
Themenheft Nr. 48: Digitalisierung als Katalysator für Diversität an Hochschulen et vice versa. Herausgegeben von Natalia Reich-Stiebert, Jennifer Raimann, Carsten Thorbrügge und Len Ole Schäfer

Auf dem Weg zur skalierbaren Unterstützung des kollaborativen Schreibens in hochdiversen Fernlerngruppen

Marc Burchart¹ 

¹ FernUniversität in Hagen

Zusammenfassung

Heutzutage besitzt das Lernen in Gruppen in einer virtuellen Umgebung eine hohe Relevanz, erst recht natürlich in der Fernlehre und während der andauernden Covid-19-Pandemie. Daher sind die Hochschulen bestrebt, Software-Werkzeuge für das gemeinsame Online-Lernen und zur Unterstützung einer grossen Anzahl von gleichzeitig arbeitenden Studierenden zu entwickeln. Dafür ist es wichtig, Informationen über den laufenden Prozess des kollaborativen Arbeitens zu sammeln, insbesondere Lern- bzw. Interaktionsdaten, also z. B. Daten, die zeigen, wie die Studierenden mit den anderen Gruppenmitgliedern interagieren und ob bzw. wie sie sich mit den Lehrenden austauschen. Diese erhobenen Daten werden dann mit Verfahren der Learning-Analytics mithilfe der Software-Werkzeuge untersucht und die Ergebnisse werden zur Unterstützung der Studierenden bzw. Lehrenden verwendet. In der vorliegenden Arbeit wird eine Architektur vorgeschlagen, die das Studieren im Bereich des kollaborativen Schreibens von Hunderten von Studierenden, eingeteilt in viele Gruppen, erleichtert. Sie nutzt die Synergie der Lernumgebung Moodle und des Online-Editors Etherpad Lite. In ihr lassen sich die erforderlichen Software-Werkzeuge leicht integrieren. Ein Prototyp der Architektur sowie erste grundlegende Methoden der Datenerfassung und Learning-Analytics wurden bereits entwickelt und in einem ersten Piloteinsatz mit ca. 300 Studierenden erfolgreich getestet. Die langfristige Ziel des Projekts besteht darin, das kollaborative Schreiben mittels eigenentwickelter Software in nahezu Echtzeit zu unterstützen.

Paving the Road Towards Supporting Scalable Collaborative Writing in High-Diversity Distance Learning Groups

Abstract

Nowadays, collaborative learning in a virtual environment is highly relevant, especially in distance education and during the ongoing Covid-19 pandemic. Therefore, higher education institutions are striving to develop software tools for collaborative online learning

and to support large numbers of students working simultaneously. For this purpose, it is important to collect information about the ongoing process of the collaborative work, especially learning and interaction data, e.g. how the students interact with the other group members and whether or how they exchange information with the teachers. These collected data are then analysed with methods of learning analytics with the help of the software tools and the results are used to support learners and teachers. In this paper, an architecture is proposed that enables collaborative writing by hundreds of students divided into many groups. It uses the synergy of the learning environment Moodle and the online editor Etherpad Lite. The needed software tools can be easily integrated into it. A prototype of the architecture and first required methods of data collection and learning analytics have already been developed and successfully tested in a first pilot usage with about 300 students. The long-term goal of this project is to support collaborative writing in near real time using self-developed software.

1. Einführung in das Forschungsthema

Die jüngsten Fortschritte der Informations- und Kommunikationstechnologien haben die Entwicklung der Hochschulbildung hin zum Fern- und Online-Lernen vorangetrieben (vgl. Crow und Dabars 2015). Der Fernunterricht im 21. Jahrhundert erfordert neue Fähigkeiten, die sowohl Studierende als auch Lehrende beherrschen müssen. Eine der wichtigsten dieser Fähigkeiten ist die Zusammenarbeit in virtuellen Teams (vgl. Binkley et al. 2012), insbesondere in Form des kollaborativen Schreibens. Dessen Beherrschung gilt in den MINT-Fächern als wichtige Fähigkeit und ist folglich ein wesentliches Lernziel in der universitären Hochschulbildung (vgl. Binkley et al. 2012; Marjan Laal, Mozhgan Laal, und Kermanshahi 2012).

In der Fernlehre nehmen an den Kursen oft Hunderte von Studierenden teil, wodurch die Lehrenden vor dem Problem stehen, ihnen eine geeignete Lernumgebung bereitzustellen. Aber auch die Formierung der Gruppen und ihre Unterstützung sind grosse Herausforderungen. Diese Aufgaben müssen möglichst automatisiert von der zugrunde liegenden Lernumgebung übernommen werden und das gemeinsame Schreiben muss als computerunterstütztes kollaboratives Lernen (CSCL) erfolgen, unter Einsatz eines kollaborativen Online-Texteditors. Dabei müssen die eingesetzten Werkzeuge die Diversität der Gruppenmitglieder berücksichtigen.

Damit die Lernumgebung die o. g. Aufgaben übernehmen kann, wird eine Vielzahl an Daten benötigt, die besonders das Lern- und Interaktionsverhalten der Studierenden widerspiegeln, wie z. B. die Online-Zeiten der Studierenden, Zeiten der Inaktivität, das Scroll-Verhalten, das Verhalten im Chat, aber auch detaillierte Informationen über Textänderungen. Eine geeignete Online-Lernumgebung bietet die Möglichkeit, diese Daten zu erfassen und sie mithilfe von Methoden der Learning-Analytics auszuwerten. Die erhobenen Daten helfen u. a. dabei, Frühwarnsysteme (*early warning*

systems – EWS) zu entwickeln, die in der Lage sind, Lerngruppen zu identifizieren, die Gefahr laufen, an den gestellten Aufgaben zu scheitern. Ausserdem kann so der Lernfortschritt der Studierenden beobachtet werden.

In diesem Artikel wird die Forschungsfrage behandelt, wie man das kollaborative Schreiben in die betrachtete Lernumgebung geeignet einbindet und die genannten Daten für die benutzten Werkzeuge erhebt. Er stellt als Beitrag dazu einen Vorschlag für eine skalierbare Architektur zur Unterstützung des kollaborativen Schreibens und erste Ansätze für notwendige Methoden der Learning-Analytics vor. Es wird ein System entwickelt, das die Synergie von zwei bekannten Tools nutzt: der Lernumgebung *Moodle*¹ und des Online-Editors *Etherpad Lite*². Die Architektur erlaubt es einer grossen Anzahl von Studierenden bzw. Studierendengruppen, gleichzeitig und parallel im integrierten Editor zu arbeiten. Darauf aufbauend sollen weitere Werkzeuge zur Unterstützung des kollaborativen Schreibens entwickelt werden. Der gewählte Ansatz gibt zudem die Möglichkeit der erweiterten Datensammlung und Datenanalyse für automatisierte Unterstützungsprozesse – sowohl innerhalb der *Moodle*-Umgebung als auch innerhalb des *Etherpad*-Editors. Die anfallenden Daten geben zusätzliche Informationen über das Lernen in der Gruppe und das Verhalten der einzelnen Gruppenmitglieder. Das implementierte System bietet einen Prototyp eines realisierten analytischen Dashboards mit ersten relevanten Informationen über die aktuellen Leistungen der Gruppen und ihre Aktivitäten. Auf diesen Prototypen soll in der weiteren Entwicklung aufgebaut werden. Damit können die Lehrenden die Gruppen von Studierenden identifizieren, die Gefahr laufen, an der gestellten Aufgabe zu scheitern.

