



Department of Informatics  
Technical University of Munich



TECHNICAL UNIVERSITY OF MUNICH  
DEPARTMENT OF INFORMATICS

BACHELOR 'S THESIS IN INFORMATICS

**Targeted Analysis and Visualization of eLearning Meta Data**

Sarah Jost



TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Bachelor's Thesis in Informatics

**Targeted Analysis and Visualization of  
eLearning Meta Data**

**Zielgerichtete Analyse und Visualisierung von  
eLearning Metadaten**

Author:	Sarah Jost
Supervisor:	Prof. Dr.-Ing. Georg Carle
Advisor:	Dr. Marc-Oliver Pahl Stefan Liebald, M. Sc.
Date:	November 15, 2018



I confirm that this Bachelor's Thesis is my own work and I have documented all sources and material used.

---

Garching, November 15, 2018

Location, Date

---

Signature



## **Abstract**

The goal of this thesis is to support the teachers of online learning platforms to improve their content. The platform is called labsystem and is part of the iLab/iLab2 courses which are offered by the Technical University Munich.

We achieve this by giving the teachers statistical feedback of the learners meta data. The visualizations display irregularities and difficulties to the teachers who can then adjust the content. To accelerate the process of creating new visualizations in the future, we developed an architecture. We designed two types of features for the environment: A performance feature that visualizes the achieved credits and a engagement feature which tracks how much time the students spent on each question.



# Content

Introduction.....	1
1.1 Goals of this thesis.....	1
1.2 Outline.....	2
1.3 Methodology.....	3
Analysis.....	5
2.1. Introduction.....	5
2.2. iLab and iLab2.....	5
2.2.1 Roles.....	6
2.2.2 Structure of a Lab.....	6
2.2.3 Focus on the Teachers.....	7
2.2.4 Existing Features.....	8
2.2.5 Summary of Features.....	14
2.3 Learning Analytics.....	15
2.3.1 Online Learning.....	15
2.3.2 Definition of Learning Analytics.....	15
2.3.3 Academic Analytics, Educational Data Mining and Learning Analytics.....	16
2.3.4 Four Dimensions of Learning Analytics.....	16
2.3.5 Data Privacy.....	18
2.4 Visualization.....	19
2.4.1 Chart Types.....	19
2.4.2 Criteria of Visualizations.....	23
2.5 Goals of the thesis.....	24
2.6 Conclusion.....	25
Related Work.....	27
3.1 Coursera.....	27
3.2 edx.....	30
3.3 Open University UK.....	34
3.4 Discussion.....	35
3.4.1 Enrollment.....	36
3.4.2 Engagement.....	36
3.4.3 Performance.....	36
3.4.4 Conclusion.....	36
Design.....	39
4.1 Applying Learning Analytics.....	39
4.2 Achieving the goals.....	39

4.2.1 Architecture.....	40
4.2.2 Credit Distribution Analytics.....	42
4.2.3 Time Tracking Algorithm.....	45
Implementation.....	49
5.1 Settings.....	49
5.1.1 Overview.....	49
5.1.2 Lab.....	50
5.1.3 Part of the Lab.....	50
5.1.4 Data.....	50
5.1.5 Type of Representation.....	50
5.2 Methods of Calculation.....	51
5.3 Data Types.....	51
5.3.1 Database to Calculation.....	51
5.3.2 Calculation to Drawing.....	52
Evaluation.....	55
Conclusion.....	57
7.1 Future Work.....	57
Bibliography.....	59

# List of Figures

2.1 Multiple Choice Questions.....	6
2.2 Free Input Questions.....	7
2.3 Credits Overview for all Students.....	8
2.4 Time Tracking Algorithm.....	11
2.5 Time Tracking Visualization.....	13
2.6 Process of Learning Analytics.....	15
2.7 Coursera Enrollment.....	18
2.8 Circle Chart.....	20
2.9 Coursera Origin.....	20
2.10 Edx: Video Views.....	21
2.11 Error Bar Charts.....	22
2.12 Box Plots.....	22
3.1 Coursera Country.....	28
3.2 Coursera Rentenion.....	29
3.3 Coursera Grade Distribution.....	29
3.4 Coursera A/B Testing.....	30
3.5 edx Education.....	31
3.6 edx Engagement.....	32
3.7 edx Video Views.....	32
3.8 edx Performance distribution.....	33
3.9 edx Answer Variation.....	33
3.10 OUK Predictions.....	34
3.11 OUK Engagement.....	35
4.1 Overview Architecture.....	42
4.2 Relative Credits - Practical Part.....	45
4.3 Absolute Credits - Practical Part.....	45
5.1 Data Types.....	51
5.2 Two Dimensional Array.....	52



## List of Tables

2.1: Visualizations Overview.....	24
3.1 Coursera Overview .....	28
3.2 edx Overview.....	31
3.3 Open University UK Overview.....	34
3.4 Overview.....	37



# Chapter 1

## Introduction

### 1.1 Goals of this thesis

The objective of this thesis is to support the teachers of an online learning platform to improve the content. This platform is called ‘labsystem’ and is used for the iLab/iLab2 courses offered by the Technical University Munich.

The quality of the content is directly linked to the learners success. This is why we want to help the teachers to improve the content by analyzing and visualizing user meta data. This statistical feedback can be used to detect problems and difficulties that students may have. The teachers can then adjust the content appropriately. Instead of just helping one particular student, this helps all future students and is therefore much more effective.

At the moment, there are two features for this purpose: The first one visualizes the performance of the students. This can be used to detect unclear or too easy/difficult questions. The second feature tracks the time the students spent on each question. The teacher can use this to see irregularities.

Since there is no environment to easily integrate new features at the moment, we want to create one: This feature takes data coming from a new feature and visualize the data automatically. This supports the achievement of our goal since it allows the easy integration of many new features to the labsystem that accomplish our goal.

The two existing features have inconveniences and we therefore want to improve them and add the new versions directly to the new environment.

## 1.2 Outline

In this section we explain the structure of this thesis:

We start Chapter 2 – the Analysis – by explaining the structure and roles of the iLab/iLab2 courses and the corresponding online learning platform called ‘labsystem’. Then we present the existing features of user feedback and visualizations of user meta data. We discuss the features of the latter one – credit distribution analytics and the time tracking algorithm – in detail.

Then we present the background of Learning Analytics. This is the science of (amongst other things) analyzing data about learners in order to optimize learning. We introduce the four dimensions of Learning Analytics. These are used to define the goals and parameters.

Afterwards we present different types of charts and discuss their uses, advantages and inconveniences.

At the end of the Analysis, we state our specific goals and requirements.

In Related Works – which is Chapter 3 – we present the visualizations three other online learning platforms: Coursera, edx and Open University UK. We discuss the used data and the visualization. Additionally we explain the uses for the teachers.

In Chapter 4 – the Design – we discuss the design of our artifacts. We present their structure and features, We do this by arguing why we chose this exact design and comparing different solutions.

In the Implementation, which is Chapter 5, we present details of the implementation. We don’t cover the whole code but rather explain some details and specific solutions for problems.

Chapter 6 evaluates – as the name implies – our artifacts. In Chapter 7, we conclude the evaluation and present other possible features based on this thesis.

## 1.3 Methodology

The Analysis presents needed background in order to achieve our goal: We introduce the features of the labsystem, Learning Analytics and discuss different chart types. Then we state our requirements which are based on the discussions of the analysis. In Related Work we analyze the different features using what we have learned from Learning Analytics and chart types. In Design, we use the realizations about charts from the Related Work and background knowledge about the dimensions of Learning Analytics and chart types from the Analysis to fulfill our requirement. In the Implementation, we explain specific parts of our designs. Then we evaluate how our artifacts meet the set requirements and find a conclusion. At the end, we present possible future works based on the findings and artifacts of this thesis.



# Chapter 2

## Analysis

### 2.1. Introduction

In this chapter we introduce and discuss the needed background to achieve our goal. We cover three topics in this chapter: the iLab/iLab2 courses, learning analytics and visualizations.

The analysis of the iLab/iLab2 courses introduces the course explains the structure of the course (especially its online learning system called ‘labsystem’) and reviews the existing feedback features that help improving the courses.

The second part introduces the scientific field of learning analytics.

The last part gives an overview over existing charts and argues about their advantages and inconveniences.

### 2.2. iLab and iLab2

The iLab and the iLab2 are practical courses offered by the Chair for Network Architectures and Services of the Technical University of Munich (TUM). These courses teach the bachelor and master students of computer science about computer networks using the labsystem. This system is an online learning-platform created by Marc-Oliver Pahl in 2003.

The iLab focuses on the creation of the Internet and gives the students the opportunity to build their own Internet. The iLab2 on the other hand covers many different more specific topics.

Each week there is a different exercise – which is called Lab. It consists of three parts: The Lecture (Introduction), the PreLab (individual preparation for the practical part) and the Lab (practical part done in team work). Since both the whole exercise and the practical part are called Lab, we refer to the whole exercise as ‘Lab’ and call the practical part only practical part.

In this section we present the structures and roles of the iLab/iLab2 course. Then we introduce the structural elements of the labsystem and discuss the existing feedback features of the labsystem that help the teachers to improve the iLab/iLab2 course.

### 2.2.1 Roles

Since there are multiple roles in the iLab/iLab2 course, we briefly explain their functions and tasks:

- The *Instructors* are part of the Chair for Network Architectures and Services and organize the courses. They give the lectures and
- The *Tutors* correct the practical parts and answer the questions of learners via the labsystem.
- The *Students* are the participants of the courses.

Since both the Instructors and the Tutors receive the feedback, we refer to both of them for the remainder of this thesis as ‘teachers’.

### 2.2.2 Structure of a Lab

In the labsystem there are two parts: The PreLab and the practical part. In this section we present the content of both parts.

#### a) PreLab

The PreLab is done individually and presents the subject in more detail than the lecture. It contains both text elements – explanations – and multiple choice questions (Figure 2.1). These questions are also called m-elements.

A screenshot of a web-based multiple choice question. At the top right, there is a small icon and the text "m2 (1) This is a multiple choice question. Click on 'give' | Visible in 'c\*' only. 📄 🗃 🔍". Below this, the question text reads: "1.2. This is a multiple choice question. Click on "give answers" to set the check marks. Don't forget to "save"..." followed by three options: "This answer is correct." (with a checked checkbox), "This answer is wrong." (with an unchecked checkbox), and "This answer is correct." (with a checked checkbox). At the bottom, a note says: "Here you might find some remarks why the solution is correct." A vertical green bar is on the left side of the screenshot.

Figure 2.1: Multiple Choice Question [4]

A student can answer them up to three times. Depending on how long it takes him to get the right answer, he achieves a different amount of credits. The credits of the PreLab are not taken into account for the grade.

### b) Practical Part

The practical part is done after the PreLab. The students do it in a team of two each. It consists of text elements – explanations and instructions – and questions (also called i-element). The questions of the practical part are free input questions (Figure 2.2), meaning the students write the answer as a free text input.

The screenshot shows a digital interface for a free input question. At the top, there is a status bar with icons and text indicating the question is visible only to certain users. Below this, the question text reads: "1.2. ...and here will be an interesting question to the experiment. You can answer this demo question by [Visible in 'c' only]". A yellow highlighted area contains the placeholder text "Well here will be what you should have answered...". In the bottom right corner, there is a note "[0 credits]".

Figure 2.2: Free Input Question [1]

Once the Lab is finished, they get corrected and the students receive their achieved credits. Each question has a different amount of possible credits. There are also 'bonus questions'. That means the possible amount of credits is zero and the students can get additional credits.

The labsystem saves answers of the students as well as their credits. Since the students have to open (clicking 'open' and 'close') a question – i and m elements – in order to answer it and close it afterwards, the system can also log the time of the opening and closing event for each question.

### 2.2.3 Focus on the Teachers

As discussed before, the goal of this thesis is to support the teachers to improve the iLab/iLab2 courses. We believe that showing the teachers of the iLab/iLab2 which content (questions, explanations, ...) needs to be improved, is more helpful to the students in the long term. The adapted content helps all the future students – not just individual students who have difficulties right now.

## 2.2.4 Existing Features

The labsystem already offers multiple features that help the teachers to notice difficulties or improve the Labs.

