen University.

- EDUCAUSE. (2016). *Educational Community License*. Retrieved May 12, 2021, from <https://web.archive.org/web/20111017184126/http://www.educause.edu/wiki/Educational+Community+License>
- Einhardt, L., Tavares, T. A., & Cechinel, C. (2016). Moodle analytics dashboard: A learning analytics tool to visualize users interactions in moodle. In *2016 XI Latin American Conference on Learning Objects and Technology (LACLO)*.
- Elias, T. (2011). Learning analytics: Definitions, Processes and Potential.
- Elkina, M., Fortenbacher, A., & Merceron, A. (2014). Lehren mit Learning Analytics - erste Erfahrungen mit dem Tool LeMo. *Proceedings of Die 12. E-Learning Fachtagung Informatik Der Gesellschaft Für Informatik EV*, 15–17.
- Ellis, D. (2012). Google Analytics as a tool in the development of e-learning artefacts: A case study. In *Brown, M.; Hartnett, M.; T. Stewart (eds.), Wellington, Nueva Zelanda. Actas de ASCILITE. Australian Society for Computers in Learning in Tertiary Education Annual Conference*.
- Essa, A., & Ayad, H. (2012). Student success system: risk analytics and data visualization using ensembles of predictive models. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- EU General Data Protection Regulation (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*(L 119/88).
- Evans, J. D. (1996). *Straightforward statistics for the behavioral sciences*. Thomson Brooks/Cole Publishing Co.
- Farzan, R., & Brusilovsky, P. (2006). Social navigation support in a course recommendation system. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 91–100). Springer.
- Fekete, J.-D., van Wijk, J. J., Stasko, J. T., & North, C. (2008). The value of information visualization. In *Information visualization* (pp. 1–18). Springer.
- Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5-6), 304–317.
- Ferguson, R., & Clow, D. (2017). Where is the evidence? A call to action for learning analytics. In *Proceedings of the seventh international learning analytics & knowledge conference*.
- Few, S. (2006). *Information dashboard design: The effective visual communication of data* (Vol. 2). O'reilly Sebastopol, CA.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Vol. 7). University of California, Irvine Irvine.
- Filvà, D. A., Guerrero, M. J. C., & Forment, M. A. (2014). Google analytics for time behavior measurement in Moodle. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). IEEE.
- Finstad, K. (2006). The system usability scale and non-native English speakers. *Journal of Usability Studies*, 1(4), 185–188.

- Finstad, K. (2010). The Usability Metric for User Experience. *Interacting with Computers*, 22(5), 323–327. <https://doi.org/10.1016/j.intcom.2010.04.004>
- Fortenbacher, A., Beuster, L., Elkina, M., Kappe, L., Merceron, A., Pursian, A., Schwarzrock, S., & Wenzlaff, B. (2013). LeMo: A learning analytics application focussing on user path analysis and interactive visualization. In *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)* (Vol. 2, pp. 748–753). IEEE.
- Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics*. Pisani, R. Purves, 4th Edn. WW Norton & Company, New York.
- Furukawa, M., Yamaji, K., Yaginuma, Y., & Yamada, T. (2017). Development of learning analytics platform for OUJ online courses. In *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*.
- García, E., Romero, C., Ventura, S., & Castro, C. de (2009). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction*, 19(1-2), 99–132.
- Gartner, Inc. (2020). *Market Guide for Higher Education Learning Management Systems* (G00732448). G. Morgan, T. Sheehan.
- Gaudio, E., Montero, M., Talavera, L., & Hernandez-del-Olmo, F. (2009). Supporting teachers in collaborative student modeling: A framework and an implementation. *Expert Systems with Applications*, 36(2), 2260–2265.
- Genkov, D., & Kesova, V. (2014). Building an e-learning system in Technical university of Gabrovo, Bulgaria. *Proceedings of EDULEARN*, 14, 619–626.
- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference: 11.0 update* (4th edition). Boston: Allyn & Bacon.
- Gibson, D., Ostashevski, N., Flintoff, K., Grant, S., & Knight, E. (2015). Digital badges in education. *Education and Information Technologies*, 20(2), 403–410.
- Gieseke, J. (2011). *Die Stasi: 1945 - 1990* (Aktualis. u. erw. Ausg). Pantheon Verl.
- Gkontzis, A. F., Karachristos, C. V., Lazarinis, F., Stavropoulos, E. C., Verykios, V. S., Ubachs, G., & Konings, L. (2017). A holistic view on academic wide data through learning analytics dashboards. In *Conference Proceedings: The Online, Open and Flexible Higher Education Conference*.
- Govaerts, S., Verbert, K., & Duval, E. (2011). Evaluating the student activity meter: two case studies. In *International Conference on Web-Based Learning* (pp. 188–197). Springer.
- Govaerts, S., Verbert, K., Duval, E., & Pardo, A. (2012). The student activity meter for awareness and self-reflection. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 869–884).