Der folgende Abschnitt 2 stellt verwandte Arbeiten vor. Abschnitt 3 bespricht daraufhin die Anforderungen an die entwickelte Architektur. In Abschnitt 4 wird zunächst die Begründung für den Einsatz des Online-Editors *Etherpad Lite* gegeben und danach in Abschnitt 5 die Systemarchitektur dargestellt. Der nachfolgende Abschnitt 6 beschreibt die Art der durch die aktuelle Lösung erhobenen Daten zur Unterstützung der Studierenden bzw. Lehrenden, während Abschnitt 7 den Prototyp des entwickelten Lehrenden-Dashboards behandelt. In Abschnitt 8 wird von dem ersten Piloteinsatz der beschriebenen Architektur berichtet. Abschnitt 9 diskutiert die bisher erzielten Ergebnisse und in Abschnitt 10 wird ein Ausblick auf zukünftige Arbeiten gegeben.

1 Vgl. <https://moodle.org/>.

2 Vgl. <https://github.com/ether/etherpad-lite>.

2. Verwandte Arbeiten

Nach Storch 2005 wirken sich kollaborative Schreibaufgaben positiv auf die Textqualität aus und führen zu kürzeren, aber besseren Texten, und zwar in Bezug auf Aufgabenstellung, grammatikalische Genauigkeit und Komplexität. Aus Software-Sicht können Lehrende zu diesem Zweck Textverarbeitungsprogramme – wie Google Docs, Microsoft Office 365, Collabora, OnlyOffice und Etherpad Lite – oder Markup-Editoren – wie Overleaf oder CodiMD – verwenden, die alle die gleichzeitige Bearbeitung von Textdokumenten in Gruppen ermöglichen. Kollaborative Programmierumgebungen wie Visual Studio Live Share ermöglichen das Stellen ähnlicher Aufgaben im Bereich des gemeinsamen Schreibens von Programmen. Um die zugrunde liegenden Textdokumente konsistent zu halten, stützen sich die genannten Software-Lösungen auf einen zentralen Server. Dies ist z. B. bei Tapia et al. 2020 der Fall. Hier wird eine Microservice-Architektur für das kollaborative Schreiben mithilfe des o. g. Online-Editors Etherpad Lite entwickelt. Die Benutzenden erhalten jeweils eine eigene Etherpad-Instanz, die ihnen als Container mithilfe der Software Docker bereitgestellt wird. Den Zugriff darauf können sie mit anderen Benutzenden teilen.

Der Einsatz eines einzigen Servers stellt bei massenhafter gleichzeitiger Bearbeitung der Dokumente einen Engpass dar. In Bezug auf Google Docs weisen Ignat et al. 2015 auf untragbare Verzögerungen (> 10 Sekunden) bei der Synchronisierung von Dokumenten hin, die von mehr als zehn Benutzenden gleichzeitig bearbeitet werden. Ebenso wurde für ein gemeinsam bearbeitetes Dokument unter *Google Docs* zur Einhaltung einer akzeptierbaren Synchronisationszeit eine Grenze von 38 gleichzeitig und schnell schreibenden Nutzenden festgestellt. Für das kollaborative Schreiben in Kursen mit vielen Studierenden sind derartige Beschränkungen inakzeptabel. Robin und Romero 2018 schlagen deshalb vor, die Last auf verzögerungstolerante Peer-to-Peer-Netzwerke zu verteilen und die Dokumente optimistisch unter den beteiligten Peers zu replizieren. Aber auch durch komplexe Replikations- und Verbindungsstrategien konnte schon für fünf simulierte Peers, die 100 Ereignisse innerhalb von 40 Sekunden auslösen, die Verzögerung nicht unter 5,85 Sekunden gesenkt werden. Obwohl algorithmische Verbesserungen möglich scheinen (vgl. Attiya et al. 2021), ist der Wartungs- und Unterstützungsaufwand für das technische Personal und die Lehrenden im Vergleich zu einer zentralisierten Lösung sehr hoch.

Aus der Perspektive von Learning-Analytics und Data-Mining stellt sich zunächst die Frage nach der individuellen Leistung der Studierenden im Hinblick auf den Prozess und das Ergebnis des kollaborativen Schreibens. Ein Grossteil der Arbeiten in diesem Bereich zielt auf Massnahmen ab, die Studierenden auf Fehler aufmerksam zu machen: Zur Verbesserung der Schreibfähigkeiten werden korrespondierende Formulierungen (vgl. Shibani et al. 2017; Shibani 2017), konstruktives Feedback (vgl. Gibson et al. 2017; Shibani 2017; Shibani et al. 2017; Shibani, Knight, und Shum 2019) und formative automatisierte Schreibanweisungen (vgl. Allen, Jacovina, und

McNamara 2016; Roscoe und McNamara 2013) eingesetzt. Die in der Schreibanalytik angewandten Methoden reichen vom *Process-Mining* und Temporalanalysen bis hin zur Computerlinguistik, z. B. *Coh-Matrix* (vgl. Graesser et al. 2004; Perin und Lauterbach 2018). In Bezug auf den Schreibprozess werden Dokumentrevisionen verglichen und grafisch dargestellt, z. B. als Analyseinstrument für individuelle Revisionsaktivitäten mithilfe der Sequenzhomologieanalyse (vgl. Winger 2014) oder als Revisionsgraphen (vgl. Shibani, Knight und Buckingham Shum 2018). In Southavilay et al. 2013 wurden Revisionskarten, Themenentwicklungsdiagramme und themenbasierte Kollaborationsnetzwerke eingesetzt, um kollaborative Schreibaktivitäten zu untersuchen. Dabei wurde *Google Docs* als Schreibumgebung verwendet, um Dokumentrevisionen und die Versionsgeschichte zu analysieren. Aufgrund der Beschränkung der *Google-Docs-API* konnten die einzelnen Änderungen des jeweiligen Dokuments allerdings nicht bestimmten Gruppenmitgliedern zugeordnet werden. Frühwarnsysteme (*EWS*) (vgl. Howard, Meehan, und Parnell 2018; Sharabiani et al. 2014; Aguiar et al. 2015; Arnold und Pistilli 2012) ermöglichen Lehrenden, rechtzeitig auf die individuellen Bedürfnisse der Studierenden einzugehen, und sollen die Zusammenarbeit der Studierenden unterstützen sowie ihr Gruppenbewusstsein durch die Wahrnehmung der Aktivitäten und Beiträge der anderen Gruppenmitglieder fördern (vgl. P. A. Kirschner, F. Kirschner und Janssen 2014; Gutwin, Greenberg und Roseman 1996).

Bei der Durchsicht der verwandten Arbeiten wurde deutlich, dass es beim kollaborativen Schreiben von Hunderten von Studierenden, eingeteilt in viele Gruppen, sowohl an geeigneten technischen Werkzeugen als auch an analytischen Unterstützungsinstrumenten für die Lehrenden fehlt, die diese Lernprozesse überwachen und steuern müssen. Nach aktuellem Kenntnisstand sind in der Literatur weder Lehrenden-Dashboards (vgl. Schwendimann et al. 2017) zur Darstellung von Gruppenaktivitäten noch Frühwarnsysteme zum kollaborativen Schreiben dokumentiert (vgl. Howard, Meehan und Parnell 2018; Sharabiani et al. 2014; Aguiar et al. 2015; Arnold und Pistilli 2012). Die Durchführung von Feldforschungen zum Thema der kollaborativen Schreibprozesse ist zudem eher auf der Ebene der getätigten Tastaturanschläge als auf der Basis von grobkörnigeren Überarbeitungen angesiedelt.