### 2.2.4.1 Credit Overview

The most obvious one is an overview over the achieved credits. (Figure 2.3)

#	team		m-credits	i-credits
1	201	D300F7NN, D300F7NN	IPv6 77% 50/75	95% 54/57.25
2	201	SUDI7EZZ, SUDI7EZZ	76% 57/75	95% 54.57/57.25
3	202	6F8ZXEE, 6F8ZXEE	55% 43/75	76% 43.25/57.25
4	202	3GLTDX00, 3GLTDX00	65% 49/75	76% 43.25/57.25
5	203	YOQLXM44, YOQLXM44	72% 54/75	90% 51.25/57.25
6	203	CLRK3CMM, CLRK3CMM	17% 13/75	90% 51.25/57.25
7	204	KT7FHXSS, KT7FHXSS	60% 45/75	86% 49.25/57.25
8	204	9WOPU3WW, 9WOPU3WW	60% 45/75	86% 49.25/57.25
9	205	RZT87U00, RZT87U00	53% 40/75	78% 44.75/57.25
10	205	JBW4CYRR, JBW4CYRR	73% 55/75	78% 44.75/57.25
11	206	WYY3KF66, WYY3KF66	51% 38/75	83% 41.25/57.25
12	206	UBE49VBB, UBE49VBB	65% 49/75	83% 47.5/57.25
13	207	J4RY5O33, J4RY5O33	81% 64/75	72% 41.25/57.25
14	207	H764TB44, H764TB44	77% 58/75	72% 41.25/57.25
15	208	E206OBTT, E206OBTT	75% 55/75	92% 52.75/57.25
16	208	6QM3ZGEE, 6QM3ZGEE	71% 53/75	92% 52.75/57.25
17	209	QRT106VV, QRT106VV	72% 54/75	84% 48/57.25
18	209	IYPAQXX, IYPAQXX	53% 40/75	84% 48/57.25
		Answers of ALL USERS	64.15% 48.11/75	83.94% 48.06/57.25

Figure 2.3: Credit-Overview for all Students [2]

This table visualizes three things for each student:

- Absolute achieved credits for each exercise of the PreLab
- Mean value of absolute achieved credits for each exercise of the PreLab for all the students
- Sum of achieved credits relatively (to the amount of possible credits) of the PreLab and practical part
- Sum of achieved credits relatively (to the amount of possible credits) of the PreLab and practical part for all the students

It is also possible to view the graphic for one student and comparing the results of all the Labs.

The sum of the absolute numbers for PreLab and practical part is included but not visualized. Additionally this is written in gray which makes it difficult to read.

This visualization doesn't visualize both relative and absolute amount of credits for the (sum of the) PreLab and the practical part. One can not see the achieved credits for the different questions in a PreLab.

Only the mean value is displayed but no other statistical values. This makes an analysis of the distribution complex.

#### 2.2.4.2 Ticket System

Another feature of the labsystem is the possibility for the students to write a ticket to the teachers while doing the PreLab or practical part. Each element in these parts has a button which makes it possible to write a ticket that includes a link to this element.

This way the teachers know exactly which question or element the students are writing about. This ticket can have different priorities, depending on whether a student needs help, gives feedback or wants to ask about the correction. Both when a student gives direct feedback in the tickets or uses them to ask questions can be used to improve the Labs in the future.

It is of course also possible for the teachers to write a ticket. This may be helpful when a teacher realizes while correcting that many students had difficulties with a question or misunderstood it.

#### 2.2.4.3 Feedback and Questionnaire

The labsystem also offers a feedback feature. This is included at the end of every PreLab and practical part and gives the students the opportunity to tell the teachers

whether they liked the Lab and give suggestions. The students are motivated to do that by giving them bonus-credits for helpful feedback.

Another source of feedback is an extra questionnaire which is linked after every lab. It offers insight in whether the students liked the Lab, how long it took them to do it and whether they found it too difficult or too easy.

#### 2.2.4.4 Time Tracking Algorithm

The labsystem logs many user meta data, such as the time of a login, logout, opening of a question, closing of a question. This data can be used by an algorithm to calculate the time the students spent on each question in correlation to the achieved credits.

This information is useful to the teachers of the iLab/iLab2 because it shows them irregularities.

This algorithm was created by Fabian Oehlmann. In this section we explain how the algorithm works in general, afterwards we give an overview over the code, discuss the used visualization and present the disadvantages of this solution.

##### a) The Time Tracking

The algorithm calculates the time spent on each question the following way: It is assumed that all the time spent in between questions is used as preparation for the next question. It is important to notice that the next questions doesn't have to be the next chronological question in the Lab. It is just the next opened question.

The algorithm looks at all the logged events and categorizes them into opening and closing events. An opening event means that a question was opened and the students are working on it. A closing event makes the algorithm stop the time tracking for the question and starts for the next one. This kind of event can be caused by multiple things: The question was closed, another question was opened, a timeout (60 minutes) or a logout happened. When the last question was closed, the algorithm stops. [3]

##### b) The Code

Since the code is quite complicated, we don't describe it in detail and rather give a broad overview over what each class does and how they are connected (Figure 2.9).

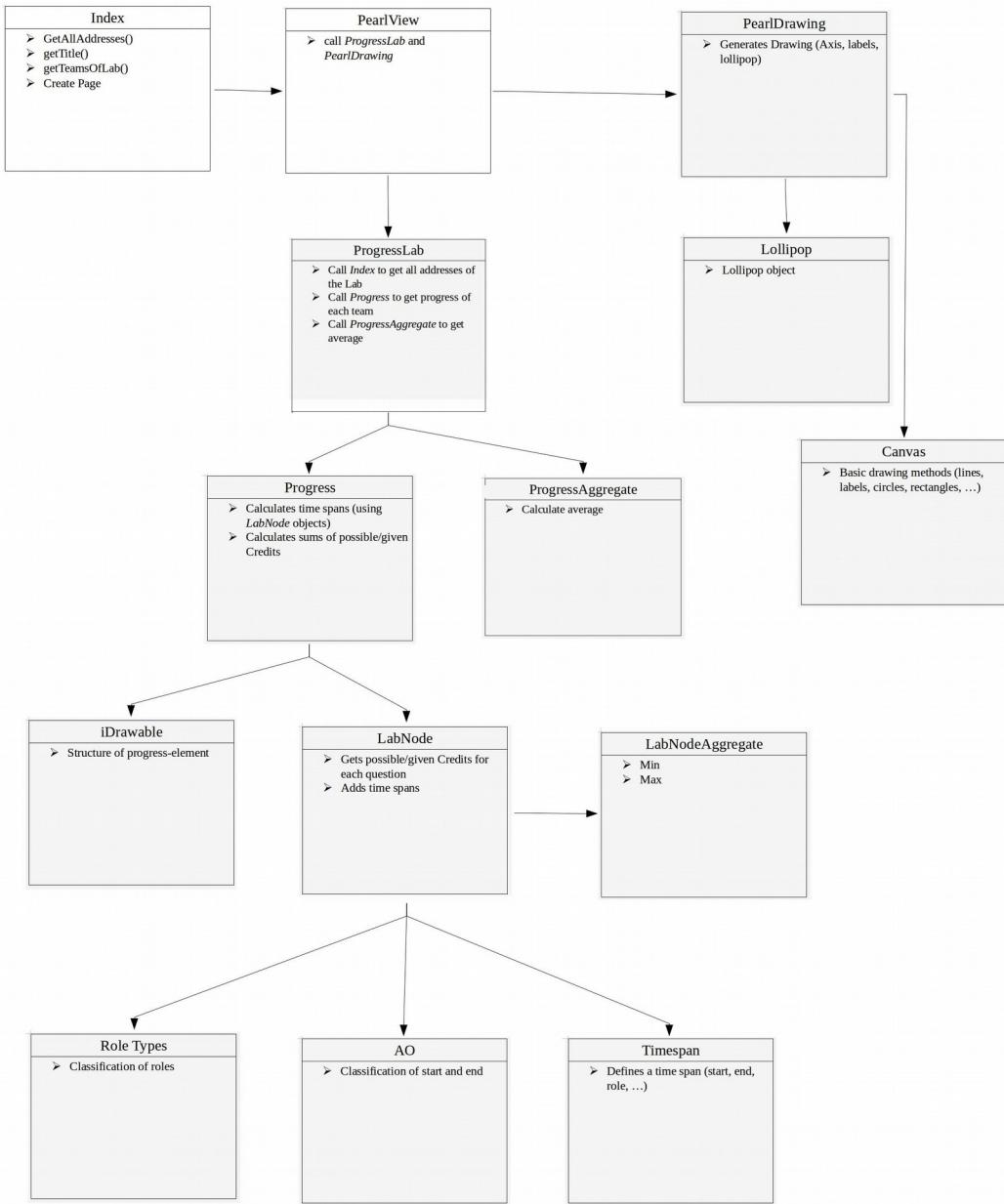


Figure 2.4: Time Tracking Algorithm

The code starts off with the class *index*. For starting the process it calls the class *pearlView*.

This class has two main purposes: Getting the progress – meaning the time tracking data – of the lab calling *ProgressLab* (or *ProgressPreLab*) and afterwards visualizing it using the class *PearlView*.

*ProgressLab* - in this case ‘Lab’ stands for ‘practical part’ - call *Progress* to get the progress of each team. Afterwards it normalizes the values using *ProgressAggregate*.

The class *Progress* contains the interesting methods of this code: It calculates the progress using the algorithm explained in the section above. *iDrawable* makes sure that the calculated progress is drawable.

*LabNode* returns the achieved and possible credits for each question to *Progress*. It uses *RoleTypes*, *AO* and *Timespan* as helper classes (they define role types, start/end and a time span object). *LabNode* uses *LabNodeAggregate* to save the minimal and maximal credits.

Now that the calculation of the progress is finished, it is returned to *PearlView*. This class passes it to *PearlDrawing* – which visualizes it. *PearlDrawing* needs two helper classes in order to do that: *Canvas* (contains all the basic drawing methods) and *Lollipop* (used as store for drawing values).

### c) The Visualization

Now we come to the visualization of the time tracking algorithm (Figure 2.5):

The x-axis represents the credit amount, while the y-axis describes the time spent on the lab.

In the chart, the left line illustrates the actually achieved credits in correlation to the time passed, while the line in the middle represents the time if the students had received all the credits. The right straight line represents a linear passing of time in correlation to the achievement of all the credits.

This visualization can be used by the teachers to see when students spent too much or not enough time on a question in correlation to the achievable credits. This becomes visible when the middle line differs too much from the line on the right. [3]

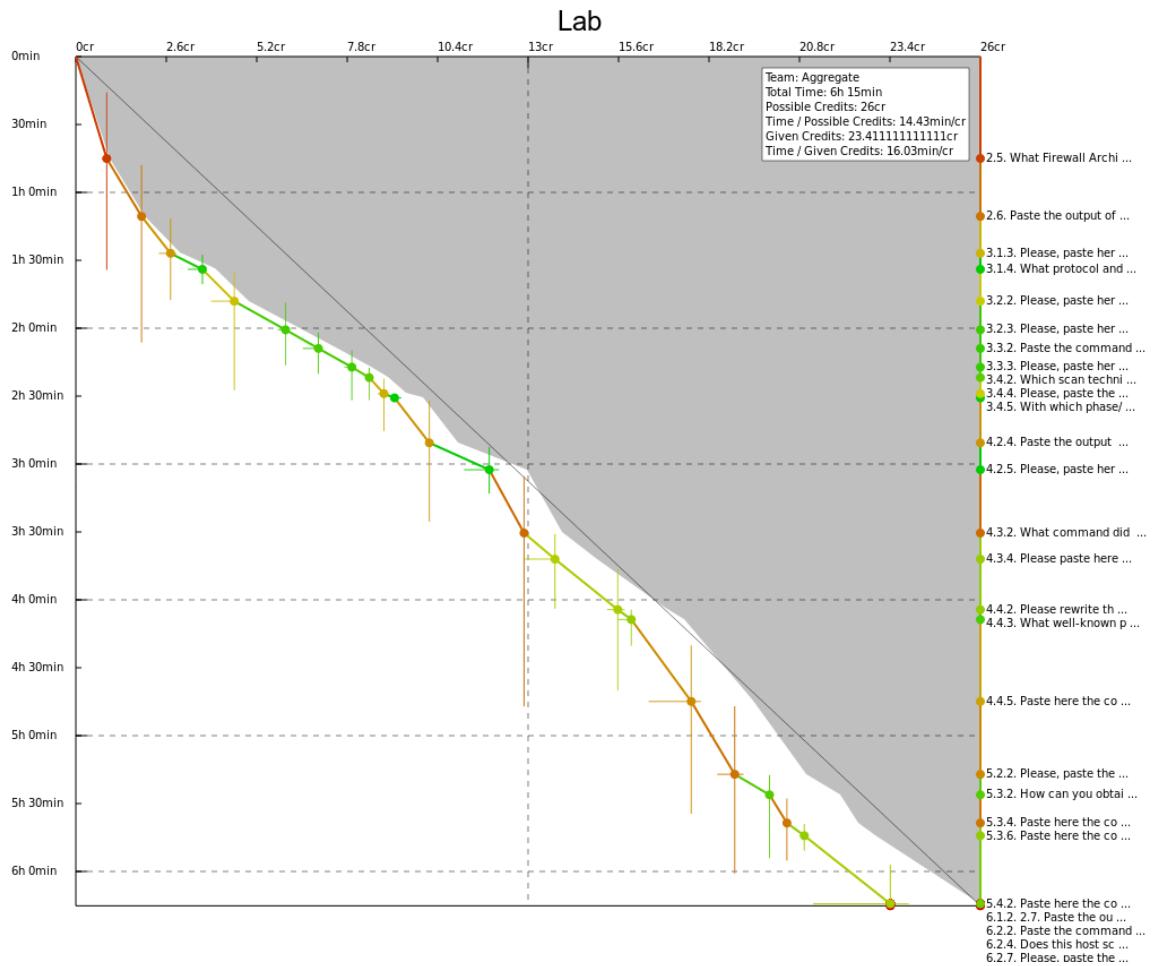


Figure 2.5 Time Tracking – Visualization [9]

#### d) Disadvantages of the Time Tracking Algorithm

There are many inconveniences to the existing algorithm, which is why we categorize them in the same order as we presented it: The algorithm, the code and the visualization.

