INSTITUT FÜR INFORMATIK
der Ludwig-Maximilians-Universität München

Recognizing and Classifying Errors in Student Assignments using Human Computation

Mathias R. Schlenker

Masterthesis

Aufgabensteller	Prof. Dr. François Bry
Betreuer	Prof. Dr. François Bry, Niels Heller
Abgabe am	19. Juni 2017

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

München, den 19. Juni 2017 Mathias R. Schlenker

Abstract

Tutors and faculty members struggle to provide individual feedback on student assignments in large university classes. Furthermore, it is important that students receive their feedback shortly after their submission and that the feedback is of good quality.

This work tries to improve the correction process as well as the feedback quality in STEM[1]. In these fields learners independently make similar errors. Often, those errors are "systematic errors" resulting from misconceptions or misunderstanding of essential principles. Other recurring errors made by STEM learners are generally related to the specific subfield learned or to the context of the exercise. However, they do not need to be based on misconceptions. In this work they are referred to as "named errors" because they are named by the tutors or faculty members when encountered in students assignments. It is assumed, that the marked named errors for all assignments of a same course result in a small an consistent list of errors.

Using the named errors concept, a annotation component was created to assist in the correction process of student submissions. It was built into the Backstage 2.0 project platform. Furthermore, the annotation component allows tagging of named errors and text annotations in various kind of learning materials (PDF files, Markup snippets or code snippets). For the evaluation, formerly collected evaluations of students' assignments have been used. The participants tagged some provided named errors in the old submissions to collect information about the inter-rater agreement and information about the named error distribution. Finally, the participants had to answer a questionnaire covering questions about the component usability and the named error concept in general.

The annotation component appears to significantly improve the submission correction process. Furthermore, the named concept appears to be helpful for teachers and tutors as well as for students due to an easier correction process and improved feedback. However, the evaluation of the named error concept lacks generalizability, due to the small sample size. Thus, it is important to evaluate the concept in a bigger context which could be for example a lecture. Evaluating the component in a larger context, useful information about systematic and named error distribution can be collected.

[1] short for: science, technology, engineering and mathematics

Zusammenfassung

In großen Universitätsvorlesungen fällt es Tutoren und Fakultätsmitarbeitern oft schwer, individuelles Feedback auf Studentenabgaben zu geben. Außerdem ist es wichtig, dass Studenten ihr Feedback zeitnah nach ihren Abgaben bekommen und dass dieses Feedback von guter Qualität ist.
Diese Arbeit versucht denden Korrekturprozess und die Feedbackqualität in MINT Fächern zu verbessern. In diesen Disziplinen machen Lernende, unabhängig voneinander, ähnliche Fehler. Häufig sind dies "systematic errors", die von Fehlvorstellungen und Missverständnissen von essentiellen Prinzipien herrühren. Andere Fehler, die wiederholt auftreten und von Lernenden im MINT Bereich begangen werden, sind häufig abhängig von dem spezifischen Themengebiet oder dem Kontext der gestellten Aufgabe. Diese müssen jedoch nicht auf Missverständnissen basieren. In dieser Arbeit werden diese Fehler als "named errors" bezeichnet, da sie von Tutoren oder Fakultätsmitarbeitern benannt werden können, wenn sie in Studentenabgaben entdeckt werden. Es wird angenommen, dass die markierten Fehler für alle Abgaben von dem selben Kurs zu einer konsistenten Liste von Fehlern konvergieren.

Mit Hilfe von diesem Konzept wurde die Annotationskomponente implementiert, die den Korrekturprozess von Studentenabgaben unterstützt. Die Komponente wurde in die Backstage 2.0 projects plaform eingebunden. Die Annotationskomponente erlaubt das Markieren von named errors und Text Annotationen in verschiedenen Typen von Lernmaterialien (PDF Dateien, Markup und Code Auszüge). Für die Evaluation wurden frühere Ausarbeitungen von Studenten benutzt. Die Teilnehmer markierten einige bereitgestellte named errors in alten Abgaben, um sowohl Informationen über die Übereinstimmung zwischen den Teilnehmern als auch über die Fehlerverteilung generell zu sammeln. Letztendlich sollten die Teilnehmer einen Fragebogen beantworten, der Fragen zu der Usability der Platform und zu dem generellen named error Konzept abdeckt.

Die Annotationskomponente verbessert den Korrekturprozess von Studentenabgaben signifikant. Zudem scheint das named error Konzept hilfreich für Lehrer, Tutoren und Studenten zu sein. Dies basiert auf der Verbesserung des Korrekturprozesses und der Verbesserung des Feedbacks. Aufgrund der kleinen Teilnehmeranzahl ist die Evaluation des named error Konzeptes jedoch nicht generalisierbar. Mit der Evaluation der Annotationskomponente in einem größeren Maßstab könnten nützliche Informationen über systematic error und named error Verteilungen gesammelt werden.

Acknowledgments

I would like to thank Prof. Bry for his support and that he provided help and answers at any time. Furthermore, I would like to thank Niels for his outstanding backing and patience in every situation and question as well as for those various discussions e.g. about code style. Also, I would like to thank Basti for his support and introduction to Annoto.

Personally, I would like to thank my family, friends and Conny for the never ending backing in every stage and mood of my master degree. Finally, much love for Nic and his outstanding proof reading work as well as his countless (English) advices.

"We are men and our lot in life is to learn and to be hurled into inconceivable new worlds."
Carlos Castaneda, A Separate Reality

Contents

1 Introduction **1**

2 Related Work **3**
2.1 Learning Psychology . . . 3
2.1.1 Systematic and Named Errors . . . 3
2.1.2 Conceptual Change . . . 4
2.1.3 An Excursion into Self-Assessment and Self-Reflection . . . 6
2.1.4 Technology Enhanced Learning . . . 7
2.2 Human Computation . . . 8
2.2.1 Human Computation in General . . . 8
2.2.2 Human Computation for this Thesis . . . 10

3 Conception and Implementation **13**
3.1 Outline . . . 13
3.1.1 The Backstage 2.0 Platform . . . 13
3.1.2 Error Recognition and Classification Component . . . 14
3.2 Conceptional Aspects . . . 15
3.2.1 Teaching Scenarios . . . 15
3.2.2 Motivation for Participation . . . 16
3.2.3 Ethical and Privacy Aspects . . . 16
3.2.4 Design Decisions . . . 17
3.3 Implementation Process . . . 19
3.3.1 Development Outline . . . 20
3.3.2 Difficulties Encountered . . . 21
3.3.3 Lessons learned . . . 21

4 Evaluation **23**
4.1 Evaluation Goals . . . 23
4.2 Study Design . . . 23
4.2.1 Questionnaire . . . 24
4.3 Study Methodology and Results . . . 25
4.3.1 Questionnaire . . . 25
4.3.2 Named Error Distribution . . . 27
4.4 Result Interpretation and Discussion . . . 29
4.4.1 Evaluation Criticism . . . 29
4.4.2 Questionnaire . . . 29
4.4.3 Named Error Distribution . . . 30

5 Discussion **33**
5.1 Results . . . 33
5.2 Further Thoughts on the Backstage 2.0 Platform and its Components . . . 34
5.2.1 Trust Model . . . 34
5.2.2 Development of the Component . . . 35
5.2.3 Approaches of the Component Usage . . . 35
5.2.4 Platform-Extensions . . . 36
5.3 Further Thoughts in General . . . 37

6 Appendix **39**
6.1 Source Code Repositories . . . 39
6.2 Word and Phrase Explanations . . . 40
6.3 The Questionnaire . . . 41
6.4 The Exercises used in the Evaluation . . . 42
6.5 Named Errors used in the Evaluation . . . 43
6.5.1 Provided Named Errors . . . 43
6.5.2 Named Errors created by Participants . . . 43

References **45**

CHAPTER 1

Introduction

Many students know situations as the following: One is given multiple weekly assignments and is virtually working on an endless queue of deadlines. Usually, assignments need hours or even days of work to be completed. Thus, after having spent time and effort on completing an assignment, one would naturally appreciate receiving feedback. On the other hand, tutors are often students of higher semesters and whether they are faculty members or undergraduate assistants, they typically are also under a deadline pressure.

This thesis aims at improving the correction process for student assignments as well as the feedback and learning process of students in disciplines such as science, technology, engineering and mathematics (so called STEM).

To achieve this, the issues above are analyzed: Students might learn better if they get feedback on their assignments shortly after having completed them. However, in large classes, tutors and faculty alike struggle to provide individual feedback on student assignments. Such a feedback can hardly be provided without delay. Additionally, a feedback to a student on her or his assignments should be of good quality and preferably adapted to the student. Otherwise, students could be indifferent to the feedback or even misunderstand it.
Furthermore, in STEM student errors often seem to be based on similar "error ideas". These errors follow a clear scheme and have distinguishable characteristics. An example is the mistake called freshman's dream. The freshman's dream describes the erroneous assumption that $(a+b)^n = a^n + b^n$ for $n \geq 1$ and $n \in \mathbb{R}$. Errors such as this one are often referred to as systematic errors. Systematic errors appear to be based on misconceptions[3] and thus are resistant to change through traditional instruction methods.[6] Nevertheless, there are learning psychology models such as the conceptual change model which deal with the issue of misconceptions.[26] Using the idea of systematic errors, a general notion is considered, called named errors. Named errors include systematic errors as well as any other errors, that are teaching and correcting team might find appropriate. They are recurring errors in the context of a given scope such as a class or an assignment. However, they do not need to be based on conceptional misunderstandings.

Some named errors can be hard to detect and they can even be harder to detect with software in an automated fashion. An example can be written proofs in mathematics. Natural language processing is needed to access syntactical correctness. However, semantical

correctness of a proof is hardly decidable with software. Nevertheless, humans can find it easy to recognize named errors if they are familiar with the topic. Thus, a computational approach called human computation was chosen. Human computation uses humans as a computational component in that they create named error sets and aggregate error links in student assignments. With these aspects in mind, an annotation component in this thesis was designed and implemented. The application provides an annotation functionality including named errors. The implementation is done as a component for the newly developed Backstage 2.0 project platform by the *Lehr- und Forschungseinheit für Programmier- und Modellierungssprachen at LMU Munich*[1].

Using the presented implementation, tutors and teachers can tag errors in student solutions without typing or copying the whole error description every time an error occurs, thus saving time. Additionally, students gain feedback in terms of a whole concept they did wrong. Furthermore, the accurate position of their mistake is provided, which can result in a better learning experience. With the help of human computation, the errors can be aggregated. The gained data can be used to analyze named and systematic error distributions and to optimize named error sets for concrete topics. An additional approach is to ensure a certain quality of annotations in teaching models such as peer review. Therefore, the data gained within the human computation component can be used to calculate the credibility of users in terms of tagging named errors.

Outline

In the second chapter *Related Work* various topics related to this thesis are outlined. Chapter 2 covers the learning psychology background with fields such as the conceptual change model, self-assessment and technology enhanced learning. Furthermore, the human computation approach is described in general and in the context of this thesis. Chapter 3 *Conception and Implementation* displays the overview of the component and the Backstage 2.0 platform in which the component is build in. Various conceptional aspects such as teaching scenarios design decisions and ethical and privacy aspects are explained in detail. The end of chapter 3 deals with the development-process, lists issues occurred and provides lessons learned. In Chapter 4 *Evaluation*, the implementation of the component is evaluated. The goals as well as the conception of the evaluation are described. Furthermore, the results of the study are presented and analyzed. In chapter 5 *Discussion*, the results of this work are outlined. The outline is followed by a discussion and further thoughts about the annotation component, the Backstage 2.0 project platform and the named error concept. In the final chapter *Appendix*, a short word and phrase explanation is provided and additional material such as the evaluation questionnaire is appended.

[1] `http://www.pms.ifi.lmu.de/`

CHAPTER 2

Related Work

This thesis is related to various fields, for example education and learning psychology. Furthermore, software development and the areas of its application are involved. The field of technology enhanced learning combines both fields named before. Last but not least, the thesis refers to the field of human computation with the goal to aggregate, process and analyze the data.

To capture the state of the art and establish a basis for the following work, this chapter provides an overview of the topics named above. In the learning psychology section the concept of systematic and named errors is described, as well as a learning model related to systematic errors, which is conceptual change. Furthermore, basic technology enhanced learning principles and approaches are explained. The section focuses on the consequences of ubiquitous of the technology, such as the use of technical devices and applications in the classroom. Additionally, techniques are outlined which could be used for further data validation, data processing and data analysis.

2.1 Learning Psychology

A major goal of this thesis is to improve the learning process. In this work, improvements in the learning and teaching process both for teachers and students are developed: the teacher and tutor side as well as the student side. Besides the technical processes and workflows of the correction process, the psychological aspect of learning has to be considered. It would exceed the boundaries of this work to analyze and exhibit the wide field of learning psychology and the various mechanisms with which knowledge can be acquired. Thus, the focus of this section is not to provide a complete review of learning psychology, but instead to briefly outline the theory and approaches which were considered important for this work.