3. Anforderungen

Wie in Abschnitt 1 erwähnt, ist eine wesentliche Anforderung an die vorgeschlagene Architektur, dass das kollaborative Schreiben von Hunderten von Studierenden in vielen Gruppen sowohl durch geeignete technische Werkzeuge unterstützt als auch die Möglichkeit geboten wird, vielfältige analytische Unterstützungsinstrumente für die Lehrenden, die die Lernprozesse überwachen und steuern, einzusetzen. Dabei sollen u. a. die Metadaten der Interaktionen und des Lernens festgehalten werden, um sichtbar zu machen, wie die Gruppen interagieren.

Die hier beschriebene Architektur wurde für den Einsatz in einem Online-Kurs des Bachelor-Studiengangs Psychologie an der FernUniversität in Hagen entwickelt und dort eingesetzt. Gegenstand des Kurses ist die Erstellung von gemeinsamen Textdokumenten zum Lernen des wissenschaftlichen Schreibens. Dazu arbeiten Hunderte von Studierenden in Gruppen zu acht zusammen. Dies setzt eine hohe Skalierbarkeit sowie eine grosse Modularität der Architektur voraus, die durch zusätzliche eigenentwickelte oder externe Software erweitert werden soll. Erhobene Daten müssen aus Gründen des Datenschutzes im Besitz der Universität verbleiben. Zur Minimierung des Aufwands für die Studierenden soll das kollaborative Schreiben webbasiert über einen Online-Texteditor geschehen, sodass keine weitere Software installiert werden muss. Damit die gesamte Lernumgebung mit allen Ressourcen an einem einzigen Ort zentriert für die Studierenden zur Verfügung steht, sollen der Editor und die mit ihm erstellten Dokumente in das bestehende Lernmanagementsystem (LMS) der Hochschule bzw. des bearbeiteten Online-Kurses eingebettet werden. Für die automatisierte Unterstützung der Gruppen in Echtzeit sind niedrige Latenzen bei der Übertragung von Textänderungen besonders wichtig. Dafür sind zudem eine umfangreiche Datenerhebung, -speicherung und -verarbeitung erforderlich.

4. Werkzeuge für kollaboratives Schreiben

Eine der o. g. wesentlichen Anforderungen des entwickelten Systems ist es, skalierbar und modular das gemeinsame Schreiben einer grossen Anzahl von Studierendengruppen in der Fernlehre zu ermöglichen. Unterabschnitt 4.1 befasst sich deshalb mit der Auswahl eines geeigneten kollaborativen Online-Texteditors. In Unterabschnitt 4.2 wird danach die Notwendigkeit der Nutzung mehrerer Instanzen des gewählten Editors aufgezeigt, um die gewünschte hohe Skalierbarkeit zu erreichen.

4.1 Auswahl eines geeigneten kollaborativen Online-Editors

Zu Beginn der Entwicklung wurden die wichtigsten Anforderungen an den zu nutzenden kollaborativen Editor bestimmt. Dazu zählte insbesondere – wie in Abschnitt 2 erwähnt –, dass der Editor als Webanwendung zur Verfügung steht und somit in das bestehende Lernmanagementsystem eingebunden werden kann. Weiterhin wurde die Entscheidung getroffen, einen möglichst einfach zu bedienenden Schreibeditor (Textverarbeitungsprogramm) ohne Mark-up zu verwenden, um den Studierenden die ggf. notwendige Einarbeitung zu ersparen und so einen einfachen und schnellen Einstieg in die Dokumenterstellung zu ermöglichen. Aus diesen Gründen wurden nur die in Tabelle 1 aufgeführten Editoren untersucht. Die gestellten Anforderungen sind in der ersten Spalte aufgeführt. Da teilweise keine persönlichen Erfahrungen mit den Editoren vorlagen, sind die Ergebnisse der Tabelle aus den Webseiten der Anbieter extrahiert und zwangsläufig zum Teil subjektiv.

Alle untersuchten Editoren bieten als universell einsetzbare Werkzeuge keine speziellen Verfahren für Learning-Analytics an und erheben deshalb nur die benötigten Daten zum Schreibprozess. Daher waren die folgenden Anforderungen der vorgestellten Einsatzumgebung für die Auswahl entscheidend: freie Nutzbarkeit, open source, Datenautonomie sowie die einfache Möglichkeit der Erweiterung durch eigene, aber auch extern entwickelte Plug-ins. Aus diesen Gründen wurde der Online-Editor *Etherpad Lite*³ ausgewählt. Er ist durch seine benutzerfreundliche Oberfläche für die Studierenden leicht zu bedienen. Im Editor hat jedes Mitglied jederzeit die Möglichkeit, den Bearbeitungsfortschritt und die historische Entwicklung des Dokuments zu sehen. Ausserdem stellt er jeder Studierendengruppe einen eigenen Chat zur Verfügung. Wie gefordert, können die Kernfunktionen von *Etherpad Lite* durch eine Vielzahl von frei verfügbaren, aber auch eigenentwickelten Plug-ins erweitert werden.

Anforderungen	Etherpad Lite	Google Docs	OnlyOffice	Collabora Online
frei nutzbar	ja	ja	≤ 20 simultane Verbindungen	≤ 20 simultane Verbindungen
open source	ja	nein	teilw. proprietäre Lizenzierung	ja
keine Limitierung bei vielen gleichzeitig Schreibenden	Parallelisierung notwendig	beschränkt	bei kostenpflichtiger Version	bei kostenpflichtiger Version
Datenautonomie	ja	nein	ja	ja
Erweiterung durch Plug-ins	ja	eingeschränkt	ja	eingeschränkt
grundlegende Workspace-Awareness	ja	ja	ja	ja
Integration in Moodle	ja	nein	ja	ja
nutzbare Datenerhebung	unzureichend	nein	unzureichend	unzureichend

Tab. 1: Vergleich ausgewählter Editoren

Etherpad Lite ist zwar für eine grosse Anzahl von gleichzeitig aktiven Benutzenden konzipiert, eine einzelne arbeitende Instanz des kollaboratives Editors ist jedoch im Worst Case nur für eine relativ kleine Anzahl ausreichend, d. h., durch eine grosse Anzahl von Benutzenden mit grosser Eingabetätigkeit wird sie u. U. überlastet. Diese Überlastung führt zu Leistungseinbussen in Form von inakzeptablen Verzögerungen bei der Synchronisierung der Darstellung von eingegebenen Textänderungen und erschwert den Einsatz der o. g. Analyseverfahren (siehe den folgenden Unterabschnitt

3 Vgl. <https://github.com/ether/etherpad-lite>.

4.2 und vgl. Dang und Ignat 2016). Daher wird offiziell die Verwendung des *Etherpad Scale Calculator*⁴ empfohlen, der die minimale Anzahl von *Etherpad*-Instanzen für eine gegebene Anzahl von parallel arbeitenden Benutzenden bestimmt.

Dem Kalkulator nach benötigen zum Beispiel 200 Gruppen von jeweils vier aktiv schreibenden Studierenden (mit Eingaben von je 40 Wörtern pro Minute) zwei Instanzen. Da die vorgeschlagene Architektur allerdings darauf abzielt, Kurse mit Hunderten von Studierenden in einer realen Lernumgebung in Echtzeit zu unterstützen, ist es erforderlich, mehrere *Etherpad*-Instanzen parallel zu verwenden.