The algorithm is not accurate. There are two main reasons for that: The timeout is set to 60 minutes, which is too much. Another reason for the inaccuracy is that it is impossible to tell whether the students spent most of the time on the question itself or on the explanation beforehand. In order for the teachers to improve the right content, it would be important to know where the problem lies exactly.

The code is difficult to understand because the structure and purposes of the classes are not obvious. The classes are interlaced which makes it difficult to improve or

exchange parts of it. It takes more than one second to execute the code (and to receive a visualization). The reason for that is because the calculation process takes very long. The visualization is not self-explaining for the targeted audience (the teachers). It is difficult to draw much meaning from the displayed graphs (the three lines). This means conclusions can only be drawn after a detailed analysis. It is difficult to compare the time intervals of the questions.

## 2.2.5 Summary of Features

There are multiple features and methods in the iLab/iLab2 courses that help the teachers to improve the quality of the Labs. In this section, we summarize them into two sections: The first section is the direct feedback, while the second one is the indirect feedback – which is already visualized:

### a) Direct feedback

Direct feedback is feedback given directly by the students.

This is the feedback given by the students on the feedback pages in a Lab or in tickets (suggestions and questions). This feedback is helpful since it already specifies the problems and difficulties that the students had, but it is also more work for the teachers to read every feedback and decide whether it is helpful or not.

### b) Indirect Feedback

Now we come to the indirect feedback. Indirect feedback is not given by the students directly but rather is retrieved from collected data.

As discussed in the last section, there is a feature in the labsystem which offers an overview over the achieved credits of a Lab. Another feature is the time tracking algorithm which displays how long the students need to answer each question.

## 2.3 Learning Analytics

### 2.3.1 Online Learning

In the last few years online learning platforms gained more and more students. After all, it has many advantages, like being able to learn everywhere in the world and when you want to. But it also has disadvantages: In normal classrooms the teachers can immediately see when their students get bored, confused or overwhelmed. Then they can change their course accordingly, e.g. by explaining something in a different way or asking more challenging question [5]. On a online learning platform, this is not possible. That's why it is important to use the user data efficiently in order to improve the learning system.

In this section, we give an overview over the science of learning analytics and see the steps of a learning analytics process. These steps help us later to determine how we want to analyze the user meta data.

### 2.3.2 Definition of Learning Analytics

There is no official definition of learning analytics, but the following definition is the most common one and was adopted by the Society for Learning Analytics Research: 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.” [5]

The process of Learning Analytics is usually a cycle of three steps (as described in [6]):

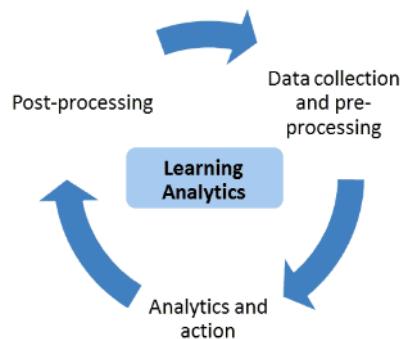


Figure 2.6: Process of Learning Analytics [6]

- (1) The first step is the collection of data and pre-processing said data. Pre-processing is the act of preparing the data for the analysis, e.g. data reduction, data transformation and user identification.
- (2) The next step is the analysis – which includes the visualization of the data – and taking action. The second part of this step can be done in many ways, like adaptation, assessment, personalization and prediction. This is the main goal of the learning analytics process.
- (3) The last step is the post-processing. It describes the adjustments of the process itself, such as adapting the analytics method, choosing new data sets or determining new indicators.

### 2.3.3 Academic Analytics, Educational Data Mining and Learning Analytics

The three emerging fields of academic analytics, learning analytics and educational data mining are very similar to each other – they all analyze educational data. In this section, we discuss the goals and differences of the fields:

Academic analytics focuses on the political and economic challenge. Its goal is to improve learning at a (inter)national level. [5]

The other two fields are more difficult to distinguish: They have overlaps – in both research and researchers. Educational data mining (EDM) focuses on automatic methods of discoveries and adjustments, while the researchers of learning analytics prefer human-led methods. In the case of adjusting the learning environment, this means that EDM adapts it automatically. LA on the other hand tries to empower the teachers and students to adapt it accordingly. Another difference between the two fields is perspective of the data. LA wants to understand the system as a whole, while EDM analyzes parts of the system and tries to find connections between these parts. [7]

### 2.3.4 Four Dimensions of Learning Analytics

There are four dimensions in learning analytics: [6]

- Data (What?)
- Stakeholders (Who?)
- Objectives (Why?)
- Methods (How?)

In this section, we describe them in more detail.

### a) Data

This dimension is about what kind of data is used in learning analytics. It can be divided into two categories: centralized educational systems and distributed learning environments. Centralized educational systems are mainly learning management systems, like Moodle. These systems log a lot of data of their students which can be used in learning analytics. Distributed learning environments on the other hand are personal learning environments, that connect many sources and formats of learning medias. [6]

For the purpose of supporting the teachers on online learning platforms to improve the quality of their content, the used data can be divided into three categories: [8]

1. Enrollment: In this category the teacher asks the question on who is using the content. This information can be useful to adapt the content of the course to its audience. It also includes information on when people joined the class.
2. Engagement: The focus of this category is how the students interact with the content. Visit counts are an example for data of this category.
3. Performance: For this category, the user performance data is collected and analyzed.

### b) Stakeholder

Stakeholders are the people the analysis is designed for. They can be students, teachers or even educational institutions [6] (like for academic analytics). Each stakeholder has their own interests which causes completely different tasks for the analysis – the objective.

### c) Objectives

The objective is the reason the analysis is done for. There are many objectives and we present some of them in this section: [6]

- Monitoring and Analysis: The goal of this objective is to analyze the student behavior in order to help the teachers to make decisions and to improve the learning environment.

- Prediction and intervention: This objective wants to predict future student behavior by analyzing the current activities of the students. Teachers can use the predictions for proactive interventions.
- Assessment and feedback: The analysis for this objective assesses the effectiveness of the learning process and gives the feedback to both students and teachers.
- Personalization and recommendation: Personalization describes the process of a system that supports the students to choose their own learning goals and helps them to achieve these. Recommender systems on the other hand present the students recommendations on what to learn next. These recommendations are chosen by comparing the activities of students with similar activities.

#### d) Methods

The last dimension are the used methods. In this section we describe three methods: [6]

- Statistics: These methods use user usage data (e.g. time online and numbers of visits) in order to calculate statistical operation like the mean value, median or standard deviation.
- Information Visualization: It is important to present the results of learning analytics in a way that is easy to understand by the stakeholders. And visualization facilitate the interpretation of data. But it is important to notice that the visualization has to be suitable for the stakeholders in terms of complexity and objective.
- Data Mining: This method tries to find patterns in the data sources. There are three main categories for data mining: Classification (supervised learning), clustering (unsupervised learning) and association rule mining. Classification is the process of finding data classes (models) in order to predict user behavior. Clustering is a method that organizes the data into groups of similarity, meaning objects in one group are similar and objects of different groups are dissimilar. Association rule mining tries to find correlations within the data. As the name implies data mining is a method usually used in educational data mining.

#### 2.3.5 Data Privacy

In order to collect data, learning analytics tracks user behavior and logs the data. And therefore it is important to be conscious of data privacy: One needs to make sure user

data is not being misused, confidential user data is being preserved, the anonymity of users is warranted [6] and users are aware of being tracked.

## 2.4 Visualization

The visualization of data is a way to make information easier understandable. In this chapter, we want to present and discuss multiple visualization types. This helps us decide how we want to visualize the analyzed user meta data.

### 2.4.1 Chart Types

#### a) Line Chart

The first type of chart we want to discuss is the line chart. It consists of a continual line in a coordinate plane.

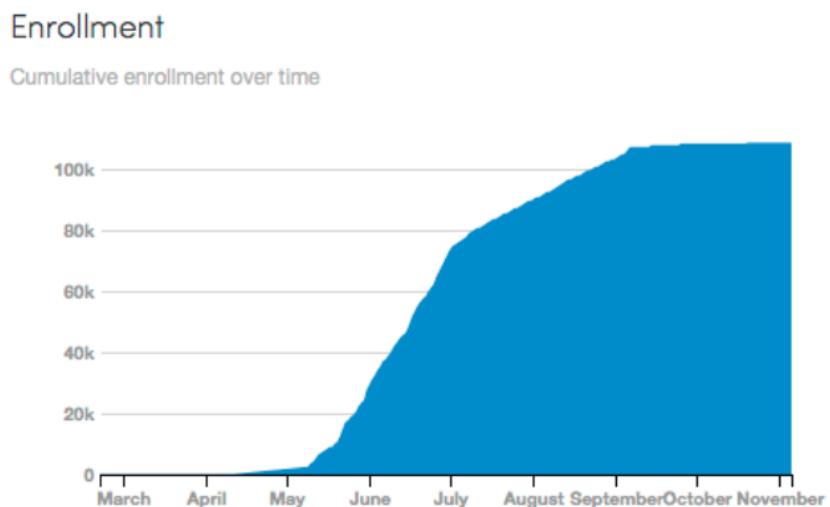


Figure 2.7: Coursera Enrollment [15]

Since it is continuous, the x-axis is often used as a time line – which makes the horizontal distance of two values meaningful. The y-axis values are easily comparable and many items (e.g. time values) can be displayed. This chart should be used to display developments over time.

### b) Circle Chart

The circle chart is a circle that is divided into multiple parts. These parts represent different items and their sizes correlate to the percentage of said items.

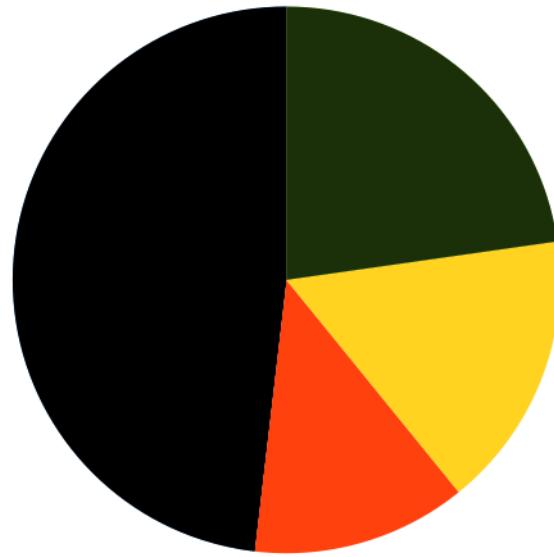


Figure 2.8: Circle Chart

These charts are useful to give a quick overview over which part is the biggest. But they make it difficult to compare the size (and therefore the percentage) of the items, especially when they have similar sizes. The chart can also appear chaotic when there are too many items. It is important to use differentiable colors in order to tell the items apart.