- Graf, S., Ives, C., Rahman, N., & Ferri, A. (2011). AAT: a tool for accessing and analysing students' behaviour data in learning systems. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (pp. 174–179). Association for Computing Machinery.
- Graf, S. J. F. (2009). *Durchgängiges Identity Management und interoperable E-Portfolios zur Unterstützung lebenslangen Lernens*.

- Green, K. C. (2013). *The National Survey of Computing and Information Technology*. The Campus Computing Project.
- Griffiths, D., Hoel, T., & Cooper, A. (2016). Learning analytics interoperability: Requirements, specifications and adoption. *Public Deliverable D*, 7.
- Gupta, M., & Sharman, R. (2012). Determinants of Data Breaches: A Categorization-Based Empirical Investigation. *Journal of Applied Security Research*, 7(3), 375–395.
- Heer, J., Bostock, M., & Ogievetsky, V. (2010). A tour through the visualization zoo. *Communications of the ACM*, 53(6), 59–67.
- Herodotou, C., Rienties, B., Boroowa, A., Zdrahal, Z., & Hlosta, M. (2019). A large-scale implementation of predictive learning analytics in higher education: the teachers' role and perspective. *Educational Technology Research and Development*, 67(5), 1273–1306.
- Herodotou, C., Rienties, B., Boroowa, A., Zdrahal, Z., Hlosta, M., & Naydenova, G. (2017). Implementing predictive learning analytics on a large scale: the teacher's perspective. In *Proceedings of the seventh international learning analytics & knowledge conference*.
- Herodotou, C., Rienties, B., Hlosta, M., Boroowa, A., Mangafa, C., & Zdrahal, Z. (2020). The scalable implementation of predictive learning analytics at a distance learning university: Insights from a longitudinal case study. *The Internet and Higher Education*, 45, 100725.
- Hochschulgesetz NRW. (2014). *Gesetz über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz - HG) vom 16.09.2014*.
- Hoel, T., & Xiao, J. (2018). Towards an open architecture for learning analytics in open and distance education. In *Asia-Pacific Society for Computers in Education*.
- Hollingsed, T., & Novick, D. G. (2007). Usability Inspection Methods after 15 Years of Research and Practice. In *Proceedings of the 25th annual ACM international conference on Design of communication (SIGDOC '07)*, El Paso, Texas, USA.
- Hunt, T. (2012). Moodle. In A. Brown & G. Wilson (Eds.), *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*. AOSA. Retrieved June 21, 2021, from <https://www.aosabook.org/en/moodle.html>
- Hwang, G.-J., Tsai, P.-S., Tsai, C.-C., & Tseng, J. C. R. (2008). A novel approach for assisting teachers in analyzing student web-searching behaviors. *Computers & Education*, 51(2), 926–938.
- IEEE. (1998). *Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. The Institute of Electrical and Electronics Engineers, Inc. Software Engineering Standards Committee of the IEEE Computer Society.
- Ifenthaler, D. (2017). Are Higher Education Institutions Prepared for Learning Analytics? *TechTrends*, 61(4), 366–371. <https://doi.org/10.1007/s11528-016-0154-0>
- Ifenthaler, D., & Schumacher, C. (2016). Learning Analytics im Hochschulkontext. *WiSt-Wirtschaftswissenschaftliches Studium*, 45(4), 176–181.
- Ifenthaler, D., & Yau, J. (2019). Higher Education Experts' Views on Learning Analytics Policy Recommendations for Supporting Study Success: A German Case. In J. T. Bastiaens (Ed.), *Proceedings of EdMedia + Innovate Learning 2019* (pp. 745–753). Association for the Advancement of Computing in Education (AACE). Retrieved April 6, 2021, from <https://www.learntechlib.org/p/210073>

- IMS Global. (2015). *Caliper Analytics*. Retrieved May 12, 2021, from <http://www.imsglobal.org/activity/caliper>
- IntelliBoard, I. (2021). *IntelliBoard*. Retrieved October 8, 2020, from <https://www.intelliboard.net/>
- ISO (2018). ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Requirements engineering. *ISO/IEC/IEEE 29148:2018(E)*, 1–104. <https://doi.org/10.1109/IEEESTD.2018.8559686>
- ISO/IEC. (2016). *Information technology – learning, education, and training – Learning Analytics Interoperability – Part 1: Reference model: ISO/IEC TR 20748-1*. ISO.
- Jacoby, W. G. (1997). *Statistical graphics for univariate and bivariate data* (Vol. 117). Sage.
- Jisc. (n.d.–a). *About us*. Retrieved May 10, 2021, from <https://www.jisc.ac.uk/about>
- Jisc. (n.d.–b). *How we are funded*. Retrieved May 10, 2021, from <https://www.jisc.ac.uk/about/corporate/how-we-are-funded>
- Jisc. (2018a). *Learning Analytics: General Overview*. Retrieved May 11, 2021, from <https://docs.analytics.alpha.jisc.ac.uk/docs/learning-analytics/General-Overview>
- Jisc. (2018b). *Learning Analytics: Security FAQs*. Retrieved May 11, 2021, from <https://docs.analytics.alpha.jisc.ac.uk/docs/learning-analytics/Security-FAQs>
- Johnson, D., & Samora, D. (2016). The potential transformation of higher education through computer-based adaptive learning systems. *Global Education Journal*, 2016(1).