4.2 *Etherpad*-Lasttests

Der *Etherpad Scale Calculator* betrachtet, wie viele Nachrichten eine *Etherpad*-Instanz pro Sekunde verarbeiten kann, allerdings nicht die Verzögerungen, die bei der Verarbeitung im Server und bei der Ausgabe der Nachrichten zu den einzelnen Nutzenden auftreten können. Daher wurden mit einer eigenentwickelten Testsoftware Analysen durchgeführt, um die Abhängigkeit zwischen Last und resultierender Verzögerung (Latenz) zu ermitteln und somit die benötigte Anzahl an *Etherpad*-Instanzen zu bestimmen. Die Tests wurden komplett auf einem einzelnen leistungsfähigen Computer durchgeführt, ohne dass sich ein Erreichen seiner vollen Systemauslastung zeigte. Da die Tests nur für die Bestimmung der benötigten Anzahl von *Etherpad*-Instanzen eingesetzt wurden, konnten durch den Einsatz auf einem einzigen Server verfälschende Netzwerkverzögerungen vermieden werden, die in der realen Umgebung allerdings berücksichtigt werden müssen. Folglich werden die dort auftretenden Verzögerungen eher höher ausfallen. Um die Validität der Tests zu gewährleisten, wurden als Ergänzung weitere Daten erfasst, wie z. B. die Anzahl der gesendeten und empfangenen Wörter.

Die Analyse lieferte die Daten für die folgende Abbildung 1. Die sich ergebenden Verzögerungszeiten sind in Abhängigkeit von der Anzahl der Gruppen mit jeweils acht Teilnehmenden dargestellt, was den Anwendungsfall eines realen Kurses in der untersuchten Einsatzumgebung (s. Abschnitt 3) wiedergibt.

Es wurde davon ausgegangen, dass im ungünstigsten Fall alle Gruppen mit einer einzigen *Etherpad*-Instanz arbeiten und dass die Nutzenden jeweils 60 Wörter pro Minute erzeugen, d. h. ein Wort pro Sekunde. Dies entspricht der Leistung einer schnellen Schreibkraft (vgl. Dhakal et al. 2018). Insgesamt wurden Tests mit bis zu 125 Gruppen durchgeführt. Da bereits bei zehn Gruppen hohe Verzögerungszeiten auftraten, wurde die Darstellung in der Abbildung auf zehn Gruppen begrenzt, um die einzelnen Ergebnisse besser beurteilen zu können. Dargestellt wird der Zusammenhang zwischen der maximalen Verzögerungszeit *wtt_max* und der durchschnittlichen Verzögerungszeit *wtt_avg*, also den Zeiten, die auftreten, bis die eingegebenen

4 Vgl. <https://scale.etherpad.org/>.

Wörter an alle anderen Mitglieder der jeweiligen Gruppe übertragen wurden. Auf der Client-Seite werden standardmässig alle Eingaben und Änderungen am Text in einem sogenannten *Changeset*⁵ gesammelt, der in einer einzigen Nachricht an den Server gesendet wird, sobald der Server die zuvor gesendete Nachricht an den Client quittiert hat. Die durchschnittliche Wartezeit für die Quittierung *mtt_avg* ist in den oben dargestellten Verzögerungen enthalten.

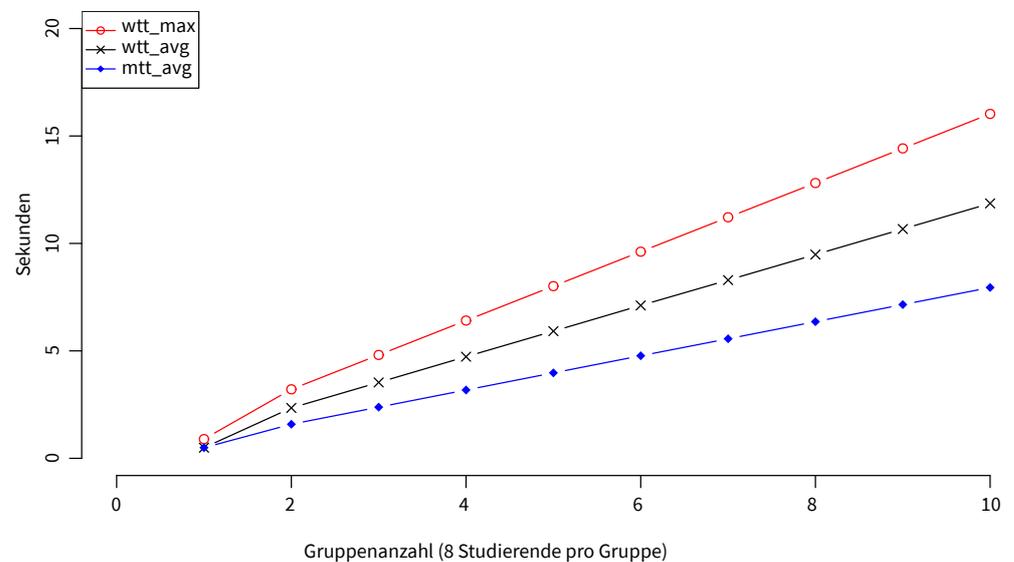


Abb. 1: Verzögerungszeiten in Abhängigkeit von der Anzahl der Gruppen.

Abb. 1 verdeutlicht, dass selbst bei fünf Gruppen, also insgesamt 40 Teilnehmenden, mit einer maximalen Verzögerung von etwa 8 Sekunden und einer durchschnittlichen Verzögerung von etwa 6 Sekunden gerechnet werden muss. Diese Zeiten liegen im unteren Bereich der bei der Verwendung von *Google Docs* ermittelten Verzögerungen, wie sie in Ignat et al. 2015 bereits für eine deutlich geringere Last angegeben sind. Die zugrunde liegende Ereignisschleife (*event loop*) der Node.js-Laufzeitumgebung, welche von der *Etherpad*-Software im Single-Thread-Modus verwendet wird, kann hierbei der Engpass sein (vgl. Zhu et al. 2015). Die Ergebnisse bestätigen das Resultat des oben erwähnten *Etherpad Scale Calculators*, der – wie erwähnt – für eine grosse Anzahl von Gruppen den Einsatz mehrerer *Etherpad*-Instanzen vorschlägt. Sie belegen die Notwendigkeit für eine skalierbare Architektur, gerade auch in Bezug auf die o. g. geplante Entwicklung von in Echtzeit unterstützenden Werkzeugen. Eine Realisierung mit mehreren Instanzen profitiert zudem von der zugrunde liegenden Multi-Core-Architektur moderner Server.

5 Vgl. https://github.com/ether/etherpad-lite/blob/develop/doc/api/changeset_library.md.

5. Systemarchitektur

In diesem Abschnitt wird die implementierte Systemarchitektur beschrieben, die das kollaborative Schreiben für eine grosse Anzahl von Studierendengruppen in der universitären Fernlehre ermöglichen soll. Sie zielt besonders darauf ab, die Anforderung aus Abschnitt 3 und 4 zu erfüllen, d. h., skalierbar mehrere Etherpad-Instanzen zu betreiben. Wie in Abbildung 2 dargestellt, besteht die Architektur aus zwei zentralen Teilsystemen: dem Lernmanagementsystem (LMS) und einer Umgebung für das kollaborative Schreiben (*collaborative writing environment* – CWE).