2.1.1 Systematic and Named Errors

To structure recurring errors, a categorization scheme is needed. In this work, two major error types are distinguished: Systematic errors, which describe error concepts predomi-

nantly occurring in STEM disciplines, are based on misconceptions created through faultily added knowledge.[26] An example is described in the following. It was provided by Niels Heller (personal communication, May 29, 2017). Students were asked to implement the function `moveIt`. For scaffolding, this code snippet was given:

```
function moveIt(objects,moveFunc) {
        return objects
        //write your code here
}
```

Some students started to write their code after the `return` statement. However, this code is unreachable. Thus, there is clearly a misconception of what the return statement does. Additionally, another dimension is visible: For future assignments the two lines should be switched.

Furthermore, there are named errors. Named errors are a superset of systematic errors. They also follow a defined scheme and are thus group-able. In contrast to systematic errors, named errors can not be linked to an error concept. They are thus more general error groupings. An example for a named error is a *misunderstood exercise*. Furthermore, one could think of the following scenario, also provided by Niels Heller (personal communication, May 29, 2017). In theoretical computer science, if M is an automaton $L(M)$ denotes the language that M accepts. A different notation for this is $T(M)$. The lecture used $L(M)$, the exercise $T(M)$. This caused some "systematic" problems. It is debatable if this is an error, because it was caused by the teaching staff. However, it is something that should be explained in the correction process.
Systematic errors have various attributes, e.g, they are showing consistency across groups with different characteristics.[3] They are resistant to change through methods such as traditional instruction.[26] This relates to concepts such as the conceptual change model which is further described in section 2.1.2. Furthermore, they are content and context based errors.

Example usage of an error categorization: Elliott et al.[4] provide an example how error categorization can be used. In their study, error grouping was used to classify the correctness of automated machine translations. They created groups such as "unnecessary determiner" or "inappropriate noun" to group occurring errors and to analyze and weight the errors in a structured manner.

2.1.2 Conceptual Change

The processing of information as well as the creation of knowledge are two very complex topics. Various models for knowledge creation and knowledge modification have been created in the past.[26] The conceptual change model deals with the integration of new information into existing knowledge. The main approach is that learners want to resolve conflicts created by new experiences to aspire a homogeneous state of knowledge. For example, children often times create a world view which is based on the world being flat. When the flat surface model is expanded by teaching the child that the earth is actually a sphere, misconceptions can emerge. One misconception could be that people are living inside the sphere.[26] Another one, which was found by Vosniadou and Stella, was that the earth was a sphere, but flat at the top and the bottom. These misconceptions have been created by adding information to the existing knowledge ("the earth is a sphere"). By using more examples or existing frameworks such as gravity, the newly added model can be revised easier. Vosniadou and Stella collected data about two more examples: the day and night cycle as well as heat transfer models.[26]

The conceptual change model was developed from at least two research fields: developmental psychology and science education. Because conceptual change is an active learning process, the aspect of motivation needs to be considered.[18] So, this process can be supported by intrinsic or extrinsic motivation.

Furthermore, scientific metaphysical beliefs seem to have a big influence on judgments, which are made about new knowledge. Scientific metaphysical beliefs are beliefs about the nature of reality [1] that are e.g. shaped by the world view or religious beliefs. The influence on judgments is due to metaphysical beliefs often shaping the existing knowledge in terms of epistemological commitments. For example: The fact that time is relative can be unintuitive for learners to understand because he or she could have been previously confident that time is an absolute value.[16] The conceptual change model was chosen as a model because it was validated through and based on various different studies (see e.g. [18] and [26]) and it treats misconceptions as part of the learning process.

In [26] Vosniadou and Stella provide an overview of the conceptual change model. Vosniadou and Stella focused on teaching in the field of physics. Children acquire a naive framework for the field of physics from a very early age. This framework theory is generated by experiences of everyday life. For example, the fact that the horizon seems like a horizontal line implies that the earth is flat. Knowledge is gained by adding new information to the existing knowledge. Thus, information is transformed into concepts. These concepts then are added into larger theoretical structures such as the framework theories.

At this point, the conceptual change model proposes two outcomes: The first one is *enrichment* of the existing knowledge. The newly gained knowledge can be added without conflicts to the existing knowledge. In other words, the knowledge can be expanded. In other works such as from Posner et al.[16] it is termed *assimilation*. The second one is called *revision*. Revision is needed if the newly gained information is conflicting with the existing knowledge. In this case, changes in individual beliefs or at the level of the framework theory itself are necessary. Changes in individual beliefs seem easier than changes in the framework theory itself. Additionally, changes in the framework theory are more likely to cause misconceptions. Misconceptions are produced if inconsistent information is reconciled and a synthetic mental model is produced. These models can be nevertheless internally consistent and well-defined. Furthermore, these models can be technically functional which can have a big influence on the beliefs. This second outcome is also called *accommodation* by others such as Posner et al.[16]

Criticism of the study by Vosniadou and Stella: One of the major points criticized is that the study focuses on the development of knowledge in children. The developmental psychology of children and the issues occurring at this point is not equatable with the situation of university students. This is primarily criticized in terms of teaching processes which concern the conceptual change model. Nevertheless the basic underlying concepts seem to be the same over various age groups.[18]

[1] https://plato.stanford.edu/entries/metaphysics/

Teaching improvements learned from the conceptual change model by Vosniadou and Stella:

- Learning situations should be created in which the student is actually "doing" science. This means that students can experiment and test a hypothesis to gain information about its correctness.
- Students should provide verbal explanations. They should further share and discuss them with others to gain feedback about the validity of their presumptions.
- Teachers should create environments that allow students to express their representations of facts. This provides the opportunity of gaining experience of revising.
- Generative Questions should be asked, such as in the following example: *"Scientifically correct responses to these questions do not necessarily mean that the students have understood the concept in question, because students often repeat the information they have received through instruction without fully understanding it."*[26, page 50] This can force the explanation of the concept itself instead of just reproducing the information students were taught.

Further approaches to improve the learning process:

- Newly introduced scientific theories should be taught in an intelligible, plausible and fruitful manner. This means that more emphasis should be given on such information that could lead to conceptual conflicts. Anomalies discovered in the past should be displayed. Finally, metaphors, models and analogies should be used to make the new concept more plausible.[16] Only adding information in terms of facts to existing knowledge can result in conflicts and create misconceptions. A better approach could be to teach concepts instead of only giving information.[26]
- Exercises in lectures, demonstrations and labs can be used to create cognitive conflicts in students.[21](cited in [16]) This point has to be handled with care, because the conflicts often fail to initiate the conceptual change needed for an understanding of a scientific concept.[18]
- Create evaluation techniques to track the progress of the learning state and conceptual change.[15](cited in [16])

2.1.3 An Excursion into Self-Assessment and Self-Reflection

It is important for students to know how they actually perform and how they perform compared to their peers. But as the task of self-assessment is very complex, this leads to multiple issues. For example the work of Kruger and Dunning shows that skills that are creating competence in a specific domain are the same skills necessary to evaluate competence in the domain. In their work, they carried out four different studies to analyze the difficulties of self-assessment.[7]

Self-assessment predictions and study results Kruger and Dunning predicted that incompetent individuals will dramatically overestimate their abilities relative to objective criteria. Also, they suffer from lack of metacognitive skills, meaning that they are worse at recognizing competence. Thus they are struggling to gain insight into their "true level of performance". Last but not least Kruger and Dunning predict that incompetent individuals *"can gain insight about their shortcomings, but this comes (paradoxical) by making them [(the individuals)] more competent"*.[7, page. 1122] The studies were designed to test various fields

such as humor, logical reasoning and grammar standards. The results show that incompetent individuals tend to overestimate themselves while competent individuals tend to underestimate themselves. In addition, the results show that an actual comparison of student performances with his or her peer students can lead to a better self-assessment. The last prediction was confirmed with the fact that students perform better in self-assessment if they were taught in the specific topic.

Conclusion The component implemented in this thesis should provide students the options to self-reflect on their prior mistakes in an adequate manner. They should be able to improve their knowledge[5] and, according to this newly gained knowledge, they should be able to assess themselves better. While self-reflection is a never ending process, we want to focus on self-reflection of students in terms of assignments - especially if they are provided with submission feedback. Other studies, such as a study by Sadler and Good, came to the same result in terms of self-assessment and self-grading.[20] Following the results of Kruger and Dunning as well as Sadler and Good, a more accurate feedback model to improve the learning process of the students can be created. Using the findings of Weaver[27], the feedback model could be specified more accurately: It should not be too general nor to short and should contain suggestions for improvement. These suggestions can be e.g. provided through the detailed information of a named error. Furthermore, students learning material and information about common named errors in a specific topic can be served. Especially not so well performing students might benefit from this.

2.1.4 Technology Enhanced Learning

Nowadays technical devices of every kind do not only influence our private and work lives, but also play a big role in education. Many schools and universities use e.g. digital devices such as projectors to display slides or make learning material accessible through content management platforms such as moodle[2]. Additionally, the learning process can be improved through applications such as intelligent tutoring systems (ITS). ITS can provide customized instructions and feedback to students. Roll et al.[19] analyzed the help seeking actions in ITS and e.g. found out that hints tend to be used faulty. They improved the feedback process by providing a meta-cognitive feedback which is adapted to each student (for example *"It could be that another hint will do the trick for you."*[19, page 4]).
In research, the field of technical learning assistance is often referred to as technology enhanced learning (TEL). Even older articles and studies such as Attwell et al. from 2007 [1] recognized the potential of the digital progress in terms of TEL. The emergence of ubiquitous computing, the development of social software as well as the option to create personal learning environments provides the foundation for further developments. In the end Attwell et al. came to conclusion that the argument for the use of the software is not only technical, but rather philosophical, ethical and pedagogical.

The Backstage 2.0 lecture and project platform are examples for technology enhanced learning. The project platform supports the student within the learning process and the teacher and tutors within the teaching process. New approaches such as these ones provide new possibilities. For example an improved peer review model can be implemented such as the following: The work [13] by Piech et al. uses data from the massive open-access online courses (MOOCs) platform *Coursera*[3]. Initially, too many solutions had to be corrected which posed a problem. To solve it, the platform uses a peer-grading approach. Every student has to correct 3-5 anonymized solutions from other students. In every 3-5

[2] `https://moodle.org/?lang=de`
[3] `https://www.coursera.org/`

solutions there is one solution which was also corrected by a tutor to generate a "ground truth" or "golden standard". This data is further used to calculate the reliability of given corrections.

Teacher review and peer review A study by Sadler and Good shows that self- and peer-grading can have a positive impact on the students learning process.[20] In their study they compared the peer review approach with a self-review approach as well as with a teacher review approach. The study is evaluated in seventh grade classes. Nevertheless, the study results refer to the learning process itself. Thus, they can be applied within the context of peer review of student assignments. They collected advantages of peer review and self-correction in various aspects:

- Logistical aspects: If students correct each other, it saves time for the teacher. Furthermore, the students gain the feedback faster and more time could be spend on each feedback. Thus the feedback tends to be more detailed.
- Pedagogical aspects: Students gain insight in different point of views when they correct solutions of each other. This results in a better understanding of the learned material.
- Metacognitive and affective aspects: The correction process leads to more self-awareness by providing another way of learning. Thus, this leads to a better self-assessment. These results are comparable to the results of Dunning and Kruger.[7] Furthermore, the changes can make the classroom more productive, for example by creating new learn experiences and tasks. The tests can be categorized as constructive feedback instead of "grading only". This can result in more motivation within the context of the classroom.[20]

Furthermore Sadler and Good recommend to use a grading rubric to gain adequate results of student corrections. Applied to the context of this thesis, teachers can provide a guideline about the systematic errors occurring in an specific assignment. A grading rubric in general should not be necessary, because errors only need to be tagged and not graded.

2.2 Human Computation

Human computation has a wide field of application. At first the general approach is described by giving an example of a use case. Then the application of the human computation approach of this thesis is specified.

2.2.1 Human Computation in General

Excursion into complex, human solvable tasks Even though the research field and the knowledge in computer science grew a lot over the last years, there are still tasks which seem impossible to compute in an automated manner. However, techniques using advanced algorithms or machine learning are capable of solving complex problems which have been previously believed to be solvable only by humans. An example is the application *AlphaGo* by Google which won a game of Go against Lee Sedol, *"the top Go player in the world over the past decade"*.[4] Go is described as a game that requires intuition instead of only tactical thinking. For example it would require a gigantic search tree of all possible moves which would be not calculable at the time of this report. Instead of using a search tree,

[4]For more information: `https://deepmind.com/research/alphago/`

AlphaGo uses the approach of artificial neural networks[5]. Most approaches in the field of machine learning are already a few years old or are reaching back to the mid 20th century. One of the reasons why this topic has become so popular nowadays is that computing power and data storage capability has increased tremendously in the last decades. Thus, fields such as computer vision, data categorization and clustering as well as data-mining approaches are developing fast.