- Johnson, L., Smith, R., Willis, H., Levine, A., & Haywood, K. (2011). *The 2011 Horizon Report*.
- Jovanovic, J., Knight, C., Gasevic, D., & Richards, G. (2006). Learning object context on the semantic web. In *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp. 669–673). IEEE.
- Jovanović, J., Gašević, D., Brooks, C., Devedžić, V., & Hatala, M. (2007). LOCO-analyst: A tool for raising teachers' awareness in online learning environments. In *European Conference on Technology Enhanced Learning* (pp. 112–126). Springer.
- Junus, I. S., Santoso, H. B., Isal, R. Y. K., & Utomo, A. Y. (2015). Usability evaluation of the student centered e-learning environment. *International Review of Research in Open and Distributed Learning*, 16(4), 62–82.
- Kadoić, N., & Oreški, D. (2018). Analysis of student behavior and success based on logs in Moodle. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 654–659). IEEE.
- Karunakaran, S., Thomas, K., Bursztein, E., & Comanescu, O. (2018). Data breaches: user comprehension, expectations, and concerns with handling exposed data. In *Fourteenth Symposium on Usable Privacy and Security (5SOUPS6 2018)*.
- keepachangelog.com. (2017). *Keep A Changelog 1.0.0 English*. Retrieved April 23, 2021, from <https://keepachangelog.com/en/1.0.0/>
- Kerres, M. (2020). Against All Odds: Education in Germany Coping with Covid-19. *Postdigital Science and Education*, 2(3), 690–694. <https://doi.org/10.1007/s42438-020-00130-7>
- Kimball, R., & Ross, M. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley Computer Publishing.

- Kirakowski, J., & Cierlik, B. (1998). Measuring the usability of web sites. In *Proceedings of the Human Factors and Ergonomics Society annual meeting* (Vol. 42, pp. 424–428). Sage Publications. <https://doi.org/10.1177/154193129804200405>
- Kirakowski, J., & Corbett, M. (1993). SUMI: The software usability measurement inventory. *British Journal of Educational Technology*, 24(3), 210–212.
- Klasen, D., & Ifenthaler, D. (2019). Implementing Learning Analytics into Existing Higher Education Legacy Systems. In D. Ifenthaler, D.-K. Mah, & J. Y.-K. Yau (Eds.), *Utilizing Learning Analytics to Support Study Success* (pp. 61–72). Springer International Publishing. https://doi.org/10.1007/978-3-319-64792-0_4
- Klindžić, J., Lazić, N., & Perković, M. (2019). Implementation and Tech Support for Moodle-based MOOC for Language Learning. In *Proceedings of the 2019 3rd International Conference on Education and E-Learning*.
- Kochnev, A. (2021). *Moodle plugins directory: IntelliBoard - Your data. Right here. Right now.* Retrieved May 21, 2021, from https://moodle.org/plugins/local_intelliboard
- Koedinger, K. (2019). *educationaldatamining.org: hosted by the International Educational Data Mining Society.* Retrieved March 16, 2021, from <https://educationaldatamining.org/>
- Kolb, D.A. (1984). *Experiential Learning: Experience as the source of learning and development.* Prentice Hall.
- Kortum, P. T., & Bangor, A. (2013). Usability ratings for everyday products measured with the system usability scale. *International Journal of Human-Computer Interaction*, 29(2), 67–76.
- Krasnova, H., & Veltri, N. F. (2010). Privacy calculus on social networking sites: Explorative evidence from Germany and USA. In *2010 43rd Hawaii international conference on system sciences* (pp. 1–10). IEEE.
- Krieg, A. (2020). *Ausblick auf die Lehre im Jahr 2021: Talk Lehre 12.11.2020.* Retrieved April 14, 2021, from https://www.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaavlespf
- Krumm, A. E., Waddington, R. J., Teasley, S. D., & Lonn, S. (2014). A learning management system-based early warning system for academic advising in undergraduate engineering. In *Learning analytics* (pp. 103–119). Springer.
- Kubiak, S., & Grzemeski, S. (2019). User-oriented introduction of a digital learning platform at rwth aachen university: a field report of first-level support. In *11th International Conference on Education and New Learning Technologies, EDULEARN19 Proceedings* (pp. 6346–6355). IATED. <https://doi.org/10.21125/edulearn.2019.1520>
- Kuzilek, J., Hlosta, M., Herrmannova, D., Zdrahal, Z., Vaclavek, J., & Wolff, A. (2015). OU Analyse: analysing at-risk students at The Open University. *Learning Analytics Review, LAK15-1*, 1–16.
- Labba, C., Roussanaly, A., & Boyer, A. (2020). An Operational Framework for Evaluating the Performance of Learning Record Stores. In *European Conference on Technology Enhanced Learning* (pp. 45–59). Springer.