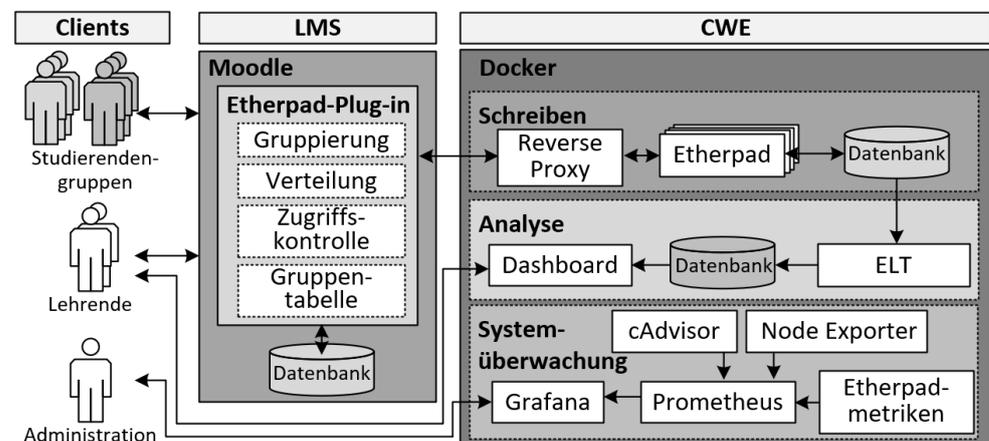


Abb. 2: Aktuelle Systemarchitektur.

Jedes Teilsystem ist dabei auf einem separaten Server installiert, um sicherzustellen, dass die Funktionalität des LMS nicht durch die erhöhten Systemanforderungen des CWE beeinträchtigt wird et vice versa. Das LMS und das CWE kommunizieren mithilfe eines gesicherten HTTPS-Kanals über das Internet miteinander. Gleiches gilt auch für die Kommunikation der beteiligten Studierenden (*clients*) mit den einzelnen Instanzen.

Als LMS wird in dieser Arbeit – wie oben erwähnt – Moodle verwendet, das in erster Linie zur Strukturierung von Lernaktivitäten und Lernquellen für die Studierendengruppen im Online-Kurs verwendet wird. Innerhalb dieses Kurses werden zudem die Studierenden durch die Lehrenden über die in Moodle integrierten Gruppenfunktionalitäten in vordefinierte Gruppen eingeteilt.

Das CWE-System nutzt die *Docker*-Technologie⁶, um mehrere *Etherpad*-Instanzen – zusammen mit den zusätzlichen Kommunikations-, Datenerfassungs-, Analyse- und Beobachtungsfunktionalitäten – modular zu betreiben. Die dabei genutzte Aufspaltung in Container erlaubt eine stärkere Skalierbarkeit und bessere Wartbarkeit als bei einer monolithischen Softwarearchitektur. Alle Teilsysteme sind für die Lehrenden bzw. die Administration zugänglich, um die notwendigen Analysen durchführen

6 Vgl. <https://www.docker.com/>.

zu können, Wartungsarbeiten vorzunehmen oder eventuelle Probleme im System zu lösen. Die Teilsysteme und ihre Komponenten werden in den folgenden Unterabschnitten genauer beschrieben.

5.1 Lernmanagementsystem (LMS)

Studierende einer Gruppe erhalten Zugriff auf ein gemeinsames Textdokument (*pad*), das von *Etherpad Lite* bereitgestellt wird. Ein bestehendes *Moodle Activity Plug-in*⁷ namens *mod_etherpad* wurde integriert und weiterentwickelt, um die – in Abschnitt 3 genannten – speziellen Anforderungen zu erfüllen. Dazu wurde das Plug-in um folgende Komponenten erweitert:

- *Gruppierung*: Die Zuordnung der Studierenden zu den Lerngruppen erfolgt durch die Lehrenden mithilfe von *Moodle*. Dabei können sogenannte Gruppierungen oder *Moodle-Gruppen*, d. h. Auswahlen und Kombinationen aus mehreren Studierendengruppen, definiert werden. Das Plug-in erlaubt es, eine dieser Gruppierungen für jede gestellte kollaborative Schreibaufgabe auszuwählen. Es stellt dann automatisch jeder Gruppe der Gruppierung ein eigenes Arbeitsdokument, ein sog. *Pad*, für die jeweilige Aufgabe zur Verfügung.
- *Verteilung*: Das für die vorliegende Arbeit erweiterte *Etherpad*-Plug-in verteilt die Gruppen statisch möglichst gleichmässig auf die *Etherpad*-Instanzen.
- *Zugriffskontrolle*: Die Studierenden erhalten innerhalb von *Moodle* Zugriff auf den eingebetteten *Etherpad*-Editor und das der Aufgabe jeweils zugeordnete Dokument. Durch die Aufteilung der Gruppenaktivitäten auf mehrere *Etherpad*-Instanzen, die sich auf dem CWE-Server befinden, haben nur die Gruppenmitglieder exklusiven Zugriff auf ihr Pad. Daher musste in der entwickelten Software die automatische Anmeldung der Benutzenden angepasst werden.
- *Gruppentabelle*: Die für die Lehrenden realisierte *Gruppentabelle* enthält Verknüpfungen (*links*), die es ihnen ermöglichen, über *Etherpad Lite* sicher auf die jeweiligen Arbeitsdokumente, die Pads, zuzugreifen. Die Tabelle enthält auch die notwendigen gruppenbezogenen Informationen über alle Gruppen und ihre Pads (z. B. Gruppen-ID, Gruppenname, erster Zugriff, letzte Änderung, *Etherpad*-Gruppen-ID, *Etherpad*-Instanz, Studierendennamen usw.).

5.2 Kollaborative Schreibumgebung (CWE)

Die kollaborative Schreibumgebung besteht aus drei Kernkomponenten:

- *Schreiben*: Diese Komponente stellt jeder Gruppe *Etherpad Lite* für das kollaborative Schreiben als gemeinsamen Texteditor zur Verfügung – einschliesslich der Sammlung von Rohdaten.

⁷ Vgl. https://moodle.org/plugins/mod_etherpadlite.

- *Analyse*: Diese Komponente dient zur Analyse der gewonnenen Daten und zur Visualisierung von gruppenspezifischen Kennzahlen.
- *Systemüberwachung*: Hierbei handelt es sich um eine Administrationskomponente zur Überwachung der Systemressourcen (*monitoring*).

Jede dieser Kernkomponenten besteht aus mehreren Unterkomponenten, die jeweils spezifische Funktionalitäten für den gesamten Schreib-/Analyseprozess bereitstellen. Sie sind mittels der o. g. *Docker*-Technologie als Container implementiert. Die durch *Docker* garantierte Container-Isolation bietet dem System die erforderliche Modularität und Zuverlässigkeit sowie eine bessere Wartbarkeit. Wie verlangt, kann das gesamte Kernsystem leicht erweitert und den spezifischen Anforderungen entsprechend skaliert werden.