### c) Bar Charts

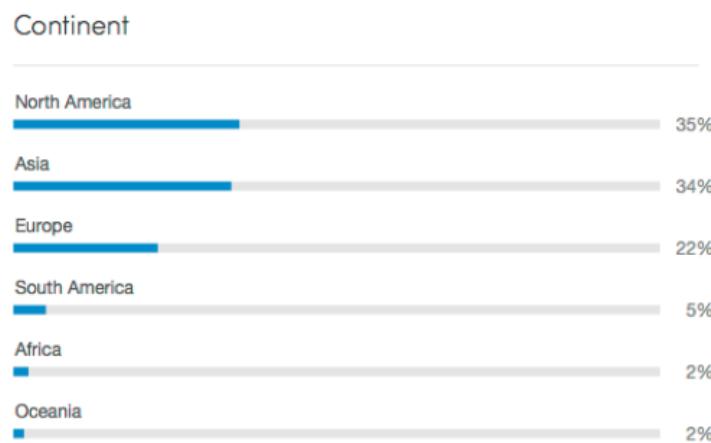


Figure 2.9: Coursera: Origin [16]

Bar charts – or the horizontal version called line charts – are the most commonly used charts. [10] They consist of multiple bars and their sizes represent the value of each item. The items are easily comparable and it is possible to add items without endangering the simplicity of the chart. [10]

A disadvantage of the chart is that it only displays one value per item.

#### d) Stacked Bar Charts

Stacked bar charts are very similar to usual bar charts but with one distinction: Each bar is divided into (at least) two parts.

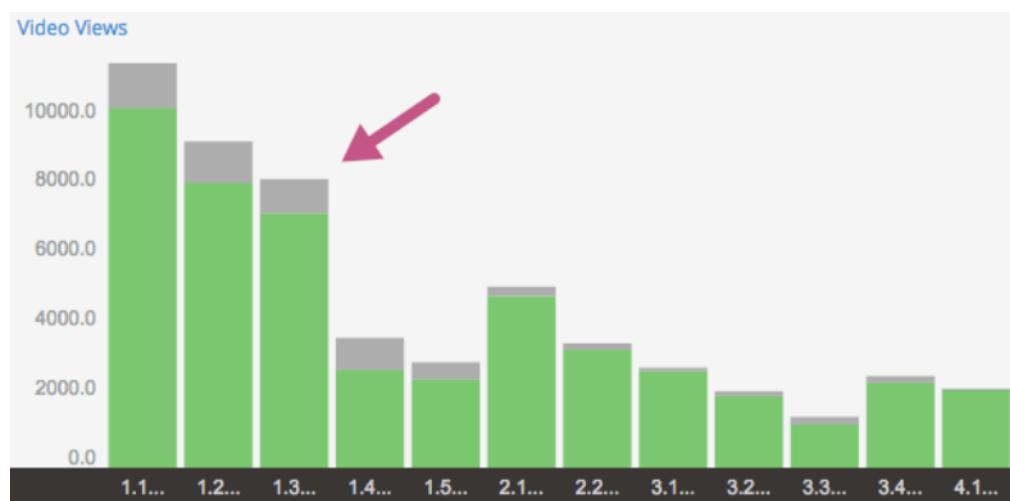


Figure 2.10: Edx: Video Views [18]

This makes it possible to present two or more statistical attributes per category (e.g. maximum value and median). Since it is possible to display many different values with this chart, it is important to explain what each part represents. It is also possible to display percentages using this type of chart. [10]

#### e) Error Bar Charts

These kind of charts are also built like the usual bar charts but with an extra feature: The bar of each item represents the median or mean value. Additionally to the bars, there is a line which signifies two values (e.g. the standard deviation of the item or the data range) [11].

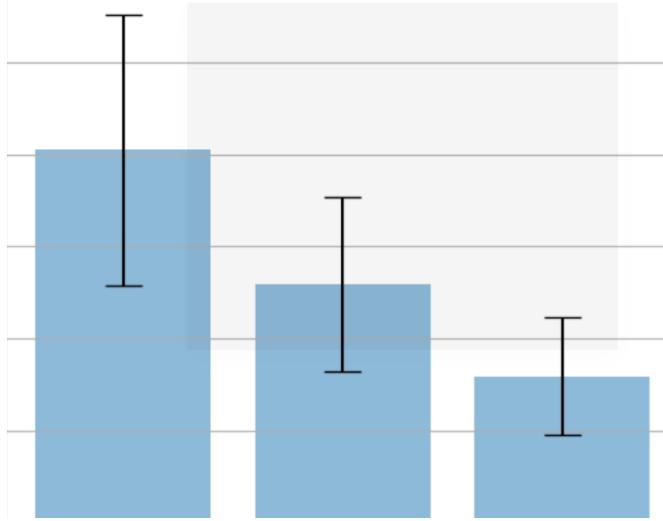


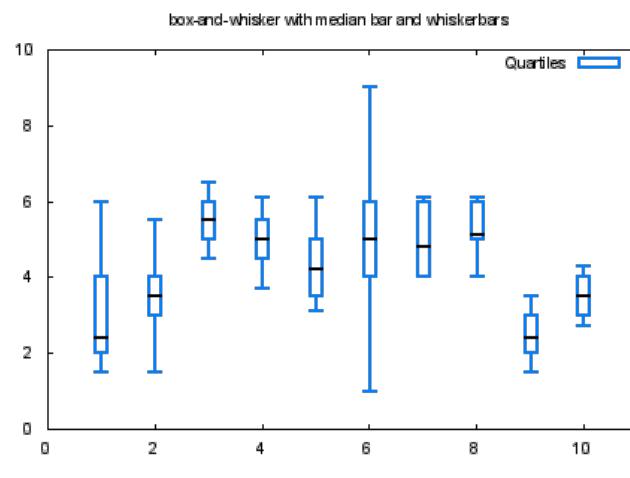
Figure 2.11: Error Bar Charts [12]

Unlike the stacked bar charts, the statistical information is ‘visually prioritized’, meaning the bar is the first thing one focuses on.

#### f) Box Plots

Box Plots are charts that make it possible to present multiple statistical values per item, such as the median, the first quartile, the third quartile, minimum and maximum.

The chart consist of a box for each category. The box represents the first quartile up to the third quartile. In the middle of the box, there is a line showing the median. This means 50% of the values are in the box. On both sides of the box there is a line. This line represents all the other values until the minimum/maximum. Sometimes there are outliers added to the chart. [13]



Figures 2.12: Box Plots [2]

An advantage of this chart are the many included statistical attributes. When looking at the chart in detail, one can understand many things.

But in order to read the chart, one has to know exactly what it means. It is also not as facile to understand and compare to bars because there is so much information included in the chart.

Since the statistical operations median, first quartile and third quartile only make sense on bigger amounts of data, this chart can only be used when there are enough values per item to be evaluated.

#### 2.4.2 Criteria of Visualizations

In this section we want to compare the presented chart types by using the following criteria:

- Information: What information is contained in the chart?
- Items: How many items can be displayed?
- Comparability: Are the values of different item (easily) comparable?
- Stakeholders: Is additional knowledge required in order to read the chart?

	Information	Items	Comparability	Stakeholders
Line Charts	Continual Information, One Value per Item	Many items can be displayed	The values of the items are easily comparable	Everyone
Circle Charts	Percentage of items	Only useful for a couple of items	The values of the items are difficult to compare	Everyone
Bar Charts	One value per item	Many items can be displayed	The values of different items are easily comparable	Everyone
Error-Bar Charts	Three values (e.g. mean, maximum, minimum) per item	(See Bar Charts)	(See Bar Charts)	Explanation of lines is required
Stacked	At least two	(See Bar Charts)	Similar colors and	Explanation of

Charts	values can be displayed		too many values make items/values difficult to compare	parts is required
Box-Plot Charts	- Min/Max value - Mean value - Median value - First/Third quartile	(See Bar Charts)	Items are comparable, but detailed analysis is required	Foreknowledge of Box-Plots is required

Table 2.1: Visualizations Overview

## 2.5 Goals of the thesis

In this thesis we want to improve the quality of the Labs in order to improve the iLab/iLab2 courses. We do this by analyzing user meta data – such as the time logging data or the credits – and visualizing them in a way that indicates problems of the Lab to the teachers.

Since there isn't a environment that makes it easy to add new features, we want to create one. We call it *architecture*, since it is the foundation for every feature. We want to make sure that it is easy to add new features. Therefore it should be modularized and very flexible. It also should have a better performance time and be self-explaining. We also want to improve two features and add them to the *architecture*: The first one in the performance visualization. Right now only some parts are visualized. The distribution of the values is also not displayed. We call this feature *Credit Distribution Analytics*.

The other feature is the *Time Tracking Algorithm*. We want to make sure that the algorithm is more accurate and the visualization easier to understand. The elements should be comparable and the distribution should be shown.

### Architecture

- (1) Modularized and self-explaining code (structure)
- (2) Good performance
- (3) Flexibility

### Credit Distribution Analytics

- (1) Setting extensions

- (2) Distribution
- (3) Comparability
- (4) Targeted Audience

### **Time Tracking Algorithm**

- (1) More specific analysis
- (2) Shorter Timeout
- (3) Setting extensions
- (4) Distribution
- (5) Comparability
- (6) Targeted Audience

## **2.6 Conclusion**

In the analysis of this thesis we discussed the three domains to our goal of improving the courses via visualizations. We started off with the iLab/iLab2 courses and introduced the structure of the courses (structural part) as well as the labsystem (technical part). We also discussed already existing features, such as the time tracking algorithm. This part introduced the domain we are working in and trying to improve.

The second section of the chapter theoretical domain of the improvement process: learning analytics. We gave an overview of the field, distinguished it from related fields and explained the four dimensions of learning analytics.

Then we presented visualization. Afterwards we compared them and saw their advantages and inconveniences.

In the last section, we stated our requirements.



# Chapter 3

## Related Work

In the last chapter we introduced and analyzed the environment and tools we need to achieve our goals. In this chapter we present the solutions of three other online learning platforms: Coursera, edx and Open University UK.

We present the visualizations of each platform in the following order: Enrollment, Engagement and Performance.

For each visualization, we can ask about the four dimensions of Learning Analytics (Analysis – Chapter 2):

- Data (What?)
  - Type of data (Enrollment, engagement and performance)
- Stakeholders (Who?)
  - Who are the teachers? (= What is their statistical background?)
- Objectives (Why?)
  - Why is this kind of data useful to the teachers to improve the content of their exercises?
- Methods (How?)
  - How is the data gathered?
  - How is the data visualized?

Afterwards we summarize the results and compare them to the existing features of the labsystem.

### 3.1 Coursera

The platform *Coursera* was founded by Daphne Koller and Andrew Ng in 2012. It is used by over 35 million learners and offers over 2.000 courses which makes it one of the biggest online learning platforms [14]. Their main goal is to make education available to everyone. They collaborate with over 150 universities which makes it possible for the students to get a university-recognized degree online. [15]

The courses are based on video lectures, quizzes and discussion forums. The reviewing of the assignments is done automatically and via peer-grading. [15]

The stakeholders of the visualizations are the teachers. Since they teach many different subjects, only basic visualizations can be used.

The platform offers a dashboard including many different graphics for the teachers. This table (Table 3.1) gives an overview over what kind of data is displayed:

Enrollment	Engagement	Performance
Enrollment evolution	Dropping the course	Performance Distribution
Gender		A/B-Testing of course versions
Age		Credit variation for the same answer
Education		
Country		

Table 3.1: Coursera Overview

As you can see in the Table 3.1, there are many different enrollment-graphics. They include the basic information about who is registered to the course: Gender, age, education, country. [16][17] These information is given freely by the users or derived from the ip address. They are presented in simple bar charts.

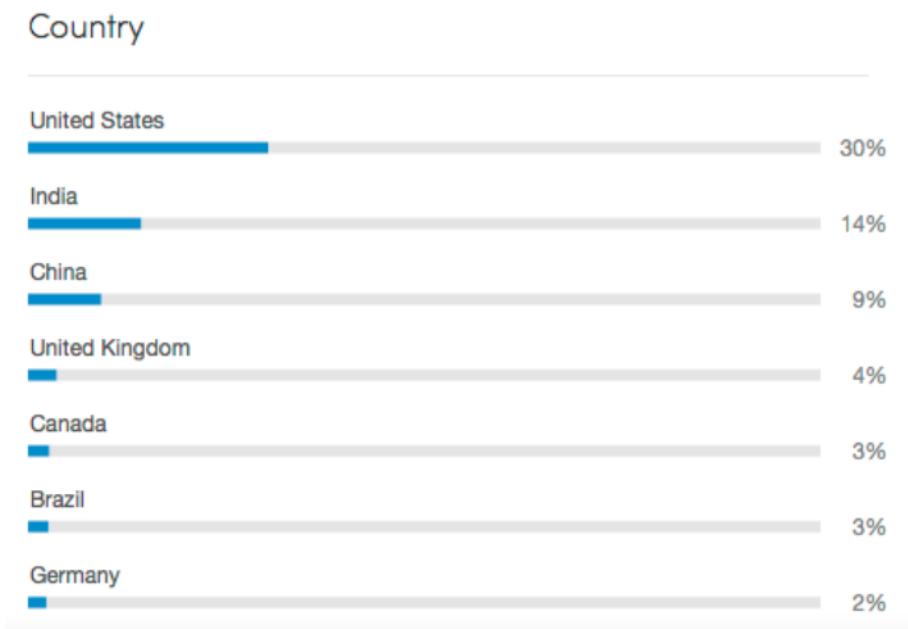


Figure 3.1: Coursera - Country [16]

Additionally, there is a line chart which shows the enrollment of students. [16]

There is also an engagement graphic. It displays retention of students, meaning the development of students dropping the course. [17] The data is derived from the logged engagement of the students with the course. When they stop participating at a certain point of the course, this can indicate a too difficult part of the course.