- LACE. (2015). *About LACE - Learning Analytics Community Exchange.* <https://web.archive.org/web/20150921234714/http://www.laceproject.eu/lace/>
- LACE. (2016a). *Associate Partners - LACE - Learning Analytics Community Exchange.* <https://web.archive.org/web/20160709090500/http://www.laceproject.eu/partners/>

- LACE. (2016b). *SHEILA project joins SoLAR SIG LACE - Learning Analytics Community Exchange*. <https://web.archive.org/web/20200208204135/http://www.laceproject.eu/blog/sheila-project-joins-solar-sig-lace/>
- Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group* (pp. 63–76). Springer.
- Laurillard, D. M. (2002). *Rethinking University Teaching: a conversational framework for the effective use of learning technologies*. Routledge Falmer.
- Leece, R., & Hale, R. (2009). *Student Engagement and Retention Through Emotional Intelligence*. Retrieved April 9, 2021, from https://wiki.mq.edu.au/download/attachments/98173671/Leece_E-Motion.pdf
- Leicht-Scholten, C., & Nacken, H. (2010). Mobilising Creativity. Das Zusammenspiel der Zukunftskonzepte Forschung und Lehre an der RWTH Aachen. In S. Mandel, M. Rutishauser, & E. Seiler Schiedt (Eds.), *Medien in der Wissenschaft: Vol. 55, Digitale Medien für Lehre und Forschung* (pp. 303–316). Waxmann.
- Leitner, P., Ebner, M., Ammenwerth, E., Andergassen, M., Csanyi, G., Gröbinger, O., Kopp, M., Reichl, F., Schmid, M., Steinbacher, P., & others (2019). Learning Analytics: Einsatz an österreichischen Hochschulen.
- Lewis, C. (1982). *Using the "thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.
- Lewis, J. R. (1992). Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. In *Proceedings of the Human Factors Society Annual Meeting* (Vol. 36, pp. 1259–1260). Sage Publications.
- Lewis, J. R. (2013). Critical Review of 'The Usability Metric for User Experience'. *Interacting with Computers*, 25(4), 320–324.
- Lewis, J. R. (2018). The system usability scale: past, present, and future. *International Journal of Human-Computer Interaction*, 34(7), 577–590.
- Lewis, J. R., Utesch, B. S., & Maher, D. E. (2013). UMUX-LITE: when there's no time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
- Lightstone, S. S., Teorey, T. J., & Nadeau, T. (2010). *Physical Database Design: the database professional's guide to exploiting indexes, views, storage, and more*. Morgan Kaufmann.
- Lingel Learning. (n.d.). *Moodle Reporting*. Retrieved May 21, 2021, from <https://lingellearning.com/moodle/moodle-reporting/>
- Liu, D. Y. T., Richards, D., Dawson, P., Froissard, J.-C., & Atif, A. (2016). Knowledge Acquisition for Learning Analytics: Comparing Teacher-Derived, Algorithm-Derived, and Hybrid Models in the Moodle Engagement Analytics Plugin. In H. Ohwada & K. Yoshida (Eds.), *Knowledge Management and Acquisition for Intelligent Systems* (pp. 183–197). Springer International Publishing.
- Liu, D. Y.-T., Atif, A., Froissard, J.-C., & Richards, D. (2015). An enhanced learning analytics plugin for Moodle: student engagement and personalised intervention. *ASCILITE 2015 - Australasian Society for Computers in Learning and Tertiary Education, Conference Proceedings*.

- Łodzikowski, K. (2021). Association between allophonic transcription tool use and phonological awareness level.
- Lonn, S., & Teasley, S. D. (2014). Student Explorer: A Tool for Supporting Academic Advising at Scale. In *Proceedings of the first ACM conference on Learning@ scale conference*.
- López-Zambrano, J., Martínez, J. A., Rojas, J., & Romero, C. (2018). A tool for preprocessing moodle data sets. In *Proceedings of the 11th International Conference on Educational Data Mining, Buffalo, NY, USA*.
- Lorenzen, S., Pham, N., & Alstrup, S. (2017). On predicting student performance using low-rank matrix factorization techniques. In *European Conference on e-Learning* (pp. 326–334). Academic Conferences International Limited.
- Lukarov, V. (2019). *Scaling up learning analytics in blended learning scenarios. Dissertation*. RWTH Aachen University.
- Luna, J. M. i., Castro, C., & Romero, C. (2017). MDM Tool: A Data Mining Framework Integrated Into Moodle. *Computer Applications in Engineering Education*, 25(1), 90–102.
- Maldonado-Mahauad, J., Hilliger, I., Laet, T. de, Millecamp, M., Verbert, K., Ochoa, X., & Pérez-Sanagustín, M. (2018). The LALA project: Building capacity to use learning analytics to improve higher education in Latin America. In *companion proceedings of the 8th international learning analytics & knowledge conference*.
- Mandinach, E. B. (2012). A perfect time for data use: Using data-driven decision making to inform practice. *Educational Psychologist*, 47(2), 71–85.