The human computation approach Nevertheless, there are still issues which are hard to compute or nearly impossible to solve for machines but are easier to handle for humans. For those issues the human computation approach can be used. In turn, the gained data then can be used as training data for various algorithms. Human computation uses the human as an computational component for those kinds of problems that are considered unsolvable by computers. The area of applications for this approach reaches from artificial intelligence, business, cryptography, and art over evolutionary algorithms[6] to human computer interaction (short: HCI).[17] Many current works belonging to the field of human computation refer to the work of von Ahn. Von Ahn introduces the approach with implementations and evaluation on applications within a serious context (such as CAPTCHA) and within a gamified context (such as the ESP game).[23] If the human computation approach is used in a game context, the application is often referred as a *"game with a purpose"* (short: GWAP).

Law and Ahn state, that there are two types of truth: the objective truth and the cultural truth.[8] The objective truth is a truth which is *"external to human judgment"*.[8, page 26] In the cultural truth, *"the true answer refers to the shared beliefs amongst the set of people that we sample, and determining this answer usually involves some sort of perceptual judgment"*.[8, page 26] In addition this truth is affected by cultural change. Ahn provides in this case the example of Michael Jackson: *"an image of Michael Jackson twenty years ago might have been labeled as "amazing" whereas today [2005] it might be labeled as 'guilty.' "*[23, page 23] Furthermore, Quinn and Bederson collected information about the usage and taxonomy of applications and research in the human computation field. They expanded the basic definition by Ahn with the following two additions: *"The problems fit the general paradigm of computation, and as such might someday be solvable by computers."* and *"The human participation is directed by the computational system or process"*.[17, page 2] They also created a more detailed differentiation with related concepts such as crowd-sourcing, social computing, data-mining and collective intelligence.

Human computation tasks Basically human computation tasks tend to be rather small and are often called *micro tasks* or *human intelligence tasks* (HITs). If a task is complex, it is usually split into smaller micro tasks. Those tasks are then presented to users within a serious or game like context. In both contexts motivation plays a major role. The most important motivation factors according to Quinn and Bederson are payment, altruism, enjoyment, reputation and implicit work.

Data processing and quality assurance If the task is delivered in a serious context, so called micro tasking platforms are often used to deliver the task to users which should

[5]Definition: `https://en.wikipedia.org/wiki/Artificial_neural_network` - The approach is using a two-step network: In the first step, called policy network, the move search process is evaluated. In other words, a few possible moves are selected instead of iterating over all possibilities. In the second step, the value network processes the following steps of the the selected moves to an approximately depth of 20 moves. The result is then valued in terms of the chance of winning the game.

[6]Definition: `https://en.wikipedia.org/wiki/Evolutionary_algorithm`

solve them in turn. An example for a micro-tasking or crowd-sourcing platform is Amazon Mechanical Turk[7]. The gained data then has to be aggregated. If the collected data consists of e.g. user votes, it can be very noisy. To solve this issue, approaches such as the aggregation through principled voting exist.[10]

Example for a human computation approach Providing an example application of the human computation approach, the one named above is explained in more detail: *CAPTCHA* (*"Completely Automated Public Turing test to tell Computers and Humans Apart"*) uses an unsolved artificial intelligence problem to ensure that a user is a real human and not a bot. Other approaches are online polls, free email services, search engine bots, preventing spam and dictionary attacks.[23] The task given is to recognize the letters and numbers in a distorted image, which is generated automatically.[24] Submissions are then matched to the solution and evaluated.

The concept was adopted by many websites and two issues occurred: On the one hand companies provided services to e.g. solve CAPTCHAs for money.[11] On the other hand machine learning algorithms as well as processing power improved over time and are capable of solving various CAPTCHA implementations.[28] Therefore, von Ahn et al. developed and implemented *re:CAPTCHA*, which is using two real text snippets instead of one computer generated, distorted text.[25] The new approach did not solve the issues completely, but it supported the transcription of machine unreadable material, such as the old archive material of the New York Times.[8] In 2009 Google acquired re:CAPTCHA.[9] It was further used to gain more transcription data about e.g. distorted text from Google Books entries. In 2014 Google introduced *"No CAPTCHA reCAPTCHA"*.[10] The component analyses the user behavior and calculates a probability that the user is a human or a bot. If an issue or ambiguity occurs, an additional dialog is shown which implements the human computation approach in a different manner: either a distorted text task is shown or an image selection grid is displayed in which all images of a specified category should be selected. This example is illustrating the relevance of the human computation approach in fields such as research and economy.

2.2.2 Human Computation for this Thesis

Within the Backstage 2.0 project platform, especially in assignment submissions, systematic and named errors have to be recognized and aggregated. In addition, systematic errors are complex error concepts and for some errors it is hard to detect them using only a computational approach. Furthermore, there are various fields which have to be covered for a short period. For example, given a lecture in which theoretical computer science is taught. In one assignment the tasks refer to formal languages in general and in the following week the tasks refer to the field of automata theory. Thus, there is not enough information about systematic error distributions in every field that a machine learning algorithm could be trained in an adequate manner. So, human computation can be used for the process of error tagging. In this case, humans are needed to create and tag errors to gain the data needed.

[7] https://www.mturk.com/mturk/welcome

[8] http://www.nytimes.com/2011/03/29/science/29recaptcha.html

[9] https://googleblog.blogspot.de/2009/09/teaching-computers-to-read-google.html

[10] https://security.googleblog.com/2014/12/are-you-robot-introducing-no-captcha.html

Data validation and analysis The data gained through the tagging process of named errors in submissions can be analyzed and used for various improvements. For example, feedback loops can be created in which the data is used to reorder the displayed named errors. Furthermore, the named error sets themselves can be optimized. Newly created and tagged errors can be added and named errors which are hardly occurring can only be displayed on purpose. With the help of this data, named error distributions can be calculated.
For further data validation in terms of correctness, the following paragraph describes a possible approach.

Credibility and trust model The main field of application of the annotation component is the teacher review in which the teacher or tutor correct the submissions. The teaching methods are further described in section 3.2.1. However, if the annotation component is used within a peer review context, it can be assumed that judgments from different users differ in quality and correctness. To determine the quality and correctness more accurately, self-assessment of the user can be an option. However, there are various issues belonging to the self-assessment process of individuals.[7] Proving an appropriate quality of the data collected, it is necessary to develop a user model representing this trust and credibility.

Piech et al. created a model for the peer review method, which was tested in a large massive open online course (MOOC).[13] It focuses on the estimation and correction of user biases and user reliability. The model is transferred in terms of the Backstage 2.0 project platform demands and is further described in 5.2.1. Additionally, various "validation" aspects can be added. In this case a "validation" aspects refers to the equality with a fact which is assumed true. For example, the comparison with a "ground truth" can be added.[17] This is sometimes also referred to as "gold standard". The gold standard defines an answer which is marked true by e.g. the teacher. Furthermore, it can be assumed that aggregated information from an independent group is yielding accurate information in terms of correctness. This occurrence is also referred to as "the wisdom of crowds".[22]

CHAPTER 3

Conception and Implementation

In this section, Backstage 2.0 and the annotation component within the Backstage 2.0 projects platform is outlined first. The outline is followed by the explanation of the design decisions, technical as well as user-interface based. In the end, a short description of the Implementation process is given and some lessons learned are listed.

3.1 Outline

The annotation component is part of the Backstage 2.0 projects platform. The projects platform in turn is part of the Backstage 2.0 platform. In the following the structure is explained as well as the annotation component integration.

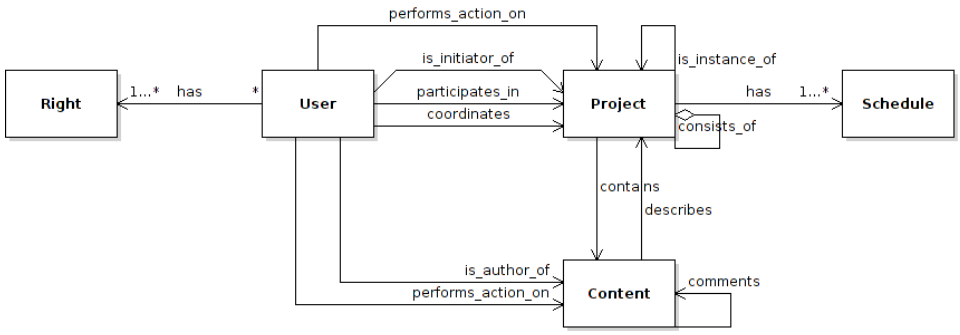

Figure 3.1: *Overview over the Backstage 2.0 project platform conception. (Visualization created by Niels Heller.)*

3.1.1 The Backstage 2.0 Platform

The component is part of the Backstage 2.0 project platform by Niels Heller (originally named crowdlearning platform). The project platform is closely working together with the newly developed Backstage 2.0 lecture platform by Sebastian Mader. The Backstage 2.0 platform focuses on lectures and provides features such as the live update of slides or

feedback on learning material. Internally, both platforms use so called units as an abstract data structure for learning material. Those units can be pulled from the Backstage 2.0 via an internal API. The project platform manages projects which consist of content, participants, assignments and a schedule. The outline of the projects platform is visualized in Figure 3.1. Courses, content and assignments can be created, uploaded, commented and tagged, events scheduled and content can be annotated (as implemented in this work). Furthermore, various statistical models are implemented to compute predictions based on information about the process of a student. It is focused on questions such as "What is the chance that the student passes a given test?" or "What is the chance that a given student is willing to hand in his or her next assignment solution?". Besides the prediction function, an analysis view is implemented to show the named-error distribution in relation to the handed in contents. The Backstage 2.0 platform is the new version of the classroom communication system Backstage.[14] The aim of backstage is to provide advantages of smaller classes, such as immediate feedback, to larger-class environments.

3.1.2 Error Recognition and Classification Component

This thesis focuses on the annotation part of the project platform. Content can be uploaded to the platform which then can be annotated. For example, a teacher added a weekly assignment to a lecture to which a student is subscribed. The student can then work on a solution and upload it to the platform. Uploaded solutions to assignments are also called *submissions* in this work. The teacher or tutor can then correct the submission by tagging named errors and systematic errors. With the tagged errors, the grade and feedback can be provided for the corrected submission. This step is provided by the annotation component: Users - in this case teachers and tutors - can create annotations, which define a context and a content. The context could be given by Cartesian coordinates in a PDF document, or by a XPath[1] position in a HTML snippet. In turn, a HTML snippet can either implement markup or code content. The content can be a reference to a named error and/or a text annotation. After this process the student can observe his or her correction process and can use the information about the error concepts to learn more about his misunderstandings and mistakes.

Another example could be the usage of the annotation component as an assignment itself. Students can be given a specific document with errors in it and the task to mark this errors. In this use-case the annotations themselves could be graded. This is an approach which could further be used as a peer review teaching model. More information on teaching models are provided in section 3.2.1.

Detailed description Besides the description of the use-cases, the component implemented in this work enables annotations on all kind of uploaded content: markup based text (including LaTeX snippets), code based content and PDFs. Annotations can either be of the type *named error* or *text*. If text is chosen, the given annotation contains only additionally text information. If named error is chosen, the annotation links to a previous created named error and could contain an additional text comment as well. Using this approach, errors can be clustered and referenced within a project, but nevertheless be individualized for a specific submission. In the beginning the intention is to provide a basic set of named errors, e.g. handed in by the professor or exercise instructor, which is then expandable by the students themselves.

[1] `https://www.w3schools.com/xml/xpath_intro.asp`

Right management within the Backstage 2.0 project platform A basic role management is implemented within the project platform - this covers *Admin*, *Teacher* and *Student*. The rights management itself differs from project to project and from task to task to provide a multi-functional approach. As described previously, the annotator component could be used in various cases. In one case, when only teachers and tutors correct, the students are not allowed to create annotations. In another case however, students have to own the right to annotate. This flexible rights management is task related on the one hand (in annotation exercises, students were given the rights needed). On the other hand it is provided through the fact that users with a higher position or more rights can modify the rights of the users below and so can satisfy any individual cases.

3.2 Conceptional Aspects

Besides the component outline there are many conceptional aspects for the human computation component. The following section tries to clarify the reasons for the implementation-design decision: Different review methods are displayed as well as the motivation for participation within the platform and for the platform itself. Furthermore, basic ethical and privacy aspects are presented and discussed and the conceptional approach in terms of design decisions is explained with the help of mock-ups.