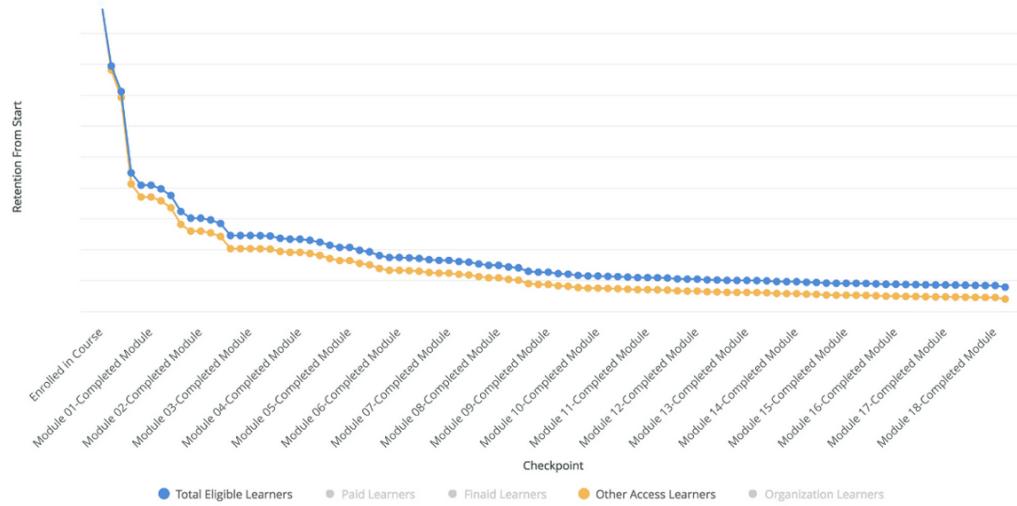


Figure 3.2: Coursera – Retention [19]

As can be seen in Figure 3.2, this development is displayed in a line chart. The two different lines represent two different types of students.

The website has also some performance methods: There is a grouped bar chart (Figure 3.2), which shows the performance distribution. It has two items – the first and the last attempt of an assessment – since it is possible to do a test multiple times. [16]

### Grade Distribution

The percentage of learners, in each grade range, for their first and last attempt.

First Attempt      Last Attempt

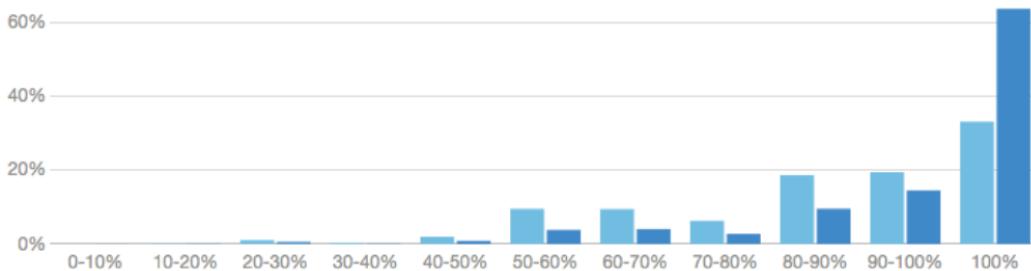


Figure 3.3: Coursera – Grade Distribution [16]

The data for this graphic is derived from the database.

Another feature of this learning platform is the possibility to use the A/B testing method. This means that one can compare multiple versions of the course (or parts of the course) by randomizing which students take which version of the course. When looking at the performance of both groups, it is possible to see, which group did better. Teachers can use this tool and thereby adapt the content of the course. [18]

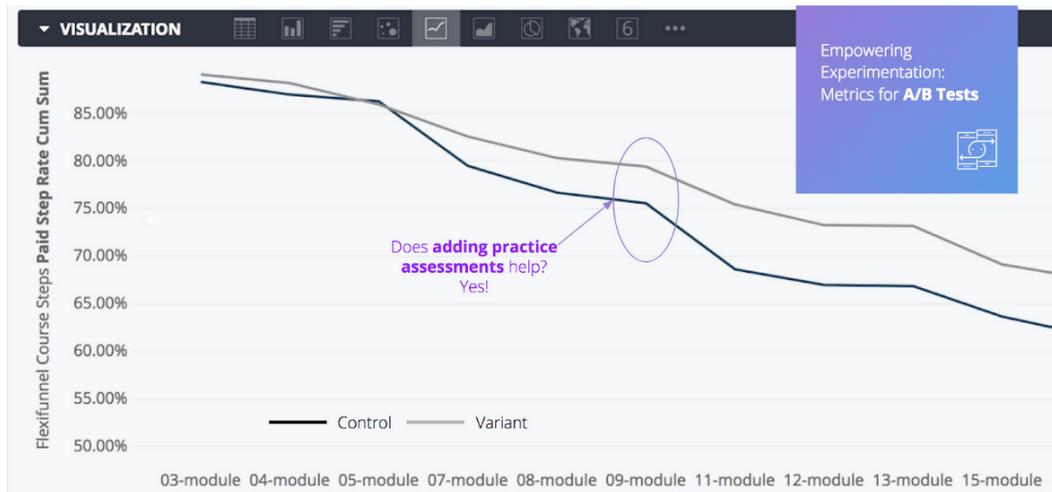


Figure 3.4: Coursera – A/B testing [18]

The results are presented in line charts. The two lines in Figure 3.3 represent the two different versions of the course.

It is possible to use peer-grading on *Coursera*. This means the students can correct the answers of other student. And since everyone corrects differently, this can be used to asses the clarity of questions and sample solutions. This is done by showing the variation of creditis students received for an identical answer. The data for this diagram is derived by means of the bootstrap method. [16]

## 3.2 edx

The platform *edx* is nonprofit and open source. It was founded in 2012 by the MIT and the Harvard University. It is used ‘on campus and online’ and is freely available for everyone. [20]

Enrollment	Engagement	Performance
Country	Engagement count	Performance Distribution
Gender		Answer variation
Age		
Education		

Table 3.2: edx - Overview

The dashboard of edx presents visualizations derived from methods of all three types to their teachers: (Table 3.2)

The enrollment charts offer information about the age, gender, education and country of the students. They are self-reported by the students and like on Coursera, this content is presented using bar charts:

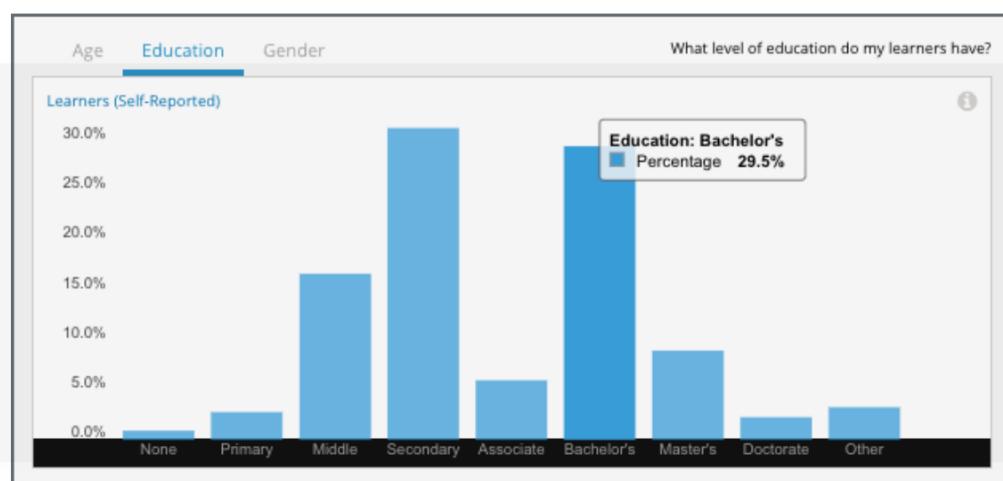


Figure 3.5: edx – Education [21]

The engagement graphics show the students interaction with the course.

The platforms presents the views for each exercise, assessment and video. They are displayed using bar and line charts (Figure 3.6).

The graphs for exercises and explanations can be used to figure out which parts of the course were more useful to the students than others (meaning they reviewed it multiple times). While the graphs for the assessments show their difficulty (because the students have to do them more than once). [21]

The line chart contains four different lines: The amount of students, watched videos, tried problems and participation in discussions.

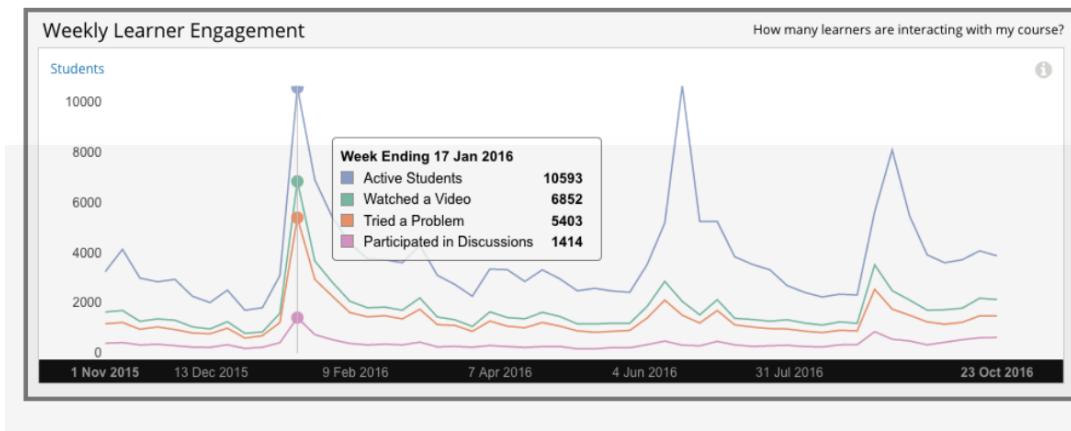


Figure 3.6: edx – Engagement [21]

It is also possible to see the complete (green) and incomplete (gray) video views. They are presented in Stacked Bar Charts (Figure 3.7). This may help to identify too long or uninteresting videos. [21]

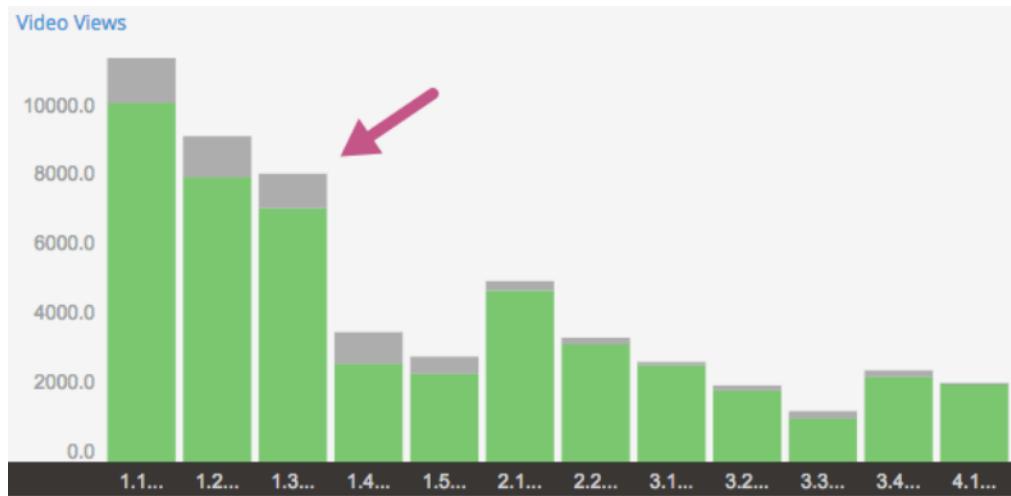


Figure 3.7: edx – Video Views [21]

edx has also visualizations of performance data: The performance distribution is shown in stacked bar charts (Figure 3.8). The color blue represents the correct/passed tests, while violet stands for the failed ones. [21]