- Manzo, M. (2017). A model for users behavior analysis and forecasting in Moodle. *Journal of E-Learning and Knowledge Society*, 13(2).
- Markl, V. (2014). Breaking the chains: On declarative data analysis and data independence in the big data era. *Proceedings of the VLDB Endowment*, 7(13), 1730–1733.
- Mazza, R., Bettoni, M., Faré, M., & Mazzola, L. (2012). MOCLog - Monitoring Online Courses with log data. *Proceedings of the 1st Moodle Research Conference, 2012*.
- Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies*, 65(2), 125–139.
- Mazza, R., & Milani, C. (2004). GISMO: a graphical interactive student monitoring tool for course management systems.
- McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. *Harvard Business Review*, 90(10), 60–68.
- McGraw Hill. (2021). *ALEKS PPL. Data and Analytics*. Retrieved May 12, 2021, from <https://www.mheducation.com/highered/aleksppl/data-analytics.html>
- McKay, T., Miller, K., & Tritz, J. (2012). What to do with actionable intelligence: E2Coach as an intervention engine. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.
- Medved, J.P. (2015). *LMS Industry User Research Report*. Retrieved April 13, 2021, from <http://web.archive.org/web/20170919154750/http://www.capterra.com/learning-management-system-software/user-research>
- Monllaó, D., & Dalton, E. (2017). *Moodle plugins directory: Inspire*. Retrieved May 21, 2021, from https://moodle.org/plugins/tool_inspire

- Moodle Users Association. (2020). *Members*. Retrieved April 14, 2021, from <http://web.archive.org/web/20200803210323/https://moodleassociation.org/local/ma/>
- Moodle.org. (2021a). *Moodle plugins directory*. Retrieved April 14, 2021, from <https://moodle.org/plugins/>
- Moodle.org. (2021b). *Statistics*. Retrieved April 14, 2021, from <https://stats.moodle.org/>
- MoodleDocs. (2013). *About Moodle*. Retrieved April 13, 2021, from https://docs.moodle.org/24/en/About_Moodle
- MoodleDocs. (2020a). *Navigation API*. Retrieved April 22, 2021, from https://docs.moodle.org/dev/Navigation_API#Extending_the_navigation
- MoodleDocs. (2020b). *Plugin types*. Retrieved April 14, 2021, from https://docs.moodle.org/dev/Plugin_types
- MoodleDocs. (2021a). *Coding style*. Retrieved April 14, 2021, from https://docs.moodle.org/dev/Coding_style
- MoodleDocs. (2021b). *Core APIs*. Retrieved April 14, 2021, from https://docs.moodle.org/dev/Core_APIs
- MoodleDocs. (2021c). *Developer Documentation*. Retrieved April 14, 2021, from https://docs.moodle.org/dev/Main_Page
- Moreno-Marcos, P. M., Pong, T.-C., Munoz-Merino, P. J., & Kloos, C. D. (2020). Analysis of the factors influencing learners' performance prediction with learning analytics. *IEEE Access*, 8, 5264–5282.
- Muslim, A. (2018). *OpenLAP: A User-Centered Open Learning Analytics Platform*. Dissertation. RWTH Aachen University.
- Muslim, A., Chatti, M. A., Mahapatra, T., & Schroeder, U. (2016). A rule-based indicator definition tool for personalized learning analytics. In *Proceedings of the sixth international conference on learning analytics & knowledge*.
- Nacken, H. (2015). Student-centred Learning: Chancen und Herausforderungen. In *Gutes Lernen und gute Lehre: Welchen Beitrag leistet die Qualitätssicherung? Beiträge zur* (pp. 15–31). facultas.
- Nagi, K., & Suesawaluk, P. (2008). Research analysis of Moodle reports to gauge the level of interactivity in elearning courses at Assumption University, Thailand. In *2008 International Conference on Computer and Communication Engineering* (pp. 772–776). IEEE.
- Nielsen, F. Å. (2019). Danish in Wikidata lexemes. In *Proceedings of the Tenth Global Wordnet Conference*.
- Nielsen, J. (1994). *10 Usability Heuristics for User Interface Design*. Retrieved April 19, 2021, from <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '90)*, Seattle, WA.
- Nilakusmawati, D. P.E., Suprapti, N. W.S., Darmawan, I., & Raharja, M. A. (2021). Analysis of student interaction with learning objects on blended learning course applying cooperative learning together method on Moodle learning management system. In *Journal of Physics: Conference Series* (Vol. 1722, p. 12107). IOP Publishing.

- Nouri, J., Ebner, M., Ifenthaler, D., Sqr, M., Malmberg, J., Khalil, M., Bruun, J., Viberg, O., González, M. Á. C., Papamitsiou, Z., & Berthelsen, D. U. (2019). Efforts in Europe for Data-Driven Improvement of Education-A review of learning analytics research in six countries.
- O'Brien Liam, Merson Paulo, & Bass Len (2007). Quality Attributes for Service-Oriented Architectures. In *International Workshop on Systems Development in SOA Environments (SDSOA'07: ICSE Workshops 2007)*.