3.2.1 Teaching Scenarios

The teaching scenarios describe how the assignment submissions are handled. The following paragraphs specify the two approaches and the solution implemented in this work.

Teacher review In the workflow, assignments are created by the teachers and solutions are uploaded by the students. Usually the teachers or tutors then correct the submissions. This is the first review method, in which only teachers and tutors can review material and tag errors: Within this thesis, this approach is named *teacher review*.

Peer review The second approach describes a review through students themselves. This approach is called *peer review* within the context of this work. In this case, the students upload their submissions and correct submissions of other students. Using this approach, the students can learn about mistakes of others and so improve themselves. However, it can be hard for one to recognize his or her systematic errors, if one isn't aware of the concept of it.[3] For this approach a kind of quality assurance is important, because we do not know whether the tagged error is correct or not. This could be implemented by a credibility model, which is theoretically explained in section 2.2.2 and further described in the discussion section 5.2.1.

Mixed approach Besides the two approaches, a mixed approach is possible: If the peer review method is chosen but tutors also correct and tag the given submissions, the tags from the tutors could be used to validate the tagging quality from the students. Within the trust model this is referred as the gold standard. The other way round, in the teacher review mode there could be a voluntary option for students to tag and create errors to support the data created by the teachers and tutors.

Design decision In this work a generic approach was chosen: The implementation was designed as an independent component, which gets an initial configuration. It can then be used in any part of the project platform. The initial configuration contains information

such as the defined named error set or whether the annotator is in view or edit mode. With this design, the annotation component could be loaded as a "normal" annotator, within a given content such as an assignment or as a simple annotation viewer.

3.2.2 Motivation for Participation

Another conceptional topic of the platform is the motivation itself. Why should a student use the platform and the component? Are there any differences to "traditional" methods in case of a one-sided teacher review?

If the platform is used as a lecture accompanying tool, the submissions and corrections are producing data in an implicit manner. This refers to the implicit type of motivation described in Quinn and Bedersons taxonomy.[17] It can be assumed that initially students do not need any further motivation as they needed for their prior work in terms of studying. This is caused by the fact that students have to upload their submissions anyway and tutors have to correct the submissions in any case.

Nevertheless, this study improves the correction process as well as the learning process. This is achieved by making the correction process more easy and the feedback for the students better. With a prior set of named errors, the correctors do not need to type as much as before. Furthermore, they have support right from the beginning which makes the correction process easier. Students on the other hand get feedback with complete error concept descriptions. Added information, such as examples, can lead to a better understanding of their misconception. Both aspects should induce a greater intrinsic motivation by improving the learning process. Because motivation is hard to quantify, we asked various questions about the concept and platform feedback in the evaluation chapter 4 to determine whether this component achieves the named goals.

Moreover, the error sets for a given exercise or topic can be displayed independently to provide a basic understanding of the topic and frequently encountered mistakes. With this feed-forward approach common errors can be prevented.[26] Furthermore, this can improve the active learning process and support the feedback given by tutors.[12]

3.2.3 Ethical and Privacy Aspects

Ethical and privacy aspects are often unaccounted for when developing applications. This study considers two parts: ethical fairness and privacy aspects.

User equality Every user should be treated in an equal manner (except for roles in rights management). It should not be considered that e.g. students with better marks in the past are likely to tag better in terms of name-errors. Furthermore, Piech et al. propose that if variables such as race, ethnicity and gender are included in the trust model and even provide better predictions, they can not be fairly used.[13]

Chance to "recover" Furthermore, it must be ensured that students are not disadvantaged by their own past performance. This can happen in situations if a credibility model is used and a student was bad at tagging in a specific topic. Thus, the conclusion can be drawn that the student would be bad at tagging in the same topic again. A solution for this problem could be a decreased tagging quality related deductions over time. This could be a solution for the issue from Piech et al.[13] that the student can not start with a so called "clean slate" on each assignment.

User privacy In the field of privacy aspects, we have to clarify the situations for the peer review mode. If students recognize the solutions of their class mates, this leads to serious privacy issues such as bullying.[20] With simple techniques, such as removing the name, issues can be avoided. Another issue is the processing of the gained error data. This should be done in an anonymous way as well. However, there should be at least a hint what is done with the data.

3.2.4 Design Decisions

In the following section the motivation for user-interface design decisions are explained.

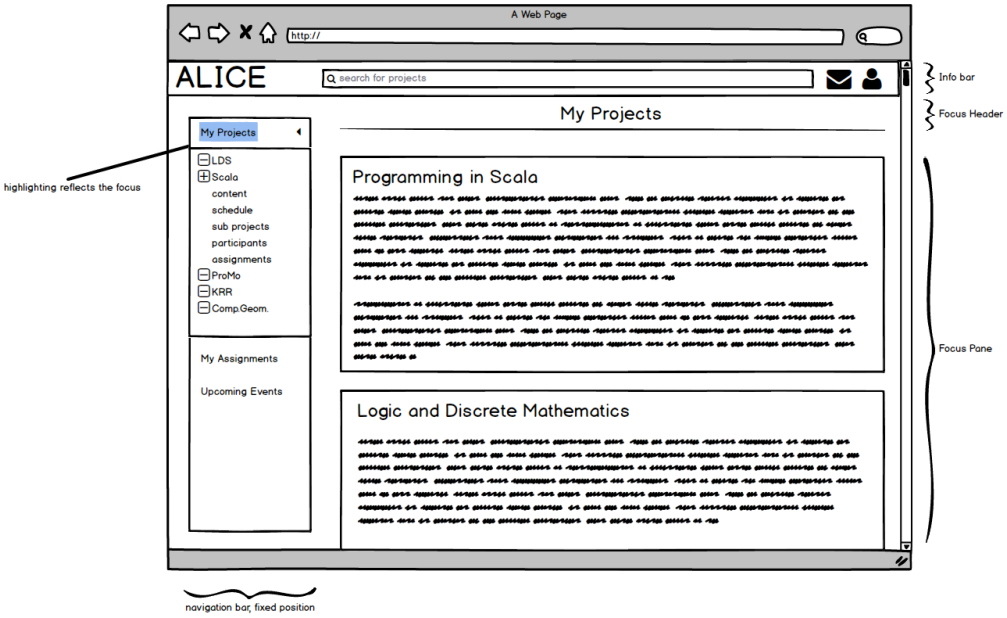

Figure 3.2: *A sample mock-up of the Backstage 2.0 project platform. (Mockup created by Niels Heller.)*

The project platform: The Figure 3.2 shows the frontend of the project platform. The base layout is structured the same within all application views: The navigation panel in the top is displaying the search bar, the notification view and the user menu. Within the main area below, the content is split into two columns: The first column contains the sidebar. The sidebar holds information about the current joined projects as well as user assignments and the users schedule. Within the projects hierarchy, the current state is marked with a highlight color. The second column contains the main-content of the current state. For example, this can be a project overview, a content view or a content annotator.

The annotation component: The Figure 3.3 shows the annotation view. The main control elements are placed on the top section and in the sidebar to make them stand out and easy to access. In the header section the annotator options can be modified between *named errors* and *textual* annotations, as well as between *highlight* and *sticky note* annotations in PDFs. The current chosen option is highlighted for an intuitive workflow. With the intention to keep the interface clean, the sidebar was subdivided into tabs. In the beginning the available named errors tab is displayed. The text of the errors is only a teaser-text to provide a

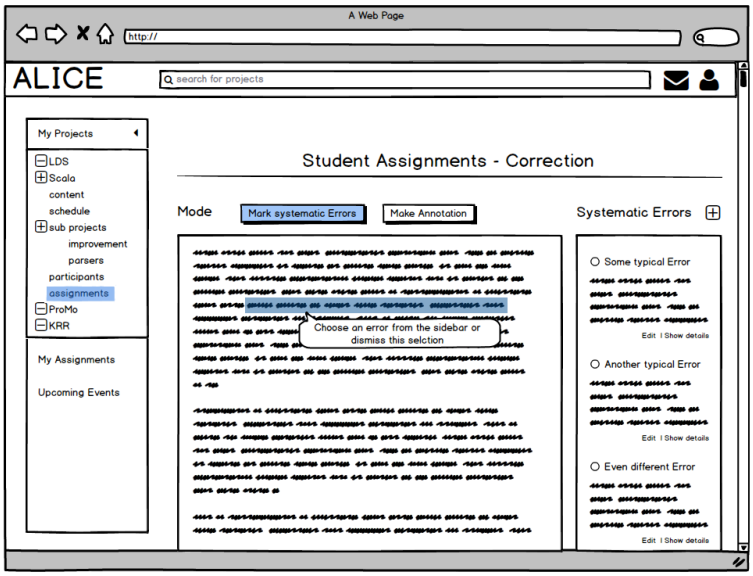

Figure 3.3: *This a an early stage view of the annotation component within the Backstage 2.0 project platform*

good readability. To show additional information and so the whole text, the button *"show more details"* can be clicked. The second tab in the sidebar holds the current state of the annotation store as well as the current annotation count. Within the display of the current annotations, the linked error title is displayed as well as the tag text. The user-interface allows the deletion of an annotation and the display of further linked named error information through links.

The component workflow is deeply integrated in the user interface. If for example the current mode is *textual* annotation, the annotation tab in the sidebar is active and the mode is switched to *named error*, the sidebar changes to the named error view listing as well to provide the needed options. Another example is the selection of an annotation within the document. In this case the sidebar is automatically updated to the annotation store. Furthermore, the selected annotation in the document is highlighted as well as the annotation detail in the sidebar. With further interface tests it was decided to auto hide the navigation panel, which is expandable again if needed. Thus the annotator itself is wider and provides a clearer view of the document. Additionally, a back button was added for easier navigation process when the navigation bar is hidden. This back button leads back to the content overview of the project.

The final state of the project platform Figure 3.4 shows a screenshot of the final implementation. The screenshot refers to the latest version at the time of this report. This is mentioned, because the platform is still in the development-process. The content overview of a given project is shown. In comparison to the mock-up, the interface was improved by various modern icons. All icon based buttons have been given tool tips which are shown if the buttons are hovered over. The tool tips provide information about the purpose of the button. This increases the usability in terms of comprehensibility and readability. Furthermore, the sidebar which is holding the navigation is resizeable.

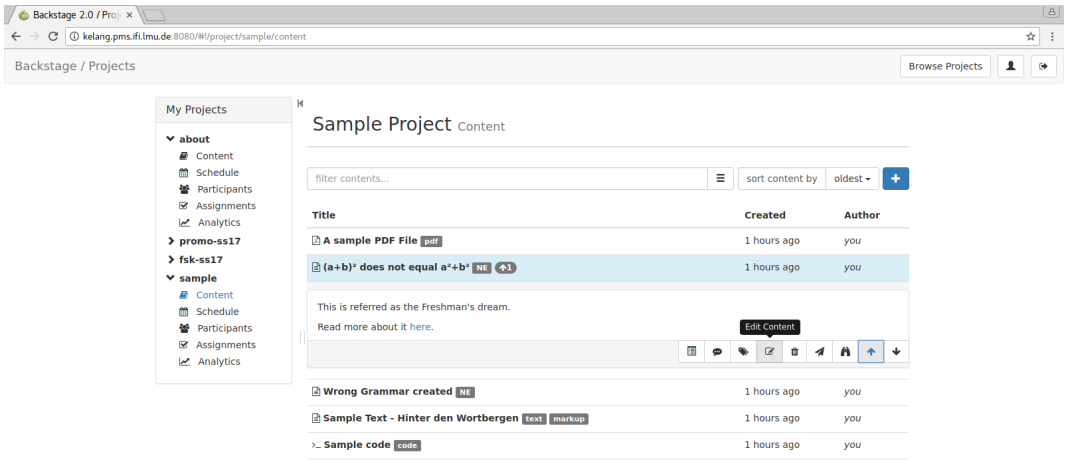

Figure 3.4: *The final design state of the project platform at the time of this report. The screenshot shows the content listing of a sample project.*

The final state of the annotator Most of the aspects of the mock-up have been implemented in the final application. A screenshot of the final application is shown in Figure 3.5. Noticeable are the changes within the annotation pop-up, which allows the entry of additional annotation information. Furthermore, the tabbed view in the sidebar provides more space within the application view itself. As described above, the auto hiding of the navigation bar is also a change allocating a wider viewport of the content.

3.3 Implementation Process

The frontend components are realized within the project platform and so are based on *Angular.js*[2] (1.6). Angular.js provides many functionalities for user interaction and dynamic content (re-)loading. The frontend logic can be capsuled into controllers and separated from the views. The framework *require.js*[3] is used to provide working dependencies even in development without a build process. To accelerate the prototyping process, the frontend framework *Bootstrap*[4] is used, which provides various basic style components, mostly CSS and JS. The server and backend is based on the *Play Framework*[5] and is implemented in Scala[6]. It handles the project platform API as well as the (Unit-) communication with the Backstage 2.0 API. As a persistent storage, the no SQL database *mongoDB*[7] is used. Model validation checks are handled by the play server-sided code.