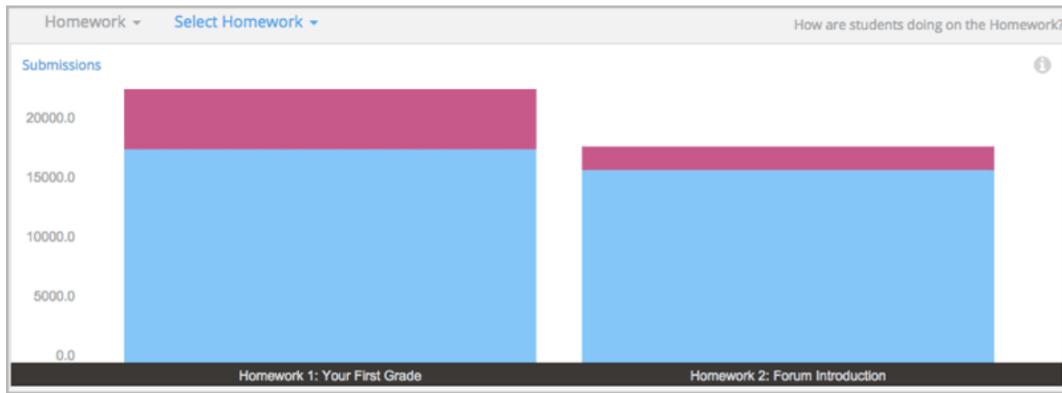


Figure 3.8: edx – Performance distribution [21]

The y-axis counts the submissions, which means that the chart is done in absolute numbers. The percentages can't be gathered from this chart. This visualization can be used to determine the difficulty of an exercise.

edx also displays the answer variation. As you can see in Figure 3.9, the bar chart compares the answer distribution. It also shows which answer were evaluated as correct (blue) or failed (violet). [21]

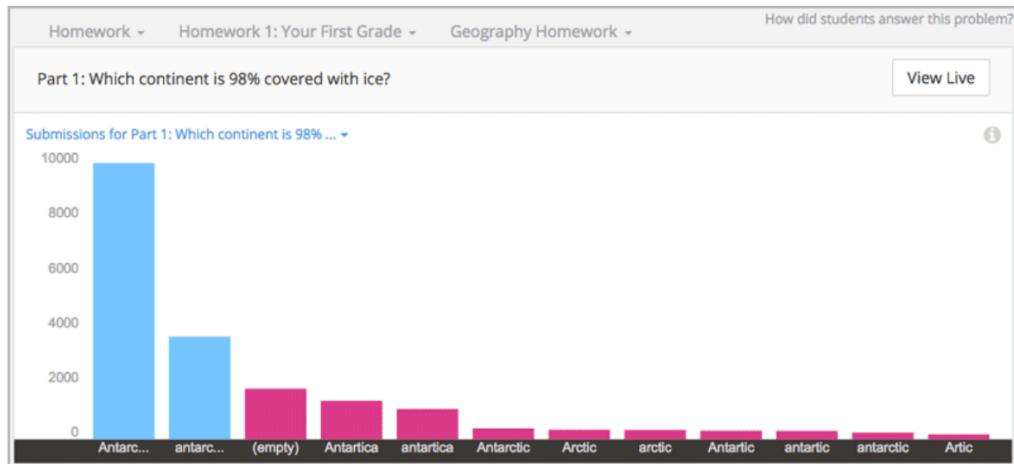


Figure 3.9: edx – Answer Variation [21]

### 3.3 Open University UK

The Open University UK was founded in 1969 by Royal Charter. It is a university for part-time students, allowing them to study online and on campus. They learn in tutorials, day schools, online conferencing, study networks and course forums. The main goal is to allow the students to study when and where they want to. [22]

Enrollment	Engagement	Performance
Age	Engagement count	Test results
Gender		Predictions
Origin		
Education		

Table 3.3: Open University UK – Overview

As you can see in Table 3.3, the *Open University UK* visualizes data of all three categories. But they are focused on the prediction on whether students complete the course. The predictions are weekly derived using machine learning. In the case of a student who probably drop the course, the teachers can do a proactive intervention. [23]

It is also possible to sort the students by many criteria, such as age, gender, origin and results. [23] This table then displays the test results (TMA), the predicted risk of failure, the final result prediction and the final result. (Figure 3.10)

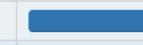
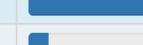
Student PI <sup>▲</sup>	Name	TMA	Risk of Failure	Final result prediction	Final result
A0001469	Nelson Patrick	NS NS NS NS	 Fail	 Fail: no resit	
A0002135	Hall Marie	90 98 90 87	 Pass	 Distinction	
A0003514	Smith Donald	75 29 58 S	 Fail	 Fail: no resit	
A0008150	Richardson Helen	80 89 77 NS	 Fail	 Pass	
A0009116	Edwards Robert	76 85 59 62	 Pass	 Pass	
A0013887	Nelson Dennis	NS NS NS NS	 Fail	 Fail: no resit	
A0014654	Gray Richard	78 57 NS NS	 Fail	 Fail: no resit	
A0017407	Sanders Stephen	96 95 83 92	 Pass	 Distinction	
A0020376	Collins Brenda	88 83 NS NS	 Fail	 Fail: no resit	
A0020561	Roberts Richard	58 57 67 S	 Fail	 Pass	

Figure 3.10: OUK – Predictions [23]

Therefore they combine all three categories in this table. They use bar charts to represent the risk of failure (a performance measurement). The rest is written down and highlighted with colors. The teachers can use this table to analyze the predictions and test results of their students.

This platform also offers an engagement and performance focused graphic. It displays the average clicks per student for each exercise (line chart) and the test results (bar charts). In the chart (Figure 3.11) the data of this year (orange) can then be compared to the data of last year (blue). [23]

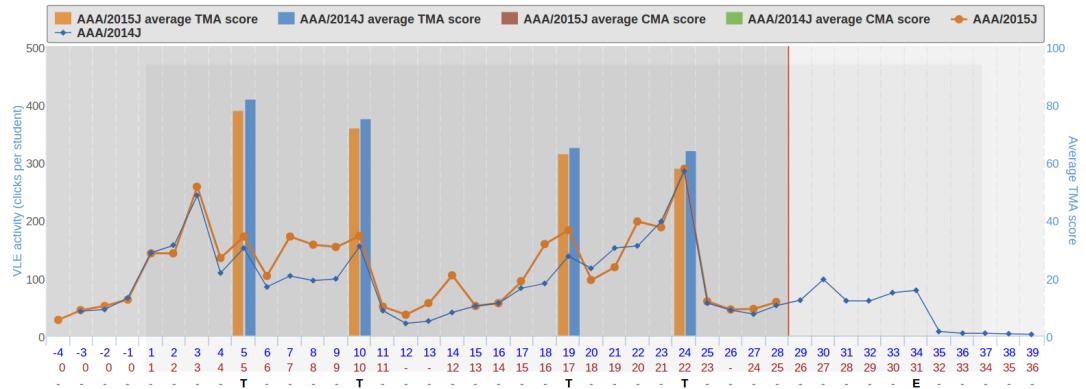


Figure 3.11: OUK – Engagement [23]

This makes it possible to compare the different parts of the course as well as compare one part over the years.

### 3.4 Discussion

In this chapter we presented the teachers dashboards of three different online learning platforms.

We could see that the platform use almost exclusively basic charts, such as line or bar charts, in order to make sure that everyone understands them. This is even more important considering that the target audience for general online learning platforms are not solely statistically educated people.

In the next subsections, we present our realizations of the three categories. Afterwards we compare the labsystem to the platforms.

### 3.4.1 Enrollment

The used data is mainly self-reported by the students and presented in bar charts. This information can be helpful to the teachers to adjust the course accordingly to their audience, like changing the difficulty, the language or the way the content is explained.

### 3.4.2 Engagement

This data is derived by analyzing the logging data. For this category (stacked) bar charts – highlighting the quantity of interaction - and line charts – highlighting the development over time/the course - are used. These visualizations can be used to identify difficulties and badly presented content.

### 3.4.3 Performance

For this category the presented platforms visualized the performance of the students (in relative and absolute numbers), the distribution of answers, the comparison of different versions of a course or the prediction on whether a student fail the course.

These data can be derived by both simple processes, such as counting the different results, or more complex procedures, like machine learning, the bootstrap method or A/B testing. It is usually visualized by using bar charts. These results can be used to detect imprecise questions or sample solutions, identify difficulties and finding the best way to teach something. It can also be used to help individual students.

### 3.4.4 Conclusion

In the last sections, we could see that all three categories of used data can be helpful to improve the quality of a online learning course.

The gathering of the data was often very simple, by counting or using simple statistic operations. We could also see that these platform use almost exclusively basic charts – or a combination of them.

As you can see in Table 3.4, all the platforms we presented in this chapter visualize data of all three types of categories. The Labsystem on the other hand only visualizes

the engagement (Time Tracking Algorithm) and the performance (Credit Distribution) of the students.

	Enrollment	Engagement	Performance
Coursera	✓	✓	✓
Edx	✓	✓	✓
Open University UK	✓	✓	✓
Labsystem		✓	✓

Table 3.4: Overview

None of the presented online learning platforms presented the performance results (e.g. credits) in both absolute and relative numbers. Coursera uses relative values, edx on the other hand displays the performance in absolute numbers. Another feature none of the platforms have is the time tracking algorithm.

We believe that the enrollment – who is using the labsystem – is not a priority, since the variety of students is very small. Therefore we want to focus on adapting the measurements of the engagement and the performance of the students.



# Chapter 4

## Design

In Chapter 3 we saw how other online learning platforms give their teachers visualized feedback. In this chapter, we apply the four dimensions of learning analytics to our main goal and then discuss our choices in how to achieve the requirements and goals as specified in the analysis.

### 4.1 Applying Learning Analytics

The main goal of this thesis is to give the teachers of the labsystem visualized feedback of user meta data in order to help them improve the quality of the content. We do this by using learning analytics.

In the analysis (Chapter 2) we discussed the four dimensions of learning analytics: Data, stakeholders, objectives and methods. Now we want to apply these dimension to our goal:

- The data we'll use is retrieved from the logging data and database of the labsystem. This includes data such as the questions, the achieved/possible credits for each question and the logged 'clicking events' of the users. For privacy reasons, we don't use any user specific data (as the name, age or origin).
- The stakeholders are the teachers of the iLab/iLab2.
- Our objective is to assess the current quality of the content and give visualized feedback to the teachers. They can use it to adapt the content accordingly.
- We do this by using statistical operations and visualizing the results.

### 4.2 Achieving the goals

In this section we explain and discuss how we want to achieve our goals – and their requirements – as specified in the analysis (Chapter 2). We present each goal subsequently in the following order:

1. Architecture
2. Credit Distribution Analytics
3. Time Tracking Algorithm

#### 4.2.1 Architecture

In this section we discuss how we designed the architecture of the code. In the analysis we stated the following requirements:

- (4) Modularized and self-explaining code (structure)
- (5) Good performance
- (6) Flexibility

In order to modularize the code (1), we divided the process of retrieving the data from the database to visualizing it into three consecutive steps:

Database → Calculation → Drawing

In Database, we get the needed data from the labsystem database/logging data. Afterwards we transform the raw data depending on what we want to display (Calculation). The results get passed on to Drawing. There the information gets visualized – no matter what we are presenting.

These three steps get connected by a Controller. We chose to add a Controller because this way the connection between the classes are very clear. One only needs to look at the Controller class, to get an overview over the code structure (1). This entity takes the results from one step and passes it on to the next one – not allowing the different parts to interact with each other. This makes it very easy to exchange parts or add new ones. In the implementation each step got transformed into a class. We now discuss every step/class in more detail:

##### a) Database

This class retrieves the needed data from the database. It also includes the step of preprocessing the data – like generating the time spans for the time tracking algorithm. Since this part can take quite long, we decided to save such information in an additional database. This ameliorate the performance and accomplished the second requirement.

## b) Calculation

This class receives the raw data from the database and calculates the values/arrays needed for the visualization.

It has four main purposes:

- Calculating the mean value
- Detecting the minimum/maximum
- Dividing each value of an array (to get relative values)

In order to pass all the information on to Drawing at once, we created another class called DataObject. We want the visualization to be flexible (3). This is achieved with the DataObject. It contains variables for all the possible needed information. And only the needed variables for the specific visualization is filled.