- Olivé, D. M., Du Huynh, Q., Reynolds, M., Dougiamas, M., & Wiese, D. (2018). A Supervised Learning Framework for Learning Management Systems. In *DATA '18, Proceedings of the First International Conference on Data Science, E-Learning and Information Systems*. Association for Computing Machinery. <https://doi.org/10.1145/3279996.3280014>
- Pardo, A., & Siemens, G. (2014). Ethical and privacy principles for learning analytics. *British Journal of Educational Technology*, 45(3), 438–450. <https://doi.org/10.1111/bjet.12152>
- Parkins, D. (2017). *The world's most valuable resource is no longer oil, but data*. The Economist. Retrieved April 15, 2021, from <https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>
- Paul, P., & Sharon, S. (2014). Student Data Privacy and Institutional Accountability in an Age of Surveillance. In M. E. Menon, D. G. Terkla, & P. Gibbs (Eds.), *Using Data to Improve Higher Education* (pp. 195–214). Brill.
- Pelletier, K., Brown, M., Brooks, D. C., McCormack, M., Reeves, J., Arbino, N., Bozkurt, A., Crawford, S., Czerniewicz, L., Gibson, R., Linder, K., Mason, J., & Mondelli, V. (2021). *2021 EDUCAUSE Horizon Report: Teaching and Learning Edition*. Educause.
- Perera, C., Ranjan, R., Wang, L., Khan, S. U., & Zomaya, A. Y. (2015). Big data privacy in the internet of things era. *IT Professional*, 17(3), 32–39.
- Pérez-Sanagustín, M., Hilliger, I., Maldonado-Mahauad, J., Pérez-Álvarez, R., Ramirez, L. N., Muñoz-Merino, P. J., Tsai, Y.-S., Ortiz-Rojas, M., Broos, T., Zuñiga-Prieto, M. Á., & others (2019). Building capacity to use learning analytics to improve higher education in Latin America: LALA Framework.
- Persike, M. (2020). Blended Learning als Spielfeld für Learning Analytics und Educational Data Mining. In M. Friedrichsen & W. Wersig (Eds.), *Digitale Kompetenz: Herausforderungen für Wissenschaft, Wirtschaft, Gesellschaft und Politik* (pp. 111–128). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-22109-6_12
- Petropoulou, O., Kasimatis, K., Dimopoulos, I., & Retalis, S. (2014). LAe-R: A new learning analytics tool in moodle for assessing students' performance. *Bulletin of the Technical Committee on Learning Technology*, 16, 2–5.
- Piety, P. J. (2015). *Assessing the educational data movement*. Teachers College Press.
- Plotly Technologies Inc. (2015). *Collaborative data science*. <https://plot.ly>
- Polo, M., Iacono, U. D., Fiorentino, G., & Pierri, A. (2019). A social network analysis approach to a digital interactive storytelling in mathematics. *Journal of E-Learning and Knowledge Society*, 15(3), 239–250.
- Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big Data*, 1(1), 51–59.

- Rabbany, R., Takaffoli, M., & Zaiane, O. R. (2011). Analyzing Participation of Students in Online Courses Using Social Network Analysis Techniques. In *4th International Conference on Educational Data Mining (EDM'11)*.
- Reese, M., & Levy, R. (2009). Assessing the future: E-portfolio trends, uses, and options in higher education.
- Retalis, S., Papasalouros, A., Psaromiligkos, Y., Siscos, S., & Kargidis, T. (2006). Towards networked learning analytics—A concept and a tool. In *Proceedings of the fifth international conference on networked learning*.
- Richards, N. M., & King, J. H. (2014). Big data ethics. *Wake Forest L. Rev.*, 49, 393.
- Riedy, M. K., & Hanus, B. (2016). Yes, Your Personal Data Is at Risk: Get over It. *SMU Sci. & Tech. L. Rev.*, 19, 3.
- Romero, C., & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1), 12–27.
- Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., Pijeira, D. H. J., Santofimia, R. J., & Kloos, C. D. (2017). Evaluation of a Learning Analytics Application for Open edX Platform. *Computer Science and Information Systems*, 14(1), 51–73.
- Rummel, B. (2015). *System Usability Scale – jetzt auch auf Deutsch*. Retrieved March 19, 2021, from <https://experience.sap.com/skillup/system-usability-scale-jetzt-auch-auf-deutsch/>
- RWTH. (2020). *Datenblatt Studierende Wintersemester 19/20*. RWTH Aachen University - Dezernat 6.0 – Planung, Entwicklung und Controlling. Retrieved October 23, 2020, from <https://www.rwth-aachen.de/cms/root/Die-RWTH/Profil/~enw/Daten-Fakten/>
- Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)* (pp. 42–47). IEEE.
- Santos, J. L., Verbert, K., & Duval, E. (2012). Empowering students to reflect on their activity with StepUp!: Two case studies with engineering students. In *Proceedings of ARETL'12 2nd workshop on Awareness and Reflection*.