[2] https://angularjs.org/
[3] http://requirejs.org/
[4] http://getbootstrap.com/
[5] https://www.playframework.com/
[6] http://www.scala-lang.org/
[7] https://www.mongodb.com/de

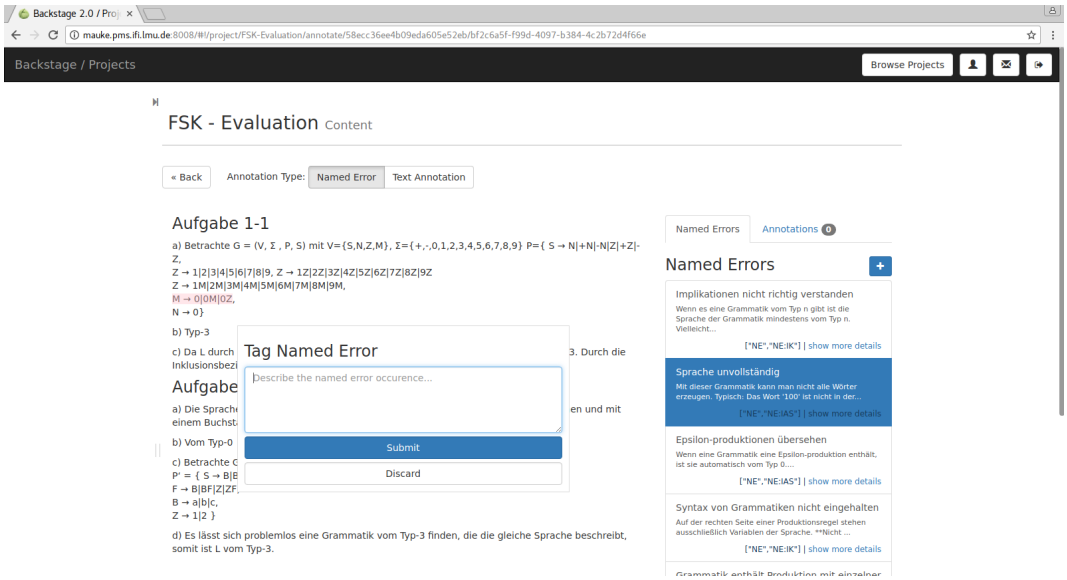

Figure 3.5: *This is a screenshot of the annotator view right from the test-server for the evaluation. The view is showing a markup annotation with named errors. An additional mistake description can be optionally added.*

Backstage 2.0 lecture platform Furthermore, the Backstage 2.0 server is based on *node.js*[8] and works with the no SQL database *RethinkDB*[9] to provide real-time support. As a frontend framework, React[10] is used. React provides various features for dynamic views like Angular.

3.3.1 Development Outline

There have been two implementation phases in this work:

Selection and annotation prototyping At first, a module was needed which enables annotations in content such as PDF and markup. For this approach, the annotation framework Annoto from Sebastian Mader was used.[9] The framework implements annotations in content material such as PDFs, images or videos. So far, the implementation did not support markup annotations. The Markup code used within the platform is rendered as pure HTML and so it was decided to create a HTML-DOM[11] annotator rather than only a markup annotator. After building a independent selection prototype, the Annoto framework was expanded with the HTML annotation functionality. At this stage, the annotator was only working within a demo page of the Annoto framework.

The Backstage 2.0 project platform component After successfully implementing the prototype, the component was integrated into the project platform. Using the modular structure of Angular, the component was implemented as directive[12]. Angular directives allow

[8] `https://nodejs.org/en/`
[9] `https://www.rethinkdb.com/`
[10] `https://facebook.github.io/react/`
[11] For further information: `https://www.w3schools.com/jsref/dom_obj_document.asp`
[12] Official documentation: `https://docs.angularjs.org/guide/directive`

an easy reuse of the implemented component in various contexts. Since Annoto provides no standard user interface implementation, a GUI orientated on the mock-up shown in Figure 3.3 was built. Thus, the annotator was built into the project platform with respect to a user-friendly annotating workflow.

3.3.2 Difficulties Encountered

The development process was delayed for multiple reasons:

- Both applications (the project platform as well as the Backstage 2.0 lecture part) have still been at an early development state while the component was developed. Due to lack of a developed server, everything had to be run locally. Multiple updates of both servers broke the database state and needed complete resets of the states.

- The development process of Backstage 2.0 lecture platform was not so important for the development of the annotation component, except for the Unit API. Nevertheless, the early stage of the project platform caused the issue, that the website itself was not quite working at the time of this report. Thus, many platform features had to be implemented simultaneously with the component implementation itself.

- Because annotating PDFs is a non trivial technical task[9], the developed annotation framework *Annoto* of Sebastian Mader was used. Various updates and refactoring steps resulted in an early stage (and undocumented) framework. It took a lot of time to get into the framework structure and to work with the framework - luckily Sebastian provided help that was needed to get the framework running.

3.3.3 Lessons learned

In this section some drawbacks and possible solutions are collected which are considered important by the author of this thesis. We want to indicate possible mistakes when working with e.g. Angular for the first time and provide hints for the developed process.

A development server One grave issue was the lack of a development server. To understand this issue, an outline is needed: The application exists for Backstage 2.0, which is ran by a node.js server and rethink db. Additionally the Scala based project platform uses sbt[13] and MongoDB. Because both platforms were developed simultaneously, often times new commits on either side broke the working state of the application. To solve this issue, a development server which is always running the newest version poses a possible solution. A development server provides multiple advantages: A stable version for testing or showing the application to possible users is beneficial. Furthermore, changes can be compared easier to changes on the local developer machine.

JavaScript JavaScript is a controversial programming language. While it is easy for beginners to start programming with a language such as JavaScript, many issues occur in further development. Points such as the missing type definition of variables or the difficult debug process create stumbling blocks while developing complex applications.
Nowadays, various approaches to solving those problems have been faced. For example,

[13] `http://www.scala-sbt.org/`

a standard called ECMAScript was introduced[14]. Furthermore, approaches such as TypeScript[15] try to add another abstraction layer. With this abstraction layer new features can be implemented and afterwards be into "normal" JavaScript.

Lessons learned in terms of use of JavaScript: One should use newer standards, which are nevertheless well supported in all common browsers. These standards support for example language constructs such as classes and promises. With these language constructs complicated workarounds no longer have to be made. If it is possible one can additionally use an abstraction layer such as TypeScript to ensure a better scaling of the application. Frontend debugging: A browser with good debugging support should be used. Google Chrome[16] or Chromium[17] have been proven to be useful for this purpose. The option to modify files while executing, setting break-points or checking out variables at a given state are valuable development tools. Extensive break-point debugging in the browser can create problems if Type-Script or other compile steps are used.

Angular The field of web-applications is developing rapidly. New frontend and backend frameworks are created, edited and improved continuously. Various web-frameworks were used during this work, for example Angular, React or Ember[18]. Because the field of application is relatively new, many approaches are developed, tried and evaluated. Newer versions, such as Angular v2, try to fix conceptual issues which occurred within the first version. In this thesis Angular v1 was used while Angular 4 was announced at the time of this report. Hence, this section is related to Angular v1.6.

Providing an example, one big issue occurs in the render cycle of Angular. In every cycle, an underlying model (provided by `$scope` objects[19]) get checked if they change: If they changed, further callbacks are executed. Changed variables can be re-rendered within the views without reloading the page itself. If a (e.g. asynchronous) computation within a controller is done, the following issue might appear: The computation in the controller is calculated before the render-step is ran or the other way round. Problems such this one can often only being solved by wrapping the computation into a `$timeout` -wrapper. This wrapper puts the computation on top of the callback queue. Thus, it changes the order in which the code is evaluated. Unwanted side effects can be solved with this approach.
In conclusion, the following hints are recommended by the author: One should choose the framework of his or her choice wisely. Every framework has its advantages and disadvantaged. They especially differ in the fields of application. One should read about different (maintained!) frameworks and then make a decision based on his or her needs. Additional, it is helpful if a framework has a big and active community. For example one reason why Angular 1.6 was chosen for the Backstage 2.0 project platform is that there are many known issues reported on StackOverflow.[20] Furthermore, if one exceeds the boundaries of a framework often workarounds are chosen which tend to be of bad coding style. One should try to find a better solution than a "dirty" workaround or e.g. even open an issue within the project if it is hosted on a platform such as GitHub[21].

[14] An overview on various JavaScript versions is provided e.g. here `https://www.w3schools.com/js/js_versions.asp`
[15] Official website: `https://www.typescriptlang.org/`
[16] `https://www.google.de/chrome/browser/desktop/`
[17] `https://www.chromium.org/getting-involved/download-chromium`
[18] `https://www.emberjs.com/`
[19] `https://docs.angularjs.org/guide/scope`
[20] `https://stackoverflow.com/`
[21] `https://github.com/`

CHAPTER 4

Evaluation

The following section describes the evaluation process. After defining the evaluation goals, design and execution are described. Finally, the gained data is analyzed and the final results are presented.

4.1 Evaluation Goals

The evaluation has two major goals: On the one hand, finding out how the application performs in terms of usability aspects. For this aspect, it is focused on the ascription of the following attributes to the project platform and component:

- Simple and intuitive use
- Consistence of the user-interface
- Evaluation of the annotation-task execution
- Level of adaption of the user workflow within the component
- Feedback provided to user, providing the current application state and process

And on the other hand, how the concept of systematic and named errors performs in everyday university life.

4.2 Study Design

After the implementation was in the beta state, the evaluation was performed. In terms of participants, four tutors of the "FSK SS 2017" lecture have been asked to volunteer. Furthermore, one student and one scientific assistant participated in the creation of named error annotations. All participants have a background in computer science.
Some initial material was needed to provide a set of assignments and submissions. With this material the application was made testable. Possible ways of providing this material are listed in the following.

- **1) A study within a real lecture:**
  Participants are intrinsically motivated to participate, which means that no further incentives are necessary. If the Backstage 2.0 project platform is stable, this is the best solution to gain bigger data-sets on named error distributions. Nevertheless changes have to be deployed in a live environment which can confuse students. If the user-interface is changed while evaluating the platform, users can get irritated. This can distort the evaluation results. In this case dissatisfaction can be created not through the implementation, but through the changes themselves.

- **2) Work with "old" data-sets from previous lectures:**
  This has the advantage that no lecture has to be held while this study is conducted. Additionally, a beta-stage of the platform would work as well and the gained data could be compared with the correction-data from the previous lecture. Students participating however need intrinsic motivation ("I want to improve the learning process in the university") or extrinsic motivation (reward for participating in the study). Peer review is not possible here because there is hardly any way to reach all previous students.

- **3) Create new user study, without direct lecture:**
  This could be done in e.g a project between semesters. The advantage is to gain new data (besides the same disadvantages as in *2)*). Additionally, the project has to be set up.

Mixed approach Furthermore, a mixed approach is possible. In all cases, motivation plays a big role and differs a lot. For example participants in *2)* probably need more extrinsic motivation than participants in *1)* because they have to use the platform nevertheless and are just producing data besides the "normal" participation in the lecture. So, a possible mixed approach could be a real lecture, which is using the same exercises as from the year before. With this approach, the newly gained data can be compared to the correction data from the past.

Ultimately, an approach based on *2)* was chosen for this study. We created a temporary production server with the beta version of the project platform running on it. Also we used old learning material and student assignment solutions from "Formale Sprachen und Komplexität, SS 16" (FSK) by Prof. Dr. Hans Jürgen Ohlbach.[1] For the evaluation, solutions as well as the assignment sheet were uploaded to the platform. The evaluation was independent from lectures since old material was used. The assignment to which the submissions belong refers to the topic of formal grammars. Participants have to correct only one part of the assignment sheet (exercise 1.1, linked in the appendix section 6.4. To analyze the usability and the feedback on the named error concepts, a questionnaire is created which is described in the next section.

4.2.1 Questionnaire

Personal questions (gender, age) and educational questions (educational background and teaching experience) were included to characterize the participators of the study. In order to evaluate the usability of the component, questions were added which are inspired by the *Questionnaire for User Interaction Satisfaction* test, short *QUIS* [2]. In the last section of the questionnaire, questions correspond to the named error concept as well as general

[1] `http://www.pms.ifi.lmu.de/`
[2] `http://lap.umd.edu/quis/`

feedback. As an example, the questionnaire contains the following questions: *Is there an improvement of the learning process? And if so, in which cases? If not, what do we learn about it?* The full questionnaire itself is available in the appendix (6.3).