The process calling the appropriate methods of Calculation and then filling the variables of a DataObject object is done by a class called Coordination. Its output is a DataObject.

## c) Drawing

The class Drawing receives a filled DataObject. It uses the information to create a visualization using the methods of a helper class – called Canvas. This class contains all the basic drawing operations such as drawing a line, a rectangle or a label.

## d) Controller

The Controller organizes and connects all the other classes. In the following we discuss the different steps of the controller (Figure 4.1):

1. Receiving the URL and figuring out what the user wants to visualize. The coding system for the different settings is designed to be very flexible. It makes it easy to add new features (3). We explain the system in the Implementation (Chapter 5) in more detail.
2. Database: Calling the appropriate methods of Database in order to get the needed data.
3. Coordination: Calling the appropriate method of Coordination and passing on the data from Database. This class then calls the needed methods in Calculation and fills the results in a new DataObject object.
4. Drawing: Calling the drawing method in Drawing with the filled DataObject. This creates the graphic using the helper class Canvas.

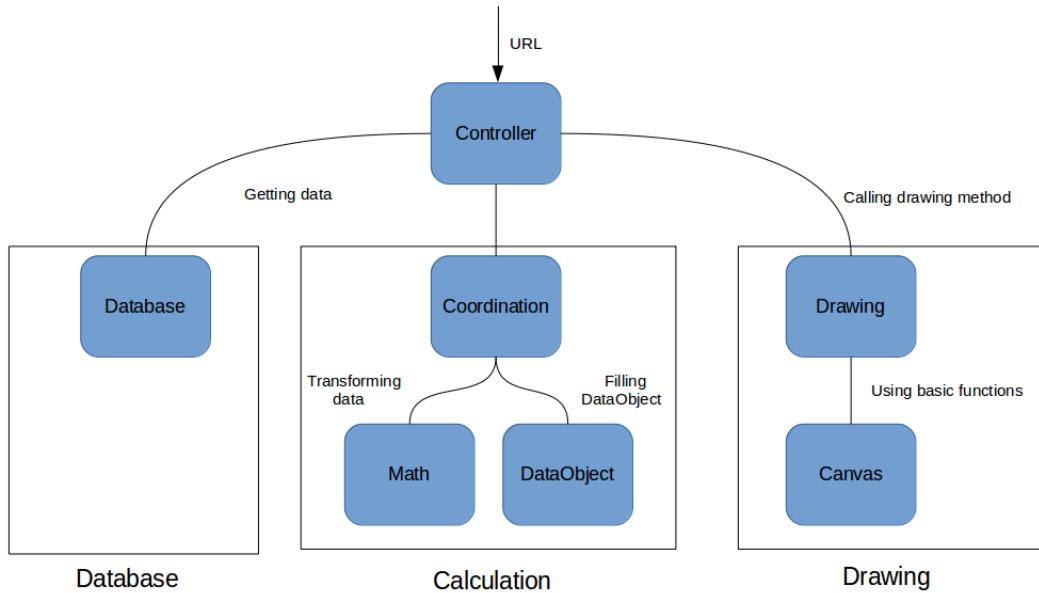


Figure 4.1: Overview Architecture

#### 4.2.2 Credit Distribution Analytics

In this section we discuss the design credit distribution analytics. This are the requirements we specified in the analysis (Chapter 2):

- (1) Setting extensions
- (2) Distribution
- (3) Comparability
- (4) Targeted Audience

Visualizing the credit distribution is a performance measurement. As we derived from the other online learning platforms (Chapter 3), the performance of students is both visualized in relative and absolute numbers. The analysis of the current credit visualization showed that either one or the other is implemented (Chapter 2). But like the other online learning platforms never both for the same data. Since both of them have their advantages, we want to implement both of them.

The current visualization allows a user to compare the questions performances of the practical part and to compare the achieved credits of different Labs. We want to add this setting to the visualization.

The existing visualization doesn't make it possible to see the achieved credits for the questions of the PreLab. We want to offer all the visualizations for both PreLab and practical part.

We believe each of the mentioned visualizations have their use and that's why we want to give the teachers the opportunity to chose the visualization they prefer.

Every setting is compatible with every other setting from another category. This means we have  $2^3 = 8$  different settings. (1)

Since each team has their own achieved credit value, we decided to calculate its mean value (from all the teams). Other statistical measurements (like the median) are insignificant, because sometimes an iLab/iLab2 course only consists of under 10 teams.

We want to be able to compare the values of different questions - exercises or Labs – easily (3). In the analysis (Chapter 2) we presented multiple visualizations:

We could use Circle Charts by creating a new chart for every question. The size of a part would represent the relative numbers of teams that have the same amount of achieved credits. There are three reasons why this solution is unsuitable: We want to be able to display both relative and absolute values. A Circle Chart would make that impossible. The next reason is that so many charts would seem quite chaotic. Additionally the values of different questions and of one questions would be difficult to compare – as we concluded in the Analysis (Chapter 2). This would contradict the third requirement.

Since we don't have continual data – which according to the analysis would make Line Charts suitable, only four types of charts remain: Bar Charts, Stacked Bar Charts, Error Bar Charts and Box Plots.

As we concluded in Related Work (Chapter 3), other online learning platforms often visualize performance measurements in bar charts. Their targeted audience differentiates from ours: They want the visualizations to be readable by everyone, we create the graphics for the teachers (who are statistically educated). Therefore, we are not limited to bar charts. (6)

As stated in the section above, we want to visualize the achieved credits and the possible credits/100%-mark. This means we already have two values per item, which makes Bar Charts not suitable. Since we only got two values, Stacked Bar Charts are a

good solution (see Analysis – Chapter 2). The first bar represents the given credits, while the second one marks the possible credits/the 100%-mark.

In order to highlight the first bar – since this is the more relevant value, we want to color it. The second bar remains blank and gets just encircled. As we could see in the analysis (Chapter 2), the Labsystem already has a color-performance system:

- Green: 80% to 100% of the possible credits
- Orange: 60% to 80% of the possible credits
- Red: Under 60% of the possible credits

We decided to use this system for the coloring of the bars. This highlights the achieved percentage for both the absolute and the relative chart. The color code is easy to understand for everyone. And a user of the labsystem already knows the (exact) meaning of each color.

Another requirement is to show the distribution of the team performances (4). As we have explained above, statistic measurements like the median are not useful. Therefore Box Plots are unsuitable. We decided to show the distribution by displaying the minimal and the maximal team-value for each item. This can be visualized using Error Bar Charts.

In order to visualize both the distribution and the achieved/possible credits, we decided to combine both charts. The combination of charts is a tool also used by other online learning platforms, such as the *Open University UK* (Chapter 3).

The resulting charts for relative values and absolute values are displayed in Figure 4.2 and 4.3. The gray questions are bonus questions (Figure 4.2). Therefore the number of possible credits is zero and it is impossible to visualize a relative value. In the absolute chart – Figure 4.3 – they are visualized like all the other items.

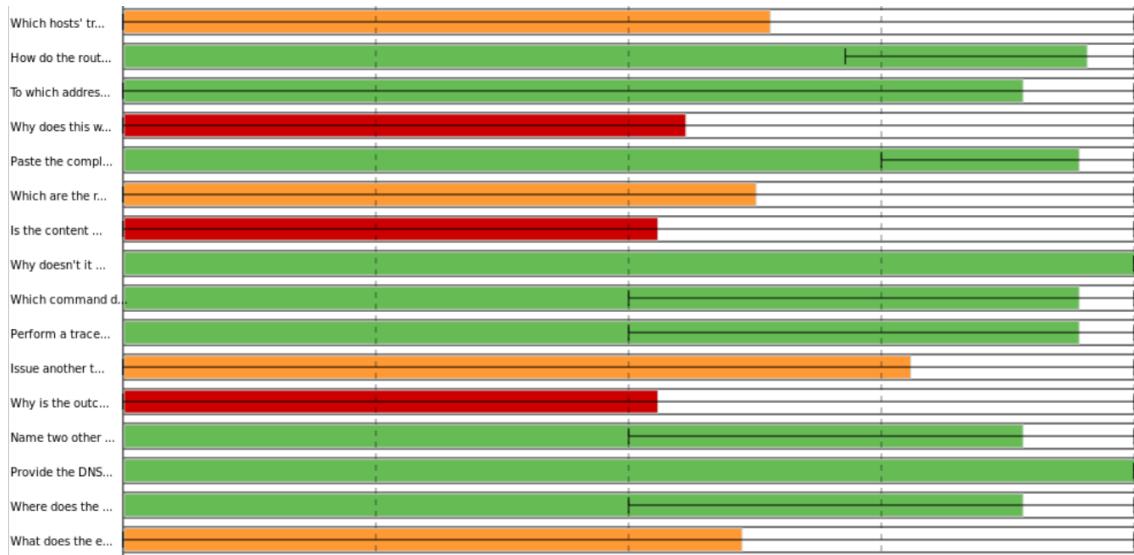


Figure 4.2: Relative Credits – Practical Part



Figure 4.3: Absolute Credits – Practical Part

#### 4.2.3 Time Tracking Algorithm

In this section, we discuss and present the design of the time tracking feature. We specified the requirements of this feature in the analysis (Chapter 2):

- (1) More specific analysis
- (2) Shorter Timeout

- (3) Setting extensions
- (4) Distribution
- (5) Comparability
- (6) Targeted Audience

#### a) Algorithm

The first part of the requirements of the Time Tracking Algorithm is about the algorithm itself. We described the existing algorithm in the analysis (Chapter 2) and concluded its flaws. In this section, we want to improve it using our requirements:

We want to briefly recapitulate the existing algorithm: It works by chronologically iterating through every opening/closing event of the students saved in the logging database. The time spans between the closing of a question and the opening of another one, is assumed to be preparation for the next question. Therefore it is added to the time spent on the question itself. It also stops the calculation in case of a logout or timeout.

In the analysis (Chapter 2), we saw that the algorithm needs to be more specific. We want to do this by differentiating text-elements and questions (1). This is done by not adding the time spent ‘in between’ questions to the next question. Instead this time is assumed to be spent on the text element(s) before the next question. It is important to notice that the next question isn’t the next question in the normal order of the Lab, but the next question that is opened.

This has the flaw that we rely heavily on the user behavior: If a student answers the question in a text editor before opening it and then just copying his text into input field, than the algorithm would assume that he spent all that time on the text – and not the question.

We also want to shorten the time span of the timeout (2). Instead of 60 minutes, we propose to adjust the timeout to the possible credits of the current question.

#### b) Data

The current visualization can’t compare the values of multiple Labs. It’s also not possible to see the visualization for the PreLab or to see the time spent on a question in relatively to the amount of credits. This is why we decided to have the same setting options as for the Credit Distribution Analytics:

- Inter/Intra Labs
- PreLab and practical part

- Relative and absolute values

For this data, we want to visualize the time spent on a question and the text element(s) before the question. Like for the credit distribution analytics, we have a value for each team. Therefore we calculate the mean of all teams for each question/text element. Since the amount of possible credits depends on the difficulty of the question, we want to visualize the absolute time spans and the relative time spans (time in relation to the possible credits) for each item. (1)

### c) Visualization

In the section(s) above we could see that we have very similar conditions as for the credit distribution analytics:

- Setting extensions: We concluded that we have the same settings for both.
- The same visualization requirements (4) (5) (6)

Therefore we decided to use the same visualization but with the new values:

One possible way would be to separate the text elements and the questions. This would mean each of them had their own visualization. Since this could cause the problem we discussed in section a), we decided to combine the text elements before a question and the question.

In the visualization this would mean that the first part of the Stacked Bar Chart represents the time spent on the question and second part displays the time spent on the text element(s) beforehand. This means we can see both values individually but still see the sum of them (the whole bar). This means we don't lose any data compared to the old algorithm, we just gain the separation (1)

The values of the error bar represent the minimal and the maximal value of the added time spans.



# Chapter 5

## Implementation

In the last chapter, we presented the design of the framework, the credit distribution and the time tracking algorithm. In this chapter we present three parts of the implementation. The different data types – and when they are used, the possible settings and the methods of Calculation.

In this thesis we focused on creating the architecture. We implemented the Credit Distribution Analytics but due to the limited time, we didn't implement the Time Tracking Algorithm.

### 5.1 Settings

There are currently 16 different settings describing what data to use and how to visualize it. In this section, we present all the possible settings and explain how they are implemented.