- Santos, J. L., Verbert, K., Govaerts, S., & Duval, E. (2013). Addressing learner issues with StepUp! an evaluation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*.
- Santoso, H. B., Batuparan, A. K., Isal, R. Y. K., & Goodridge, W. H. (2018). The development of a learning dashboard for lecturers: A case study on a student-centered e-learning environment. *The Journal of Educators Online*, 1.
- Santoso, H. B., Schrepp, M., Isal, R., Utomo, A. Y., & Priyogi, B. (2016). Measuring user experience of the student-centered e-learning environment. *Journal of Educators Online*, 13(1), 58–79.
- SAP. (2020). *Fiori Design Guidelines: Micro Chart*. Retrieved June 14, 2021, from <https://experience.sap.com/fiori-design-web/micro-chart/>
- Saqr, M., Fors, U., & Tedre, M. (2017). How learning analytics can early predict under-achieving students in a blended medical education course. *Medical Teacher*, 39(7), 757–767.
- Sauro, J. (2015). SUPR-Q: A comprehensive measure of the quality of the website user experience. *Journal of Usability Studies*, 10(2).
- Sauro, J., & Lewis, J. R. (2016). *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann.

- Scheffel, M. (2017). *The evaluation framework for learning analytics*. Doctoral thesis. Open Universiteit Heerlen, The Netherlands.
- Scheffel, M., Drachsler, H., Toisoul, C., Ternier, S., & Specht, M. (2017). The proof of the pudding: examining validity and reliability of the evaluation framework for learning analytics. In *European conference on technology enhanced learning* (pp. 194–208). Springer.
- Schmidt, J.-H., & Weichert, T. (2012). *Datenschutz: Grundlagen, Entwicklungen und Kontroversen*. Bundeszentrale für politische Bildung.
- Schmitz, H.-C., Wolpers, M., Kirschenmann, U., & Niemann, K. (2011). Contextualized attention metadata. *Human Attention in Digital Environments*, 186–209.
- Schön, D. A. (1983). *The Reflective Practitioner: How professionals think in action*. Temple Smith.
- Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *Ijimai*, 4(6), 103–108.
- Schroeder, U., Rohde, P., & Gebhardt, M. (2008). E-Strategy der RWTH Aachen: nahtlos durchgängige E-Learning-Prozesse. In J. Stratmann & M. Kerres (Eds.), *Medien in der Wissenschaft: Vol. 46. E-Strategy: Strategisches Informationsmanagement für Forschung und Lehre* (pp. 191–209). Waxmann.
- Schumacher, C., & Ifenthaler, D. (2017). Features Students Really Expect from Learning Analytics. *Computers in Human Behavior*. Advance online publication. <https://doi.org/10.1016/j.chb.2017.06.030>
- Schwendimann, B. A., Rodríguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., Gillet, D., & Dillenbourg, P. (2016). Understanding learning at a glance: An overview of learning dashboard studies. In *Proceedings of the sixth international conference on learning analytics & knowledge*.
- Slater, N. (2017). *Learning Analytics Explained*. Taylor & Francis.
- Slater, N., Berg, A., & Webb, M. (2015). Developing an open architecture for learning analytics. *EUNIS Journal of Higher Education*.
- Shabani, M., & Borry, P. (2018). Rules for processing genetic data for research purposes in view of the new EU General Data Protection Regulation. *European Journal of Human Genetics*, 26(2), 149–156.
- Shao, G., Shin, S.-J., & Jain, S. (2014). Data analytics using simulation for smart manufacturing. In *Proceedings of the Winter Simulation Conference 2014* (pp. 2192–2203). IEEE.
- SHEILA. (n.d.). *About - SHEILA*. Retrieved April 9, 2021, from <https://sheilaproject.eu/about-sheila/>
- Shum, S. B., & Ferguson, R. (2012). Social learning analytics. *Journal of Educational Technology & Society*, 15(3), 3–26.
- Siemens, G. (2010). *Call for Papers of the 1st International Conference on Learning Analytics & Knowledge (LAK 2011)*. Retrieved March 16, 2021, from <https://tekri.athabascau.ca/analytics/>
- Siemens, G. (2013). Learning Analytics: The Emergence of a Discipline. *American Behavioral Scientist*, 57(10), 1380–1400. <https://doi.org/10.1177/0002764213498851>
- Siemens, G., & Baker, R. S. d. (2012). Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*.

- Siemens, G., Gasevic, D., Haythornthwaite, C., Dawson, S., Shum, S. B., Ferguson, R., Duval, E., Verbert, K., Baker, R., & others. (2011). *Open Learning Analytics: an integrated & modularized platform: Proposal to design, implement and evaluate an open platform to integrate heterogeneous learning analytics techniques*. Society for Learning Analytics Research (SoLAR). Retrieved June 21, 2021, from <https://solaresearch.org/wp-content/uploads/2011/12/OpenLearningAnalytics.pdf>
- Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30.