Questionnaire platform For the evaluation, the website *soscisurvey.de*[3] was used. The platform allows creation of questionnaires that suite the needs of this study. The questionnaires can be tested in a so called "Pretest". If everything is working, the questionnaire can then be made public. If anything is not working as expected, the "Pretest" can be modified and tested in the new version. In the end, the gained data is available as a spreadsheet and can be downloaded as a CSV file for further processing.

4.3 Study Methodology and Results

In this section the methodology and the study results of both, the questionnaire as well as the named error study, are described.

4.3.1 Questionnaire

Three female and one male person answered in the questionnaire. The mean age was 22 years. While all four participants have a Bachelor's degree, only two have been a tutor before. All participants are STEM students. The feedback gained on the Backstage 2.0 project platform and the annotation component was predominantly positive. The gained information is explained in detail in the following.

User interaction satisfaction results A Likert scale with six items was used within the questionnaire to determine the participants opinion. An even number of items was chosen to prevent a "neutral" item. The following displayed numbers refer to the evaluation of the given scale. Providing a better readability, the results are presented in the following tables. In case a question without further attributes is used within the context of a Likert scale, the items describe the agreement. While the rating 1 describes a low agreement ("No"), the rating 6 refers to full agreement ("Yes"). The questions and ratings are numbered ascending to provide a better readability when referenced within the text.

No.	Describe your overall reaction to the application:	Avg.	Min.	Max.
1	terrible - wonderful	4,50	4	5
2	difficult - easy	3,75	3	5
3	frustrating - satisfying	4,25	3	5
4	rigid - flexible	4,25	4	5

Table 4.1: *This table displays the first rating questions of the questionnaire and their results.*

The feedback on the usability of the component is very positive. The Table 4.1 displays the first questionnaire ratings. While all questions indicate that users generally rate the application favorably, question 1, 3 and 4 were were answered better than question 2. The positive feedback on the usability is further validated by the results of the agreement questions, which are listed in table 4.2.

Almost all ratings are above the mean of 3.5 of the Likert scale. Yet there are considerable disparities. For example, question 9 was rated with the lowest score. Furthermore,

[3] https://www.soscisurvey.de/

No.	Rating question	Avg.	Min.	Max.
5	The use of terms throughout system is consistent	5,25	5	6
6	The terminology is related to the tasks	5,00	4	6
7	The position of system messages is consistent	5,25	5	6
8	The prompts for user input are clearly defined	4,50	3	5
9	The system informs about it's progress	3,25	3	4
10	The error messages are helpful	4,25	3	5
11	It was easy to learn how to operate the system	4,00	3	5
12	Performing tasks is straightforward	4,75	4	6
13	Help messages on the screen are helpful	4,75	4	5
14	The system responds in an adequate speed	4,75	3	6
15	The system seems to be reliable	5,50	5	6
16	The correction of errors is easy	4,00	2	6

Table 4.2: *The second block of rating questions and their result, describing the user agreement with an specific attribute.*

questions such as 14 or 16 show a higher disagreement in terms of the minimum and maximum range. This is validated through the rating questions 2 and 3, while the average of question 2 is clearly lower.

No.	Rating attributes	Avg.	Min.	Max.
17	Do you find the implementation helpful for tutors and teachers?	4,25	3	5
18	Do you find the implementation helpful for students?	4,25	4	5
19	Would you recommend the application to fellow students or tutors?	4,50	4	6
20	Before carrying out an exercise, knowing the named errors for this exercise would be: helpful - not helpful	1,50	1	2
21	If i made a mistake in an exercise, knowing whether - and how many - others made the same mistake would be: interesting - of no importance	2,50	1	4
22	As a tutor: If i have to correct an exercise, having a set with often made named errors would be: helpful – of no importance	1,50	1	2

Table 4.3: *The third block of rating questions and their results belonging the named error concept.*

Results referring to the named error concept The following results refer to the Table 4.3. The component for named error annotations in student assignment was rated helpful for teachers and tutors (question 17) as well as for students (question 18). The participants would recommend the application to e.g. fellow students (19). However, only one out of four participants would recommend the application unconditionally ($rating = 6$).

Open questions The participants also considered it very helpful when the named error set is provided before students have to work on the assignment (question 20). Furthermore, it appears that the participants find it helpful to have information about the named errors made by their peers (question 21). Also a named error set, which is provided to the tutors before the correction process appears to facilitate the correction process (question 22).

In the end of the questionnaire, open questions have been asked to collect feedback about the platform and the component which was not covered by the Likert scale questions. Some examples for named issues are mentioned in the following.

"A bit hard to feel into it"

"Interface could be more refined in general, e.g. provide an easier grasp of which submissions have already been evaluated and which haven't"

However, most of the feedback was very positive - for example:

"Both the selection of the NEs [named errors] and the position for the tag was very nice. The overlay that appeared when clicking on more info for the NEs was also nice. I love the ease of latex and code snippet integration."

4.3.2 Named Error Distribution

In the following section the results of the annotation process evaluation are presented. Five people participated in the annotation study, two of which are male. Three people annotated the whole 22 submissions, one person only annotated 10 submissions and one person annotated only one submission. The last one was not considered in this study. While four errors have been provided (2,3,4,5), two errors have been added from participants (1,6,7).

	Error 1	Error 2	Error 3	Error 4	Error 5	Error 6	Error 7
Submission 1	0	0	1	0	2	0	0
Submission 4	2	0	0	0	1	0	0
Submission 5	1	2	0	0	2	0	2
Submission 6	0	2	0	0	3	0	0
Submission 7	0	0	4	0	4	0	0
Submission 9	0	1	0	0	1	0	0
Submission 8	0	0	4	0	2	0	1
Submission 10	0	0	0	0	0	1	1
Submission 11	0	0	0	4	3	0	0
Submission 13	0	0	0	0	3	0	1
Submission 15	0	1	0	0	1	1	0
Submission 16	0	0	0	0	0	0	3
Submission 17	0	0	3	0	0	0	0
Submission 18	0	0	0	0	0	0	0
Submission 19	0	0	2	1	0	0	0
Submission 20	0	1	0	0	3	1	0
Submission 22	0	0	3	0	0	0	0
Submission 23	0	0	0	0	0	0	0
Submission 24	0	3	3	0	0	0	0
Submission 25	0	0	0	0	0	0	0
Submission 27	0	0	0	0	0	0	1
Submission 28	0	3	1	0	0	0	0
Occurrences	2	7	8	2	10	3	6

Table 4.4: *This table shows the accumulated error annotations per submission within the context of the evaluation. The horizontal line in the middle divides the data-set into two parts: The first part is annotated by* 4 *participants and the second part by* 3 *participants.*

Various approaches were considered for the analysis of the gained data. The main aspect which has to be analyzed is the inter-rater correlation. The inter-rater correlation should describe the agreement of the different raters in terms of named error annotations. In order to solve this issue, Cohens Kappa can be used. "Cohen's kappa coefficient is a statistic which measures inter-rater agreement for qualitative (categorical) items."[4] While

[4] https://en.wikipedia.org/wiki/Cohen%27s_kappa

	Error 1	Error 2	Error 3	Error 4	Error 5	Error 6	Error 7	
Submission 1	1	1	0.5	1	0	1	1	0.79
Submission 4	0	1	1	1	0.5	1	1	0.79
Submission 5	0.5	0	1	1	0	1	0	0.50
Submission 6	1	0	1	1	0.5	1	1	0.79
Submission 7	1	1	1	1	1	1	1	1.00
Submission 9	1	0.5	1	1	0.5	1	1	0.86
Submission 8	1	1	1	1	0	1	0.5	0.79
Submission 10	1	1	1	1	1	0.5	0.5	0.86
Submission 11	1	1	1	1	0.5	1	1	0.93
Submission 13	1	1	1	1	0.25	1	0.25	0.79
Submission 15	1	0.25	1	1	0.25	0.25	1	0.68
Submission 16	1	1	1	1	1	1	1	1.00
Submission 17	1	1	1	1	1	1	1	1.00
Submission 18	1	1	1	1	1	1	1	1.00
Submission 19	1	1	0.25	0.25	1	1	1	0.79
Submission 20	1	0.25	1	1	1	0.25	1	0.79
Submission 22	1	1	1	1	1	1	1	1.00
Submission 23	1	1	1	1	1	1	1	1.00
Submission 24	1	1	1	1	1	1	1	1.00
Submission 25	1	1	1	1	1	1	1	1.00
Submission 27	1	1	1	1	1	1	0.25	0.89
Submission 28	1	1	0.25	1	1	1	1	0.89
	0.93	0.82	0.91	0.97	0.70	0.91	0.84	Average

Table 4.5: *The transformed data-set in terms of the equality of the tagged named errors.*

Cohens-Kappa is only working for two raters, the Fleiss-Kappa approach has to be used for multiple raters.[5] At this point the issue occurred that the annotation tags are not equal to a categorization task: In a categorizing task, each item is associated (tagged) with exactly one category. In this study, the category would be a named error and an item one submission. As the results in Table 4.4 show, raters often chose to tag more than one error and sometimes chose not to tag at all. This fact conflicts with the requirement for the Fleiss-Kappa approach. For other analysis approaches, such as the chi-squared test[6], the sample size was too small. Additionally, another problem occurs: Often such statistics have a 'positive' and a 'negative case' which are treated differently. For example: If a submission s received n tags for error e, traditionally the agreement for s on e is computed as $n * (n-1)$. But if s received zero tags of e, all raters agree that e did not occur in s, while the agreement is 0.

Finally, a descriptive approach was chosen which is described in the following. In the Table 4.4, the raw data is shown. The top row lists the named errors. They are numbered to facilitate readability. The numbers are equivalent to the listed named errors in the appendix section 6.5. The following rows display the accumulated count of the specific named errors tagged per submission. However, multiple tags of a named error by one user in one submission are counted only once. Thus, the value 0 describes no tagged error. The value 4 (or 3 in the second part of the data-set) describes a error which is tagged by all participants in one submission.

Transformed data-set Providing the data in a more intuitive format, the table was transformed into a table showing agreement score of the raters. For each error and each submission, an agreement score of 0, 0.5 or 0.5 is computed as follows:

[5] https://en.wikipedia.org/wiki/Fleiss%27_kappa

[6] https://en.wikipedia.org/wiki/Chi-squared_test

- *Total agreement:* If the value equals 0 or 4 in the first part of the data-set, respectively 3 in the second part of the data-set. This case is mapped to the value 1.
- *Partial agreement tending to agreement:* If the named error is tagged, but not by all participants. This refers to the value 1 and 3 in the first part of the data-set. Furthermore, it refers to the value 1 and 2 in the second data-set. This case is mapped onto to value 0.5.
- *Partial agreement tending to equality:* This value is given if an equal number of raters chose to tag and not tag the error.

The transformed table is show in Table 4.5. Additionally, the right outer column and the bottom row contain the mean values.

4.4 Result Interpretation and Discussion

4.4.1 Evaluation Criticism

The data gained in the usability questionnaire is useful and can be further used to improve the integration of the component into the platform. However, it would be interesting and advisable to evaluate the component and the platform itself as soon as the development process is advanced further.

Furthermore, it is necessary to evaluate the named error distribution in a larger context. Interpretation could potentially be a lot more conclusive if the sample size was bigger, providing increased statistical power. For example, a lecture provides a way to obtain higher sample sizes and a complete field over various assignment topics to be measured. Nevertheless, the Backstage 2.0 platform was not ready to handle a bigger course at the time of this report.

4.4.2 Questionnaire

In conclusion, the annotation component performed well in the evaluation. The user-interface appears to be intuitive and well responding. Also, the concept of named errors was assessed well. It is considered helpful for both students and tutors/teachers. Furthermore, the feed-forward approach, in which the errors are visible to students before the assignment, seems to provide a helpful advantage.

Nevertheless, various issues due to the early development state of the platform occurred. At the time of the evaluation, features such as a working role management and notifications had not been embedded. This could have lead to the poor ratings in terms of user interface progress (see e.g. question 9). Furthermore, the way the component is built into the platform has to be improved. Especially the correction process of student submissions has to be structured more clearly. For example, it is necessary to be able to mark non-corrected assignments. As mentioned by the questionnaire participants, more colors should be applied to the platform interface. This can improve the usability when interactions (such as buttons) and the current application state (such as the current assignment state) are emphasized more strongly.

Additionally, negative feedback appears to be partly due to the hardly documented project platform. It is possible that a short video introduction to the project platform as well as the component itself could prevent a lack of information in terms of the platform

interaction. Beside the videos, a documentation could provide answers to frequently asked questions.

4.4.3 Named Error Distribution

In the following the named error distribution is characterized using a descriptive statistical approach.