Die Komponente *Schreiben* stellt die Kernfunktionalität des CWE bereit. Sie bietet den Gruppen von Studierenden die jeweilige *Etherpad*-Arbeitsumgebung und sammelt die Rohdaten über die Nutzung dieser Komponente. Die Herausforderung für das CWE-System ist die Skalierung mit einer wachsenden Anzahl von Studierendengruppen. Dazu enthält die Komponente Schreiben die folgenden Unterkomponenten:

- *Reverse Proxy*: Diese Unterkomponente – realisiert durch die Software *Nginx*⁸ – ist für das Routing der *Moodle*-Plug-in-Anfragen zur entsprechenden *Etherpad*-Instanz jeder Gruppe verantwortlich.
- *Etherpad*: Die Unterkomponente besteht aus mehreren Docker-Containern, die identische Installationen des kollaborativen Online-Editors *Etherpad Lite* enthalten. Dazu wird das offizielle *Etherpad*-Image⁹ verwendet, das um benötigte Plug-ins und um ein selbstentwickeltes Tracking-Plug-in erweitert wurde (siehe Abschnitt 6).
- *Datenbank*: Diese Unterkomponente stellt eine von allen *Etherpad*-Instanzen gemeinsam genutzte Datenbank bereit, die mithilfe der *PostgreSQL*-Software¹⁰ implementiert wurde.

Die *Analyse-Komponente* ist für die Speicherung der übertragenen Rohdaten der *Etherpad*-Nutzung, ihre Analyse und die Weitergabe der Ergebnisse an die Lehrenden verantwortlich. Die Komponente besteht aus drei Unterkomponenten:

- *Extrahieren, Laden, Transformieren (ELT)*: Diese Unterkomponente übernimmt die Aufgabe des ELT-Prozesses (vgl. Gour Vishal et al. 2010), der für das Extrahieren der Informationen aus der Datenbank der Schreibkomponente, das Laden der Daten in die Analytikdatenbank und deren Vorverarbeitung (Transformation)

8 Vgl. <https://www.nginx.com/>.

9 Vgl. <https://github.com/ether/etherpad-lite>.

10 Vgl. <https://www.postgresql.org/>.

verantwortlich ist, die vor der Durchführung der Analyse erforderlich ist. Sie kombiniert mehrere technische Lösungen: Für das Extrahieren und Laden der Daten wurde die Technologie *Pipelinewise*¹¹ verwendet, die auf der Software *Singer.io*¹² aufbaut. Der gesamte Prozess ist so realisiert, dass er einmal pro Tag ausgeführt wird; das Design ermöglicht es jedoch, diesen Prozess bei Bedarf auch häufiger ausführen zu lassen.

- *Datenbank*: Die relationale PostgreSQL-Datenbank speichert sowohl die übertragenen Rohdaten als auch deren vorverarbeitete Version (s. o.). Sie ist für die Beschleunigung der folgenden Analyse optimiert.
- *Analyse-Dashboard*: Diese Unterkomponente wird zur Präsentation der Ergebnisse für die Lehrenden verwendet. Sie extrahiert die Daten aus der Analysedatenbank und führt die Berechnungen sowie die Modellierungen durch. Sie basiert auf den *Technologien R*¹³ und *Shiny*¹⁴. Das Dashboard steht den Lehrenden über einen gesicherten Zugang zur Verfügung, der durch den sog. *Shiny-Proxy-Server*¹⁵ bereitgestellt wird.

Die letzte Komponente des CWE-Systems ist die *System-Überwachung (monitoring)*, die zur Überwachung und Aufzeichnung des Verbrauchs an Computerressourcen dient, um systemweite Engpässe bei der Speichernutzung, der CPU-Zeit, der Netzwerkauslastung usw. zu erkennen. Das System benachrichtigt die Administration des CWE-Systems sofort über erkannte Engpässe, um ihr eine schnelle Unterstützung und Lösung bei solchen Vorfällen zu ermöglichen. Sie bietet der Administration auch einen Überblick über die Anzahl der momentan aktiven Studierenden auf jeder *Etherpad*-Instanz. Darüber hinaus sammelt es die langfristigen Auslastungsdaten für die weitere Analyse und Planung. Die Komponente besteht aus zwei Unterkomponenten:

- *Prometheus*: Diese Unterkomponente bietet einen sog. *Daemon* zur Aufzeichnung und Ausgabe von Echtzeitwarnungen über Systemressourcen und deren Verbrauch. Es sammelt statistische Daten (Metriken) über die *Etherpad*-Instanzen mithilfe des *Prometheus-Plug-ins*¹⁶, über die *Docker-Container* und deren Ressourcenverbrauch mithilfe der *cAdvisor-Software*¹⁷ sowie über das *Host-System* mit *Node Exporter*¹⁸.

11 Vgl. <https://github.com/transferwise/pipelinewise>.

12 Vgl. <https://www.singer.io>.

13 Vgl. <https://www.r-project.org>.

14 Vgl. <https://shiny.rstudio.com>.

15 Vgl. <https://www.shinyproxy.io>.

16 Vgl. https://www.npmjs.com/package/ep_prometheus.

17 Vgl. <https://github.com/google/cadvisor>.

18 Vgl. https://github.com/prometheus/node_exporter.

- *Grafana*: Dies ist ein Administrations-Dashboard für die von Prometheus gesammelten Langzeitstatistiken. Der Zugriff auf das Dashboard ist über einen SSH-Tunnel auf die Systemadministration beschränkt.

6. Datenerfassung und -verarbeitung

Die Erfassung und Verarbeitung der Daten aus den Systemkomponenten stellen wesentliche Anforderungen an die Architektur dar (siehe Abschnitt 3). Diese Daten werden einerseits für die Unterstützung bei der Einteilung der Studierenden in Gruppen und der Gruppen beim kollaborativen Schreiben benötigt. Andererseits ermöglichen sie die Entwicklung von Warnsystemen, die Lerngruppen identifizieren können, die Gefahr laufen, an den gestellten Aufgaben zu scheitern. Wichtig hierbei ist, dass die im CWE anfallenden Daten nur mit randomisierten Pseudonymen (Author-ID) der Studierenden gespeichert werden, sodass die Forschenden selbst keinen Zugriff auf personenspezifische Daten (bis auf die Vornamen der Studierenden innerhalb des Chats) haben. Da die Frühwarnsysteme (EWS) ggf. den Lehrenden die Möglichkeit der Kontaktaufnahme zu gefährdeten Gruppen bzw. Gruppenmitgliedern bieten müssen, wird eine Zuweisung zwischen *Moodle*-Konto und Author-ID gespeichert. Auf personenbezogenen Daten haben die Forschenden dabei keinen Zugriff, die Lehrenden nur einen indirekten über das EWS.

Etherpad Lite erfasst standardmässig die Änderungen, die die Studierenden an den bearbeiteten Dokumenten vornehmen. Die gespeicherten Daten beinhalten dabei den Zeitpunkt und die Textmodifikationen in Form von Ergänzungen, Löschungen oder Ersetzungen (vgl. M. Liu, Pardo, und L. Liu 2017). Durch das Sammeln dieser Daten kann die Entwicklung der Dokumente – einschliesslich der individuellen Beiträge der einzelnen Studierenden – verfolgt werden. Neben dem resultierenden Text und den Daten des Schreibprozesses wird auch die Chatkommunikation zwischen den Gruppenmitgliedern aufgezeichnet, unter Einschluss der entsprechenden Vornamen der Autor:innen, Nachrichteninhalte und Zeitstempel.