#### 5.1.1 Overview

There are four categories with two setting each ( $2^4 = 16$ ) from which the user can choose the settings:

- **Lab:** all Labs or one specific Lab
- **Part of the Lab:** PreLab or practical part
- **Data:** Credit distribution or time tracking
- **Type of Representation:** absolute or relative

Example: address=l11&type=122

→ address: This means we use the data from Lab 11.

→ Type: This means we use the data from the PreLab (1), calculate with the Time Tracking Algorithm (2) and display the data in relative values (2)

The different settings are defined in the URL with the values of ‘address’ and ‘type’. It is possible to combine every setting with every other setting. The system we used to specify the settings made sure that it is very easy to add new features to all categories in the future.

### 5.1.2 Lab

This category defines whether only the data of one Lab (and therefore each element of the Lab) or all Labs (and consequently the sum of all the elements) is displayed. The chosen option is defined with ‘address’ and is a string. For one Lab, this is the specific Lab ID, for all Labs it is ‘All’.

### 5.1.3 Part of the Lab

Here a user can choose whether he wants to see the data for the PreLab or the practical part (Lab). This category is represented with the first digit of ‘type’. ‘1’ stands for the PreLab, while ‘2’ means the practical part.

### 5.1.4 Data

Data describes the different features (such as credit distribution and time tracking). It is called data because this parameter defines which data we want to display. In the future it is possible to add many more features and thereby adding them to this category. Like the category Part of the Lab, it is also specified with ‘type’ - but the second digit: ‘1’ represents Credit Distribution and ‘2’ stands for Time Tracking – even though the latter feature is not implemented yet.

### 5.1.5 Type of Representation

This category lets a user choose whether he wants to see the absolute data or the data relative to the possible credits of the Lab/question. This category is represented with third digit of ‘type’. ‘1’ means absolute data and ‘2’ the relative data.

## 5.2 Methods of Calculation

In this section we explain the system behind the methods of the class Calculation.

Since it is possible to use one-dimensional up to three-dimensional arrays as input for the methods of calculations, the names of every function include the input dimension. The output has – unless stated otherwise – one dimension less than the input. For example, max2() expects a two-dimensional array as input and the output is an one-dimensional array. Methods that expect an one-dimensional array as the input, have a float as output.

Additionally there are function for ‘allLabs’. These have three-dimensional arrays as input – for each Lab a two-dimensional array – and call for each Lab the ‘normal’ two-dimensional method.

We didn’t combine the different function in order to make sure that the programmer is always aware of the input and the output.

## 5.3 Data Types

In this section we discuss the different types of the data that are being passed on between the different steps (Chapter 4 – Design). Figure 5.1 gives an overview over the process. In reality, the Controller transfers the data from one class to another. For the simplicity of the Figure, we omitted it.

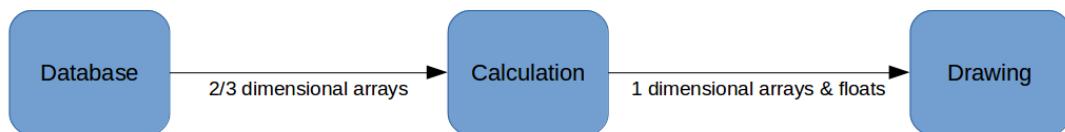


Figure 5.1: Data Types

### 5.3.1 Database to Calculation

The outputs of Database are two-dimensional arrays (or three-dimensional arrays for AllLabs).

- **Two-Dimensional Arrays:** The first parameter of the array represents the teams. And for each team there is an array that consists of a value for each question (second parameter). (Figure 5.2)

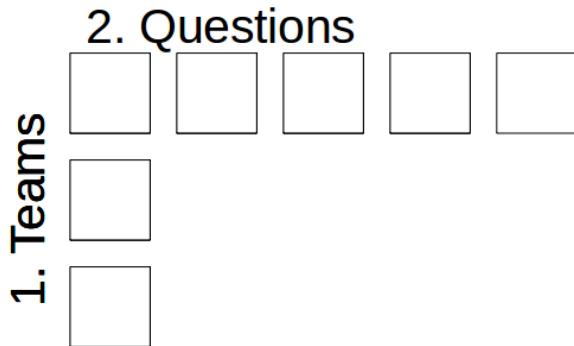


Figure 5.2: Two Dimensional Array

- **Three-Dimensional Arrays:** The first parameter describes the different Labs. Inside these elements the structure is the same as for the two-dimensional arrays.

### 5.3.2 Calculation to Drawing

Calculation receives as input one to three dimensional arrays.

- **Three Dimensional Arrays:** These kind of arrays exist only when comparing multiple Labs. This means we only want to compare sums. Therefore the first step is to build the sum of the values of every team for each Lab. The resulting two-dimensional array has as the first parameter the Lab and as the second parameter the teams. The value is the sum of the initial values.
- **Two Dimensional Arrays:** To get to a one dimensional array, the values of the second parameter get transformed. In Figure 5.2, this would be values for all the different questions. This can be done by calculating the mean value, the minimal/ maximal value or the sum.
- **One Dimensional Arrays:** These types of arrays can directly be passed on to drawing or they can get transformed in two different ways:
  - Output Float: Similar to the last step, calculating one statistic value (Sum, Min, Max, Mean)
  - Output One Dimensional Array: Dividing each value (in order to calculate the relative values)

The One Dimensional Arrays or Floats are used to fill a DataObject object and get passed on to Drawing.



# Chapter 6

## Evaluation

In this chapter we evaluate the results of this thesis. We do this by checking whether they fulfill the requirements we established in the analysis. Since our three main goals are independent, we discuss them subsequently.

The implemented architecture is modularized into three parts which are only connected via the control class. This modularization makes it easy to exchange or adapt parts of it. Therefore it is easy to add new features. This is also made possible by the flexibility of the setting system.

Since the structure of the code is based on logic steps from getting the data to visualizing it, it is easy to understand.

The performance of the architecture is ensured by the fact that results of time consuming operations (like the calculation of the time spans) are saved in an additional database.

These characteristics accomplish the requirements of the framework.

We implemented a feature called Credit Distribution Analytics. It takes the possible and achieved credits and visualizes them. This feature uses every existing option (as of now): This means it is possible to display the values of inter/intra Labs, the PreLab/the practical part and relative/absolute numbers. In the already existing visualization of the credit distribution, it wasn't possible to see the relative and absolute numbers for both PreLab and practical part.

In the visualization we use a combination of Error Bar Charts and Stacked Bar Charts. The Error Bar Charts show the distribution of the achieved credits, while the later makes it possible to display both the achieved and the possible credits. These chart forms make it possible to compare different values and are suited for our stakeholders (= targeted audience). By showing the distribution of the values, the user gets more detailed information than from the already existing visualizations.

The design of the time tracking algorithm ensures a more specific result than the current algorithm – without losing any information. This is done by differentiating the time spent on the questions and the explanations. It is also more accurate because of the shortened timeout.

Since the visualization has the same requirements like Credit Distribution Analytics, it is visualized in the same way and therefore meets the requirements. Contrary to the existing visualization, the values are comparable and the value distribution is displayed. It is also easy to understand, since we are using basic charts (like explained in the Analysis – Chapter 2). It is also possible to see the visualization for inter/intra Labs and relative/absolute numbers – which wasn't possible before.

Therefore every requirement for our artifacts is met.

# Chapter 7

## Conclusion

In this thesis, we discussed the ways of analyzing user meta data and visualizing it in order to help the teachers evaluate the quality of the content.

In the Analysis (Chapter 2) we reviewed the existing features of the labsystem. We realized that there isn't an architecture that makes it possible to easily add new features. As we have concluded in the Evaluation (Chapter 6), the now existing architecture is designed for this purpose and facilitates the process.

We discussed the two existing visualizations – the Time Tracking Algorithm and the Credit Distribution Analytics – and integrated an improved design of them (as concluded in the Evaluation) to the architecture.

We believe that the designed and implemented artifacts of this thesis sufficiently support and accomplish our goal.

### 7.1 Future Work

The implemented architecture sets the foundation for new features. It provides the processing and visualization of the data – so in order to built new features, only new data is needed:

Another possible future work is the implementation of the designed time tracking algorithm.

Another possible feature is the counting of tickets asking a question from the students for each question/text element. Many tickets for one could indicate a difficulty. Since question/text element the student is asking about is always linked in the ticket, this could be done automatically.

Another feature is based on the feedback given in the questionnaire, feedback-tickets and the feedback form at the end of each Lab. It could work similarly to

the ticket-counting. A way to automatize it, needs to be researched in future work. Otherwise the teachers could count and evaluate the feedback manually.

It is also be possible to analyze the enrollment of the students like is done by the presented online learning platforms (Related Work). It could be interesting to compare these information with the performance and engagement of the students.

# Bibliography

- [1] "Demonstation Lab", <https://ilab2.net.in.tum.d//pages/view.php?address=I2.C5.c3&config=demo>, 30<sup>th</sup> August 2019
- [2] "ilab2 Website", <https://ilab2.net.in.tum.de/>, 30<sup>th</sup> August 2019
- [3] Beatris Burdeva, "Improving eLearning through statistical Feedback", 2017
- [4] "Demonstration PreLab", <http://localhost/pages/view.php?address=I2.C6.c2&config=demo>, 30<sup>th</sup> August 2019
- [5] Ferguson, Rebecca (2012). Learning analytics: drivers, developments and challenges. International Journal of Technology Enhanced Learning, 4(5/6) pp. 304–317.
- [6] M.A. Chatti, A.L. Dyckhoff, U. Schroeder, and H. Thüs, "A Reference Model for Learning Analytics", [http://lanzarote.informatik.rwth-aachen.de/openlap/uploads/publications/IJTEL12\\_LA\\_Reference\\_Model.pdf](http://lanzarote.informatik.rwth-aachen.de/openlap/uploads/publications/IJTEL12_LA_Reference_Model.pdf), 28<sup>th</sup> July 2018
- [7] George Siemens and Ryan Baker, "Learning Analytics and Educational Data Mining: Towards Communication and Collaboration,  
<http://www.upenn.edu/learninganalytics/ryanbaker/LAKs%20reformatting%20v2.pdf>, 10<sup>th</sup> November 2018
- [8] "Insight edx Website", <https://insights.edx.org/>, accessed 28<sup>th</sup> July 2018
- [9] "ilab Website: Time Tracking Algorithm", <http://localhost/timeTracking/?address=I11&config=demo>, 28<sup>th</sup> July 2018
- [10] "Bar Charts", <https://www2.le.ac.uk/offices/ld/resources/study-guides-pdfs/numeracy-skills-pdfs/bar-charts-v0.1.pdf>, 13<sup>th</sup> November 2018
- [11] "Error Bar Charts", [https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Error-Bar\\_Charts.pdf](https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Error-Bar_Charts.pdf), 10<sup>th</sup> November 2018
- [12] "Example, Error Bar Chart", <https://pythonforundergradengineers.com/python-matplotlib-error-bars.html> 13<sup>th</sup> November 2018
- [13] Kerstin Völkl & Christian Korb, "Deskriptive Statistik", 2018
- [14] "Leadership Coursera", <https://about.coursera.org/leadership>, 28<sup>th</sup> of June 2018
- [15] "About Coursera", <https://about.coursera.org/>, 28<sup>th</sup> of June 2018
- [16] Joshua Horowitz, "Bringing Data To Teaching", <https://medium.com/coursera-engineering/bringing-data-to-teaching-20bb77ba0c00>, 28<sup>th</sup> of June 2018
- [17] "Instructor data dashboard, <http://ny-kim.com/work/dashboard/dashboard.html>, 28<sup>th</sup> of June 2018

- [18] "How A/B Testing Powers Pedagogy on Coursera", <https://medium.com/coursera-engineering/how-a-b-testing-powers-pedagogy-on-coursera-2cd10ed8365e>, 28<sup>th</sup> of June 2018
- [19]"Building data services to bring education to millions, Part II",  
<https://medium.com/coursera-engineering/building-data-services-to-bring-education-to-millions-part-ii-38bd3b2e93b3>, 28<sup>th</sup> of June 2018
- [20] "edx about Website", <https://www.edx.org/about-us>, 28<sup>th</sup> of June 2018
- [21] "Using edx insights", <https://media.readthedocs.org/pdf/edx-insights/latest/edx-insights.pdf>, 28<sup>th</sup> of June 2018
- [22] "Open University UK Website", <https://www.open.ac.uk>, 28<sup>th</sup> of June 2018
- [23] "Open University UK – Tutor Dashboard",  
[https://analyse.kmi.open.ac.uk/AnalyseDashboard\\_demo/login](https://analyse.kmi.open.ac.uk/AnalyseDashboard_demo/login), 28<sup>th</sup> of June 2018