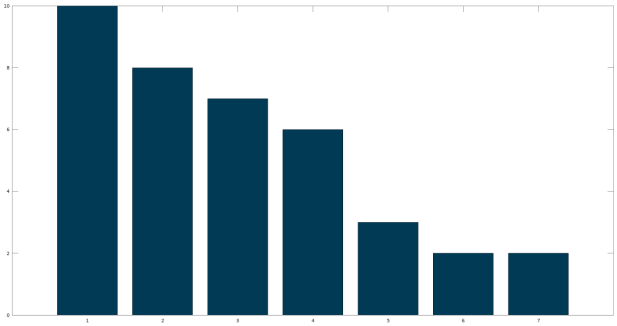

Figure 4.1: *This figure shows the distribution of the named errors tagged in the evaluation sorted by their number of occurrences.*

Occurrences In the context of this work, an occurrence is defined as at least one tag of a named error in one submission. The sum of occurrences of the individual named errors is displayed in Table 4.4. In Figure 4.1 it is noticeable that the distribution seems to a follow power law[7]. This would approve the hypothesis of Niels Heller, that the named error distribution in a specific field would assimilate a power law (personal communication, Feb, 2017). Furthermore, it can be assumed, that the distribution includes a long tail. However, it is questionable if a named error with a poor number of occurrences is still a recurring error. Nevertheless, the sample size is way too small to make a concrete statement.

The most common mistake is error 5 with 10 occurrences over all submissions. However, the agreement score is only 0.70 which is the lowest average agreement score of the tested errors. It could have various reasons that the conformity of the error tags is poor:

- The error 5 is the error “Sprache unvollständig” (engl: “Language incomplete”). This issue is very generalized and can so be tagged even if another error would fit better onto the student mistake.
- Multiple errors describe a similar mistake, e.g. error 6 (“Falsche Sprache”, engl: “Wrong language”) and error 7 (“Sprache inkorrekt”, engl: “Language incorrect”). The detailed description of the errors shows, that they are distinct. However, the errors could have been tagged without reading the description in detail.
- The agreement score within the first 10 submissions for this error is recognizable poorer than in the remaining submissions. There might be one or two students who misunderstood the error concept and so tagged falsely.

Error 1 and 4 have the smallest number of occurrences. This fact is probably due to two different reasons: Error 1 was added by a participator within the evaluation. Thus, not all

[7] https://en.wikipedia.org/wiki/Power_law

participants were shown the error while they were correcting. Furthermore, it appears that only two submissions contain this error and it is describing a more special error case. Error 4 ("Epsilon Produktion übersehen", engl. "Epsilon production rule not noticed") describes an error case in which the implication of a given circumstance was not noticed. This error appears to occur twice within the sample submission set. This could also be caused by the more special error case.

Agreement score The overall agreement score is rather high (≥ 0.70). As already mentioned above, error 5 holds the lowest agreement score. The highest agreement score belongs to error 4 (0.97). This could be due to the fact, that the error is very specific and so very clearly defined. The second highest score belongs to error 1. It seems that the high amount of zero tags of this error for every assignment leads to a high amount of rater agreement.

The agreement score of the remaining errors is between 0.82 and 0.91. It could be assumed that this values are still higher than values generated by chance. For a more specific statement, the error distribution has to be analyzed within a bigger sample size.

CHAPTER 5

Discussion

5.1 Results

Throughout this work, an annotation component was implemented to enhance the learning experience of students and the correction process of tutors and teachers. The annotation component was built as part of the new Backstage 2.0 project platform. The correction process as well as the learning psychology field were studied to improve the conceptual design of the component. Repeatedly occurring errors (including systematic errors) are mapped on the created named error concept. These named errors can be created by users and can be tagged e.g. in student submissions, thus makes correction easier and faster for tutors. Furthermore, students receive a more detailed and concise feedback. This approach is using the conceptual change model and provides a possibility for students to correct their misconceptions. The component was built using Angular.js and provides modern features such as dynamic content reloading. A generic implementation was chosen to provide reusability in various fields of application.

After the implementation, the component as well as the named error concept have been evaluated. The named error tagging process was evaluated using old learning material. The learning material contained an assignment and submissions by the lecture "Formale Sprachen und Komplexität, SS 16". After the tagging process, the usability and the named error concept have been evaluated through a questionnaire. It appears that the application performs very well in terms of its usability, hence fulfilling the evaluation goals. The annotation component appears to be simple and intuitive to use (questions 2, 11, 12) and the user interface is consistent (questions 5, 6, 7). Furthermore, the carrying of a annotation task appears to be easy and clear while the workflow is well embedded (questions 11, 12, 13, 16, 19). Also, the feedback provided to the user appears to be helpful (questions 8, 10, 13). Yet, the information about the progress of the system can be improved (question 9). Furthermore, the named error concept appears to be useful in the context of student assignment corrections (questions 19-22). Unfortunately, only a very limited number of participants was available for the evaluation of this work. Thus, the results lack generalizability and have to be evaluated in a bigger context (e.g. a lecture). Additionally, some further work was conceptualized which is provided in the following.

5.2 Further Thoughts on the Backstage 2.0 Platform and its Components

The field of technology enhanced learning and support of learning through digital approaches and devices is already present in everyday university life. Nevertheless, platforms such as the new Backstage 2.0 and the Backstage 2.0 projects platform provide means for improvement. Using new technical standards, many new features could be created which have not been possible in that way before. For example technical standards such as sockets, dynamic content reloading and flexible interaction methods through frameworks such as Angular can be used. Various features of the Backstage 2.0 projects platform have been already implemented or planned at the time of this record. However, further features are still imaginable. In the following some ideas for future implementations are described:

5.2.1 Trust Model

When using an autonomous system for e.g. peer correction, different quality assurance checks have to be implemented. The theoretical background has already been mentioned in section 2.2.2. It is further assumed that users (students, tutors and teachers) have different levels of reliability. One approach to converge to this reliability is the creation of a trust model for every user. The initial intention was to create a trust model which represents the user trust in terms of annotations. The trust model is already designed, but not implemented yet.

The idea is to use different aspects which are taken into account:

- After creation of an annotation tag, other users are able to vote for this tag. The accumulated up- and down-votes of the annotations of a user can provide a direct feedback on the annotation quality.
- If multiple users are tagging the same content, the annotations could be verified through the calculation of the tag distribution and methods such as cross-validation. For this aspects, an inter-agreement score of the users could be calculated (see section 4.3.2). The higher the score, the higher the trust.
- The trust could also be modeled with regard to the results of a previous solutions by students. It is important here to consider the ethical suggestions collected in section 3.2.3.

Trust model attributes Using the aspects mentioned above, various attributes could be considered. These attributes are based on a study by Piech et al. which created a credibility approach for large peer review systems.[13] Furthermore, these attributes are modified to fit the needs of the Backstage 2.0 project platform:

- Initial trust: This can be a trust level generated through the role of the user (professor, teacher, tutor, student). Self-assessment is not recommended for this value, because self-assessment differs from objective skill and thus credibility level of the user.[7]
- The true score of a user could be used, which is equivalent to the actual grades of a user. This could be e.g. derived from the submission uploaded in the past.
- The prior rating quality of a user, which can be calculated through correlations of old tags in comparison with other users tags.

- The bias, which describes the tendency to rate rather few or many points. This could be unnecessary in this work, because the only bias that users could have is to tag more or less in submissions. So, the bias in this case provides information about how critically a user is correcting.

5.2.2 Development of the Component

The annotation component of the Backstage 2.0 projects platform is fully functional. However, the Backstage 2.0 project platform was still in an early development stages at the time of this report. One should build new platform features, such as notifications or votes, into the context of the annotation component.

Furthermore, the annotation framework Annoto still has open issues. For example, the sticky note position moves to the top left if the sticky note is placed within a text. All open issues are minor issues which do not break the application. However, these issues should be fixed before the application is deployed.

5.2.3 Approaches of the Component Usage

The generic implementation provides various fields of application. However, at the time of this report the component was only used for the traditional teacher and tutor correction process. In the following some possible application scenarios are illustrated:

Self correction People can correct and tag their own old submissions using a sample solution. This leads to a better understanding of their own mistakes. Besides the learning effect, they can be rewarded with e.g. the chance to improve their old marks. A study by Guo and Shekoyan[5] noticed, that self reflection on own mistakes can lead to noticeable improvement of the students knowledge. As already mentioned this could lead to the issue that one is not able to recognize his or her own mistakes.[7]

Annotating as an exercise The annotation component can be used to create an assignment exercise for students. In this scenario the students were given a sample submission which has to be annotated with named errors. This approach has two major advantages: First, students learn to apply their knowledge in another context. This could support the validation of existing knowledge or even the reflection of misconceptions.[18] Second, information on inter-rater agreement can be gained.

Guessing of error concepts When students finished their solution and uploaded their submission, they can be given an optional exercise. This exercise can be a kind of quiz, in which the student can guess which (named) error will occur within the context of this assignment. Students can e.g. either gain bonus points for a good answer or collect currency units within the new Backstage 2.0 quiz component.[1]

Predictions within the component Only the complete named error set for an assignment is provided in this work. It seems meaningful, that one or more ordering and searching functionalities are added. This can provide an easier selection of the best fitting named error to a given exercise. In the following some approaches are collected:

[1]The quiz component of Backstage 2.0 is planned and not implemented yet. It should support various gamification aspects and motivate the students in longer lectures. Additionally, a scoring system is planned with a independent "currency". This should separate the learning process from the grading process.

- In the beginning the named errors can be sorted by their occurrences. Additionally, more sorting options such as *date* can be added.

- If enough data is collected, the probability that a given error will occur in a specific context can be predicted. For example the correlation of the tags of an assignment "theoretical computer science" or "formal languages" with the amount of annotations containing a specific error can be calculated. This correlation can then be used to estimate the relevance of a named error for an exercise. Furthermore, this approach can be used for the creation of a named error set on tutor and teacher side. When creating a new assignment, the topics of this assignment can be tagged.

- A search-bar can be added which is filtering the named error list. For example the search-bar can be placed above the named error listing in the sidebar. With the help of Angular, the list can be filtered while typing. This would be helpful, if the list of named errors exceeds the visible view-port of the user.

5.2.4 Platform-Extensions

While the suggestions above are related to the component itself, this sections refers to the improvement and further development of the Backstage 2.0 project platform.

Various prediction models At the time of this report, the first prediction models have been implemented. They use the data collected from student submissions to calculate the probability that e.g. one would hand in the next submission. However, there are various predictions possible. In the following, some further thoughts are collected:

- Making further use of the already made predictions about the next assignments, the data could be used to predict the probability that a student would pass a given lecture. With the help of such information the teacher receives a better feedback about the current progress of the students. With this information in mind, the teacher can then adapt the lecture to the student needs more easily.

- Furthermore, various error predictions can be made. It would be helpful in a case a teacher creates a slide-set about a new topic while having the ability to show named error distributions for the topic. If the teacher is aware of the misconceptions created by students within a topic, he or she could prevent common mistakes by prior explanations.[26]

- Additional to the data delivered by the Backstage 2.0 project platform, the data can be combined with the participation data from the Backstage 2.0 lecture platform. This approach can create new insights in terms of interdependency. Correlations can be calculated such as a correlation between the attendance at lectures and the submission contributions.

Adaptation of user affections Within the current setting of the project platform, there are no options to set up a custom configuration. It would be helpful if a user could customize some settings to his or her needs. For example, a custom dashboard or notification settings can be added. Furthermore, an adaptive content view should be possible. A work by Bures and Jelinek[2] displays an approach for adaptive web systems. In their approach the content is parsed, adapted to the user settings and readjusted by the user through a feedback loop. This approach could be applied on content types such as markup, PDF-files or code units.

5.3 Further Thoughts in General

In the following, some final thoughts are collected by the author and therefore describe the opinion of the author. They refer to the field of the named error concept as well as to advantages in the field of technology enhanced learning.

The named error concept Many advantages and improvements gained through the application of the named error concept were already mentioned. Additionally, there are various interesting aspects of this. While there are various studies for systematic errors (see [3]), there is hardly any work about systematic error distributions. This also refers to the context of higher education STEM. Furthermore, the topic is interesting for developmental research such as learning psychology. When more information about misconceptions is obtained, it should be easier to modify the teaching process from "only transferring information" to a better adjusted knowledge transfer. For example, teacher can include common misconceptions in a specific field to avoid the occurrence of those.

Technical advantages One should use more technology enhanced learning environments to a reasonable extent. The use of technical devices and software within a learning environment could significantly improve the learning process. Especially while classes are getting too big to provide a good teacher-student ratio. Platforms such as Backstage[14] and Backstage 2.0 are dealing with this issue. Nevertheless, a teacher should be aware of the way he or she is using the technical tools. It is questionable whether fourth graders need PowerPoint[2] presentations or not. However, students of every age could e.g. benefit from online homework FAQs (frequently asked questions) and access to further learning material.