Ein selbst entwickeltes Tracking-Plug-in *ep_tracking* erweitert die eben genannte Datensammlung von *Etherpad Lite* durch Funktionen, die einen detaillierten Einblick in das Verhalten der einzelnen Gruppenmitglieder ermöglichen und dadurch Rückschlüsse auf die Interaktionen zwischen den Studierenden und ihr Lernverhalten zulassen. Das Plug-in sammelt die folgenden zusätzlichen Daten:

- Verbindungszeiten der einzelnen Studierenden mit dem *Etherpad*-Editor im CWE (und damit mit dem jeweiligen Pad der Gruppe),
- Zeiten der Inaktivität, in der die Browserregisterkarte (*tab*) mit dem Online-Editor *Etherpad Lite* nicht zur Bearbeitung zur Verfügung steht, also nicht angewählt ist,
- Zeitpunkte, zu denen die Studierenden den *Etherpad*-Chat öffnen oder schliessen,
- Scrollpositionen der Studierenden im Dokument,
- Scrollpositionen der Studierenden im Chat.

Die gesammelten Informationen spiegeln die Aktivität der einzelnen Studierenden in der Arbeitsumgebung wider und können als Repräsentation ihrer eigenen kollaborativen Arbeit bzw. der der gesamten Gruppe dienen.

7. Lehrenden-Dashboard

Eines der Hauptziele des entwickelten Systems besteht darin, den Lehrenden einen tieferen Einblick in die Gruppenarbeiten – insbesondere das kollaborative Schreiben – zu geben. Dazu wurde zunächst eine prototypische Version eines Dashboards entwickelt, welches die Realisierbarkeit des Konzepts innerhalb der hier beschriebenen Architektur zeigt. Sein modularer Aufbau ist in Abb. 3 skizziert. Es liefert ihnen die wesentlichen Informationen über die kollaborativen Gruppen und ihre Interaktionen in den entsprechenden *Etherpad-Lite*-Schreibumgebungen.

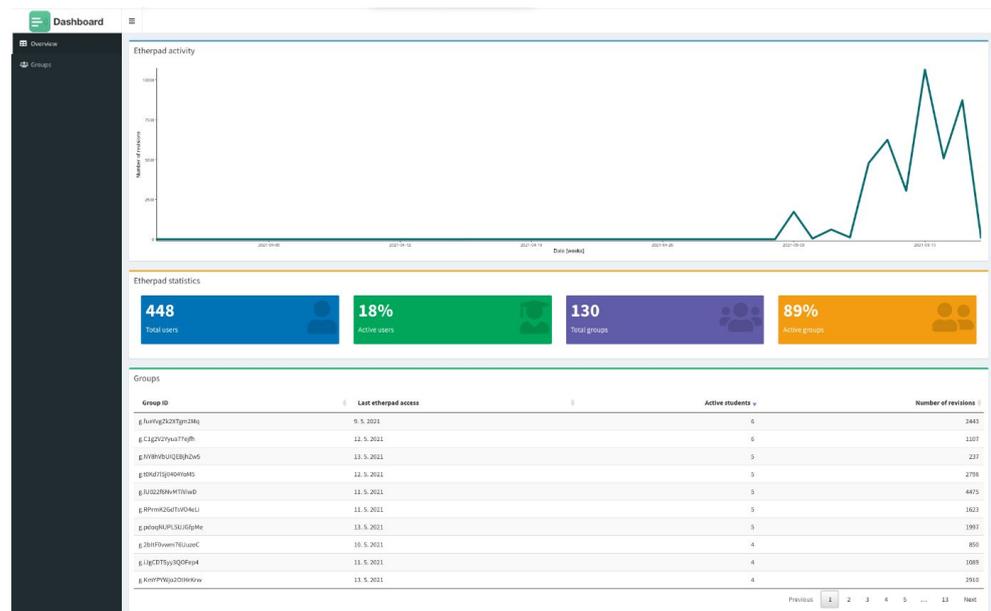


Abb. 3: Lehrenden-Dashboard.

Zurzeit besteht die erste Version des Dashboards aus drei analytischen Komponenten:

- Der oben dargestellte Kurvenverlauf zeigt die Gesamtaktivität im *Etherpad-Lite*-Editor der Komponente *Schreiben* als Anzahl der Textänderungen in allen Gruppen pro Tag.
- Die darunter gezeigte *allgemeine Statistikkomponente* umfasst die Gesamtzahl der im CWE-System registrierten Studierenden, den Prozentsatz der aktiven

Studierenden, die Gesamtzahl der registrierten Gruppen und den Prozentsatz der aktiven Gruppen.

- Die folgende *Gruppenübersichtstabelle* enthält Informationen zu den einzelnen Gruppen. Sie gibt Auskunft darüber, wie viele der den Gruppen zugewiesenen Studierenden aktiv waren, wie viele Textüberarbeitungen von der Gruppe vorgenommen wurden und wann der letzte Zugriff auf das Gruppendokument erfolgte.

8. Piloteinsatz

Die erste Version der behandelten Architektur wurde im Sommersemester 2021 als Piloteinsatz in einem Modul des Bachelor-Studiengangs in Psychologie an der Fern-Universität in Hagen mit etwa 300 Studierenden getestet. Eine darin zu bearbeitende kollaborative Schreibaufgabe zum wissenschaftlichen Arbeiten musste in Gruppen mit jeweils acht Studierenden gelöst werden.

Für die Bearbeitung der Schreibaufgabe wurden den Studierenden mithilfe der hier beschriebenen Architektur vier *Etherpad-Lite*-Instanzen (als Ergebnis des in Abschnitt 4.2 beschriebenen Lasttests) zur Verfügung gestellt. Der Einsatz verlief ohne grosse Probleme. Es zeigte sich, dass die Gruppen bzw. Gruppenmitglieder meist zu unterschiedlichen Zeiten, verteilt auf die Tage und Tageszeiten, arbeiteten, sodass es selten zu einer starken Auslastung des Servers kam. Die stärkste Auslastung trat kurz vor dem Abgabetermin auf, ohne jedoch den Server zu überlasten. Sowohl die *Etherpad*-Instanzen als auch der verwendete Proxy und die singuläre Datenbank der Komponente «Schreiben» (siehe Abschnitt 5) stellten dabei keinen Engpass dar.

9. Diskussion

Der in Abschnitt 4.2 beschriebene Lasttest betrachtet eine sehr hohe Auslastung einer *Etherpad*-Instanz, welche im Worst-Case-Szenario auftritt, aber im realen Einsatz selten vorkommt. Im Bereich der Entwicklung von (nahezu) Echtzeitwerkzeugen zur Unterstützung des kollaborativen Schreibens ist dennoch eine möglichst geringe Verzögerung in der Übermittlung aller Änderungen am gemeinsamen Dokument einer Gruppe notwendig.

Die in dieser Arbeit vorgestellte Architektur hat sich als ein erster wesentlicher Schritt hin zur Bereitstellung einer Lernumgebung für das kollaborative Schreiben mit Hunderten von Studierenden herausgestellt. Dennoch weist die sie derzeit u. a. noch folgende Schwächen auf:

- Engpässe durch die Verwendung einer einzigen Datenbank in der Komponente «Schreiben» bzw. durch den Reverse Proxy sind nicht auszuschliessen. Bei der Datenbank können sie insbesondere durch die Ein-/Ausgabelast auftreten.

- Die statische Festlegung der Anzahl von *Etherpad*-Instanzen vor jedem Kursdurchlauf führt zu einem hohen Aufwand und soll in Zukunft automatisiert in Abhängigkeit von der Anzahl der Gruppen bzw. Studierenden geschehen.
- Ausserdem ist es möglich, dass mehrere Gruppen davon betroffen sind, wenn eine der *Etherpad*-Instanzen ausfällt. Deshalb soll die Entwicklung zu einer stärkeren softwareseitigen Aufteilung der Gruppen auf eine grössere Anzahl von *Etherpad*-Instanzen eingeschlagen werden.