- Simpson, J. (1989). *Oxford Dictionary of English, 2nd ed.*
- Singell, L. D., & Waddell, G. R. (2010). Modeling retention at a large public university: Can at-risk students be identified early enough to treat? *Research in Higher Education*, 51(6), 546–572.
- Slade, S., & Prinsloo, P. (2013). Learning Analytics: Ethical Issues and Dilemmas. *American Behavioral Scientist*, 57(10), 1510–1529. <https://doi.org/10.1177/0002764213479366>
- Slade, S., & Tait, A. (2019). *Global guidelines: Ethics in learning analytics*.
- Snell, J. M., Atkins, M., Norris, W., Messina, C., Wilkinson, M., & Dolin, R. (2011). *JSON Activity Streams 1.0*. Retrieved March 24, 2021, from <https://activitystrea.ms/specs/json/1.0/>
- SoLAR. (n.d.–a). *Special Interest Groups (SIGs) - Learning Analytics Community Europe (LACE)*. Retrieved April 9, 2021, from <https://www.solaresearch.org/community/sigs/lace-sig/>
- SoLAR. (n.d.–b). *Special Interest Groups (SIGs) - Learning Analytics for Latin America (LALA) SIG*. Retrieved April 10, 2021, from <https://www.solaresearch.org/community/sigs/lala-sig/>
- SoLAR. (2021). *About SoLAR - Society for Learning Analytics Research (SoLAR)*. Retrieved April 9, 2021, from <https://www.solaresearch.org/about/>
- Spikol, D., Ruffaldi, E., Dabisias, G., & Cukurova, M. (2018). Supervised machine learning in multimodal learning analytics for estimating success in project-based learning. *Journal of Computer Assisted Learning*, 34(4), 366–377.
- Steitz, K. (2020). *Moodle-Instanzen an deutschsprachigen Hochschulen*. Retrieved September 7, 2020, from <https://blog.e-learning.tu-darmstadt.de/2010/04/09/moodle-instanzen-an-deutschsprachigen-hochschulen/>
- Stewart, C. (2017). Learning Analytics: Shifting from theory to practice. *Journal on Empowering Teaching Excellence*, 1(1), 10.
- Student (1908). The probable error of a mean. *Biometrika*, 1–25.
- Tabuenca, B., Kalz, M., Drachsler, H., & Specht, M. (2015). Time will tell: The role of mobile learning analytics in self-regulated learning. *Computers & Education*, 89, 53–74.
- Technische Hochschule Lübeck. (2020). *Die TH Lübeck vergibt erste digitale, rechtssichere Zertifikate im Internet - erste Blockchain-Lösung ist produktreif*. Retrieved April 14, 2021, from <https://www.th-luebeck.de/informationen-zum-coronavirus/corona-news/2020-04-22-die-th-luebeck-vergibt-erste-digitale-rechtssichere-zertifikate-im-internet-erste-blockchain-loesung-ist-produktreif/>
- Tinto, V. (2012). Enhancing student success: Taking the classroom success seriously. *Student Success*, 3(1), 1.

- Tlili, A., Essalmi, F., Ayed, L. J. B., Jemni, M., & others (2017). A smart educational game to model personality using learning analytics. In *2017 IEEE 17th International conference on advanced learning technologies (ICALT)* (pp. 131–135). IEEE.
- Tsai, Y.-S., Moreno-Marcos, P. M., Tammets, K., Kollom, K., & Gašević, D. (2018). SHEILA policy framework: Informing institutional strategies and policy processes of learning analytics. In *Proceedings of the 8th international conference on learning analytics and knowledge*.
- Tyler, M. E., & Ledford, J. L. (2006). *Google analytics*. John Wiley & Sons.
- Venkatachalapathy, K., Vijayalakshmi, V., & Ohmprakash, V. (2017). Educational data mining tools: a survey from 2001 to 2016. In *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*.
- Verbert, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, *57*(10), 1500–1509.
- Verbert, K., Govaerts, S., Duval, E., Santos, J. L., van Assche, F., Parra, G., & Klerkx, J. (2014). Learning dashboards: an overview and future research opportunities. *Personal and Ubiquitous Computing*, *18*(6), 1499–1514.
- Verykios, V. S., Gkontzis, A., & Stavropoulos, E. (2017). Assessing Students and Tutors with Learning Analytics Dashboards. *EADTU 2017*.
- WAA Standards Committee. (2008). *Web analytics definitions*. Web Analytics Association.
- Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 105–140). John Wiley & Sons, Inc.
- Williamson, B. (2017). *Big data in education. The digital future of learning, policy and practice*. Sage.
- Yujian, L., & Bo, L. (2007). A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1091–1095.
- Zafra, A., Romero, C., & Ventura, S. (2013). DRAL: a tool for discovering relevant e-activities for learners. *Knowledge and Information Systems*, *36*(1), 211–250.
- Zeckoski, A. (2015). *Apereo Learning Analytics Processor*. Retrieved May 11, 2021, from <https://confluence.sakaiproject.org/display/LAI/Apereo+Learning+Analytics+Processor>