Improvement of the learning process Besides the improvements through technical progress, there are various aspects in everyday university life which have got room for improvement. For example the author's subjective experience has shown that many university courses suffer from the fast growing number of students within the last years (the author studied in South Germany). Many teachers replace e.g. oral exams with written exams. Furthermore, the tutor-student ratio and so the time a tutor could spend on a single student (in terms of submission correction or explanations) decreased significantly in recent years in the opinion of the author. Therefore, it is necessary to support the teaching and correction processes by platforms such as Backstage(2). With approaches such as the aforementioned, more time could be spend on the mentoring of each student. This additional time could be spend on e.g. one-on-one tutorials (a tutor and a student). Such one-on-one tutorials can significantly improve the feedback and the learning process.[12]

[2] https://de.wikipedia.org/wiki/Microsoft_PowerPoint

CHAPTER 6

Appendix

Additional material which is relevant for the work but does not belong to the main part is appended here.

6.1 Source Code Repositories

- **cwdl-projects**
  URL: `https://gitlab.pms.ifi.lmu.de/niels/cwdl-projects`
  Revision: `d5f4ab7e6eb64be84c7bff08a8d1187efac34661` Branches: master, named-errors, evaluation
- **Annoto**
  URL: `https://gitlab.pms.ifi.lmu.de/Annoto/Annoto`
  Revision: `c5f0f8f9e8901fd0e0174418430a3f1821628ef2` Branches: project-merging-branch, requirejs-demo
- **LaTeX source files**
  URL: `https://gitlab.pms.ifi.lmu.de/Abschlussarbeiten/ma_mathias_schlenker`

6.2 Word and Phrase Explanations

To provide a stable naming convention and explanations for possible misunderstandings:

- **Backstage 2.0 projects platform: previously crowdlearning** The project platform for learning material, assignments, schedules and lectures from Niels Heller in which the annotation component is build in.
- **Backstage 2.0:** The lecture platform by Sebastian Mader, which serves Units to the projects platform.
- **Unit:** An internal data format for learning material which is generic and could be used for every kind of content from projects to uploaded material. Units were served from the Backstage2.0 Unit API.
- **Content:** (within the projects platform context) Is basically every uploaded or created material. This is reaching from markup files, over code snippets, PDFs, to internal content types such as named errors. Content is internally saved as Units.
- **Systematic error:** Are repeatedly occurring errors in STEM done independently by individuals. They are often based on misconceptions and therefore can not be resolved by traditional teaching-methods.
- **Named error:** A error categorization scheme created in this work. Named errors are a superset of systematic errors. They provide the possibility to create error schemes which are not necessarily related to systematic errors.

6.3 The Questionnaire

How old are you? (in years)
Which is your gender?
Female, Male, Not Specified
Information about your education:
In which university are you enrolled? e.g. LMU
Name your current degree? e.g. BA, MA, PhD
In which semester are you enrolled?
Have you worked as a tutor in the past?
Yes, No
Are you participating as a tutor or as a student?
As a tutor, As a student, As both
Describe your overall reaction to the application:
terrible - wonderful
difficult - easy
frustrating - satisfying
rigid - flexible
System terminology and information:
The use of terms throughout system is consistent: Not at all - Completly
The terminology is related to the tasks: Not at all - Completly
The position of system messages is constistent: Not at all - Completly
The prompts for user input are clearly defined: Not at all - Completly
The system informs about it's progress: Not at all - Completly
The error messages are helpful: Not at all - Completly
Learning-process and use of the system:
It was easy to learn how to operate the system: Not at all - Completly
Performing tasks is straightforward: Not at all - Completly
Help messages on the screen are helpful: Not at all - Completly
The system responds in an adequate speed: Not at all - Completly
The system seems to be reliable: Not at all - Completly
The correction of errors is easy: Not at all - Completly
The application tries to provide a basic toolkit of the concept of named errors to the learning workflow in universities. The following questions deal with this topic.
The named error compontent:
Do you find the implementation helpful for tutors and teachers? Not at all - Completly
Do you find the implementation helpful for students? Not at all - Completly
Would you recommend the application to fellow students or tutors? Not at all - Completly
Before carying out an exercise, knowing the named errors for this exercise would be:
helpful - of no importance for me
If i made a mistake in an exercise, knowing whether (and how many) others made the same mistake would be:
interesting - of no importance for me
(As a tutor) If i have to correct an exercise, having a set with often made named errors would be:
helpful - of no importance for me
What are the most negative aspects of the application?
What are the most positive aspects of the application?
Do you have further ideas for the use of the named errors concept?
Do you have further ideas or wishes for the platform itself?

6.4 The Exercises used in the Evaluation

Aufgabe 1-1
Grammatiken, Chomsky-Hierarchie - schriftlich bearbeiten

Sei L die Sprache der Literale, die die Programmiersprache Java für `int`–Konstanten im De- zimalsystem erlaubt. Ein solches Literal darf mit höchstens einem Vorzeichen beginnen, muss aber nicht. Danach kommt eine nichtleere Folge von Dezimalziffern, in der keine führenden Nullen erlaubt sind: 0 und $+0$ und -0 sind erlaubt, aber 00 und $+08$ und -009 nicht.
a) Geben Sie eine Grammatik $G = (V, \Sigma, P, S)$ mit $L(G) = L$ an.
b) Von welchem Typ der Chomsky-Hierarchie ist Ihre Grammatik?
c) Geben Sie für jeden Typ der Chomsky-Hierarchie an, ob

- aus Ihren obigen Lösungen folgt, dass die Sprache L von diesem Typ ist;
- aus Ihren obigen Lösungen folgt, dass die Sprache L nicht von diesem Typ ist;
- aus Ihren obigen Lösungen weder das eine noch das andere folgt.

6.5 Named Errors used in the Evaluation

Because the study was evaluated with German learning material, the named errors are written in German language.

6.5.1 Provided Named Errors

Syntax von Grammatiken nicht eingehalten: (Error 2) Auf der rechten Seite einer Produktionsregel stehen ausschließlich Variablen der Sprache. Nicht erlaubt sind: Reguläre Ausdrücke, ganze Sprachen, Automaten, ...
Tags: NE:IK

Implikationen nicht richtig verstanden: (Error 3) Wenn es eine Grammatik vom Typ n gibt ist die Sprache der Grammatik mindestens vom Typ n. Vielleicht gibt es ja noch eine *einfachere* Grammatik, mit der man die selbe Sprache erzeugen kann. (Merksatz: Der Typ der Sprache ist immer größer oder gleich dem Typ der Grammatik.) Insbesondere: Wenn man eine Typ 2 Grammatik für eine Sprache angeben kann, weiß man noch lange nicht, ob die Sprache nicht regulär ist.
Tags: NE:IK

Epsilon-produktionen übersehen: (Error 4) Wenn eine Grammatik eine Epsilon-produktion enthält, ist sie automatisch vom Typ 0.
Tags: NE:IAS

Sprache unvollständig: (Error 5) Mit dieser Grammatik kann man nicht alle Wörter erzeugen. Typisch: Das Wort '100' ist nicht in der Sprache.
Tags: NE:IAS

6.5.2 Named Errors created by Participants

Grammatik enthält Produktion mit einzelner Variable auf der rechten Seite und wird Typ 3 zugeordnet: (Error 1) Produktionsregeln der Form $A \to B$ mit $A, B \in V$ verletzen die Bedingung für reguläre Grammatiken (vgl. Folie 1-18, Schöning S. 9): *Alle rechten Seiten von Regeln bestehen entweder aus einem Terminalsymbol, oder aus einem Terminalsymbol gefolgt von einer Variablen.* Bzw.: Für alle Regeln $\omega_1 \to \omega_2 \ : \ \omega_2 \in \Sigma \cup \Sigma V$ Insbesondere solche vom Startsymbol auf ein einzelnes Nicht-Terminalsymbol werden manchmal übersehen.

Falsche Sprache erzeugt: (Error 6) Die angegebene Grammatik erzeugt eine andere Sprache

Sprache inkorrekt: (Error 7) Die Grammatik ermöglicht es, Wörter zu bilden, die eigentlich nicht in der Sprache sind. (Typischerweise erlaubt sie führende Nullen.)

Bibliography

[1] Graham Attwell et al., *Personal learning environments-the future of elearning?*, elearning papers **2** (2007), no. 1, 1–8.

[2] M Bures and Ivan Jelinek, *Description of the adaptive web system for e-learning*, Proceedings of IADIS International Conference E-Society 2004, 2004, pp. 988–991.

[3] Jere Confrey, *Chapter 1: A review of the research on student conceptions in mathematics, science, and programming*, Review of research in education **16** (1990), no. 1, 3–56.

[4] Debbie Elliott, Anthony Hartley, and Eric Atwell, *A fluency error categorization scheme to guide automated machine translation evaluation*, Conference of the Association for Machine Translation in the Americas, Springer, 2004, pp. 64–73.

[5] Wenli Guo and Vazgen Shekoyan, *Homework corrections: Improving learning by encouraging students to reflect on their own mistakes*, Proceedings of the American Society of Engineering Education (Zone 1) Conference, 2012.

[6] Leopold E Klopfer, Audrey B Champagne, and Richard F Gunstone, *Naive knowledge and science learning*, Research in Science & Technological Education **1** (1983), no. 2, 173–183.

[7] Justin Kruger and David Dunning, *Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments.*, Journal of personality and social psychology **77** (1999), no. 6, 1121.

[8] Edith Law and Luis von Ahn, *Human computation*, Synthesis Lectures on Artificial Intelligence and Machine Learning **5** (2011), no. 3, 1–121.

[9] Sebastian Mader, *An annotation framework for a collaborative learning platform*, Diplomarbeit/diploma thesis, Institute of Computer Science, LMU, Munich, 2015.

[10] Andrew Mao, Ariel D Procaccia, and Yiling Chen, *Better human computation through principled voting*, AAAI, Citeseer, 2013.

[11] Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M Voelker, and Stefan Savage, *Re: Captchas-understanding captcha-solving services in an economic context.*, USENIX Security Symposium, vol. 10, 2010, p. 3.

[12] Lisa Murtagh and Nadine Baker, *Feedback to feed forward: Student response to tutors' written comments on assignments*, Practitioner Research in Higher Education **3** (2009), no. 1, 20–28.

[13] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller, *Tuned models of peer assessment in moocs*, arXiv preprint arXiv:1307.2579 (2013).

[14] Alexander Pohl and François Bry, *Large-class teaching with backstage*, Journal of Applied Research in Higher Education (2016).

[15] George J Posner and William A Gertzog, *The clinical interview and the measurement of conceptual change*, Science Education **66** (1982), no. 2, 195–209.

[16] George J Posner, Kenneth A Strike, Peter W Hewson, and William A Gertzog, *Accommodation of a scientific conception: Toward a theory of conceptual change*, Science education **66** (1982), no. 2, 211–227.

[17] Alexander J Quinn and Benjamin B Bederson, *Human computation: a survey and taxonomy of a growing field*, Proceedings of the SIGCHI conference on human factors in computing systems, ACM, 2011, pp. 1403–1412.

[18] Justin R Read, *Children's misconceptions and conceptual change in science education*, Pridobljeno s: http://acell. chem. usyd. edu. au/Conceptual-Change. cfm (2004).

[19] Ido Roll, Vincent Aleven, Bruce M McLaren, Eunjeong Ryu, Ryan SJd Baker, and Kenneth R Koedinger, *The help tutor: does metacognitive feedback improve students' help-seeking actions, skills and learning?*, Intelligent tutoring systems, vol. 2006, Springer, 2006, pp. 360–369.

[20] Philip M Sadler and Eddie Good, *The impact of self-and peer-grading on student learning*, Educational assessment **11** (2006), no. 1, 1–31.

[21] Ruth Stavy and Baruch Berkovitz, *Cognitive conflict as a basis for teaching quantitative aspects of the concept of temperature*, Science Education **64** (1980), no. 5, 679–692.

[22] James Surowiecki, *The wisdom of crowds*, Anchor, 2005.

[23] Luis Von Ahn, *Human computation*, Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, IEEE, 2008, pp. 1–2.

[24] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford, *Captcha: Using hard ai problems for security*, International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2003, pp. 294–311.

[25] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum, *recaptcha: Human-based character recognition via web security measures*, Science **321** (2008), no. 5895, 1465–1468.

[26] Stella Vosniadou, *Capturing and modeling the process of conceptual change*, Learning and instruction **4** (1994), no. 1, 45–69.

[27] Melanie R Weaver, *Do students value feedback? student perceptions of tutors' written responses*, Assessment & Evaluation in Higher Education **31** (2006), no. 3, 379–394.

[28] Jeff Yan and Ahmad Salah El Ahmad, *A low-cost attack on a microsoft captcha*, Proceedings of the 15th ACM conference on Computer and communications security, ACM, 2008, pp. 543–554.