Grundsätzlich hat sich wie bei Tapia et al. 2020 herausgestellt, dass die Bereitstellung von mehreren *Etherpad*-Instanzen unter Verwendung der Software Docker vorteilhaft ist. Der in Abschnitt 7 beschriebene Prototyp des Lehrenden-Dashboard lässt aktuell noch sehr wenig Rückschlüsse auf den konkreten Ablauf des Lernprozesses in einer Gruppe zu, was seine Fähigkeiten als Frühwarnsystem (EWS) einschränkt. Gerade für seine Nutzung als EWS müssen hier weitere Funktionalitäten geschaffen werden, die die Leistung der einzelnen Studierenden bzw. der Gruppe genauer betrachten und darstellen.

10. Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Architektur zur Unterstützung des kollaborativen Schreibens in Kursen mit einer grossen Anzahl von Studierenden vorgestellt. Einer der Hauptbestandteile des Systems ist der kollaborative Online-Schreibeditor *Etherpad Lite*, der über ein *Moodle*-Plug-in in das Lernmanagement-System *Moodle* integriert wurde. Die Schwäche von *Etherpad Lite* im Einsatz mit vielen Studierenden, die sich bei hoher Auslastung durch untragbar lange Verzögerungszeiten zeigen kann, wurde mit der implementierten verteilten Lösung beseitigt.

Zur Architektur gehört zudem eine umfassende Erhebung von Daten, die durch zukünftig neu geschaffene Werkzeuge zur Unterstützung des kollaborativen Schreibens genutzt wird. Detaillierte Informationen über das Verhalten von Studierenden und ihrer Gruppen werden im kollaborativen Editor und im realisierten Tracking-Plug-in gesammelt. Es ermöglicht eine Analyse des individuellen und gruppenbezogenen Schreibprozesses – unter Verwendung der bereitgestellten Schreib- und Kommunikationswerkzeuge. Darüber hinaus können bestehende schreibanalytische Ansätze auf die Gruppenebene übertragen werden, um durch adaptive personalisierte Anmerkungen, Feedbacks und Instruktionen das Gruppenklima und die Schreibkompetenz zu fördern. Auf der Grundlage der gesammelten Daten wurde der Prototyp eines Lehrenden-Dashboards entwickelt. Es ist der erste Schritt zur Implementierung eines Frühwarnsystems, das hilft, die Gruppen zu identifizieren, bei denen die Gefahr besteht, dass sie die zugewiesenen Aufgaben nicht erfolgreich erledigen werden.

Aktuell wird an einer verbesserten Version der Architektur gearbeitet, die u. a. das Anlegen von *Etherpad*-Instanzen automatisiert. Zusätzlich werden in der oben behandelten Komponente «Schreiben» ebenfalls mehrere Datenbanken automatisiert angelegt, um ggf. einem auftretenden Engpass entgegenzuwirken. Um das kollaborative Schreiben weiter zu unterstützen, soll in naher Zukunft zudem damit begonnen werden, die Architektur um ein intelligentes Tutoringsystem zu erweitern. Gerade dafür ist es wichtig, dass die dafür notwendigen Daten erhoben und die Verzögerung innerhalb der Übertragung der Textänderungen klein gehalten werden. Parallel dazu geht die o. g. Entwicklung des Lehrenden-Dashboards zu einem leistungsfähigeren Frühwarnsystem für die Erkennung von auftretenden Problemen innerhalb der Gruppe in die nächste Phase, in der zusätzliche Funktionen in Form von Übersichten über die Leistungen der einzelnen Studierenden bzw. der Gruppen implementiert werden. Mit den erhaltenen Daten soll zudem die Diversität der Studierenden untersucht und in den entwickelten Werkzeugen berücksichtigt werden. Weiterhin ist geplant, zusätzliche Unterstützungsmittel für die Studierenden zu schaffen, z. B. in Form eines Studierenden-Dashboards mit wichtigen Informationen zu ihren Lernaktivitäten bzw. Interaktionen.

Literatur

- Allen, Laura K., Matthew E. Jacovina, und Danielle S. McNamara. 2016. «Computer-Based Writing Instruction». Grantee Submission. <https://eric.ed.gov/?id=ed586512>.
- Binkley, Marilyn, Ola Erstad, Joan Herman, Senta Raizen, Martin Ripley, May Miller-Ricci, und Mike Rumble. 2012. «Defining Twenty-First Century Skills». In *Assessment and Teaching of 21st Century Skills*, herausgegeben von Patrick Griffin, Barry McGaw, und Esther Care, 17–66. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-2324-5_2.
- Crow, Michael M., und William B. Dabars. 2015. *Designing the New American University*. Baltimore: JHU Press.
- Dang, Quang-Vinh, und Claudia-Lavinia Ignat. 2016. «Performance of realtime collaborative editors at large scale: User perspective». In *2016 IFIP Networking Conference (IFIP Networking) and Workshops: IEEE*. <https://doi.org/10.1109/ifipnetworking.2016.7497258>.
- Gibson, Andrew, Adam Aitken, Ágnes Sándor, Simon Buckingham Shum, Cherie Tsingos-Lucas, und Simon Knight. 2017. «Reflective writing analytics for actionable feedback». In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, herausgegeben von Alyssa Wise, Phillip H. Winne, Grace Lynch, Xavier Ochoa, Inge Molenaar, Shane Dawson, und Marek Hatala, 153–62. New York, NY, USA: ACM. <https://doi.org/10.1145/3027385.3027436>.

- Gour, Vishal, S. S. Sarangdevot, Govind Singh Tanwar, und Sharma Anand. 2010. «Improve Performance of Extract, Transform and Load (ETL) in Data Warehouse». *International Journal on Computer Science and Engineering* 2 (3). <http://www.enggjournals.com/ijcse/doc/IJCSE10-02-03-108.pdf>.
- Graesser, Arthur C., Danielle S. McNamara, Max M. Louwerse, und Zhiqiang Cai. 2004. «Coh-Metric: Analysis of Text on Cohesion and Language». *Behavior research methods, instruments, & computers: a journal of the Psychonomic Society* 36 (2): 193–202. <https://doi.org/10.3758/BF03195564>.
- Ignat, Claudia-Lavinia, Gérald Oster, Olivia Fox, Valerie L. Shalin, und François Charoy. 2015. «How Do User Groups Cope with Delay in Real-Time Collaborative Note Taking». In *ECSCW 2015: Proceedings of the 14th European Conference on Computer Supported Cooperative Work, 19–23 September 2015, Oslo, Norway*, herausgegeben von Nina Bouλους-Rødje, Gunnar Ellingsen, Tone Bratteteig, Margunn Aanestad, und Pernille Bjørn, 223–242. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-20499-4_12.
- Laal, Marjan, Mozghan Laal, und Zhina Khatami Kermanshahi. 2012. «21st Century Learning; Learning in Collaboration». *Procedia – Social and Behavioral Sciences* 47: 1696–1701. <https://doi.org/10.1016/j.sbspro.2012.06.885>.
- Perin, Dolores, und Mark Lauterbach. 2018. «Assessing Text-Based Writing of Low-Skilled College Students». *Int J Artif Intell Educ* 28 (1): 56–78. <https://doi.org/10.1007/s40593-016-0122-z>.
- Roscoe, Rod D., und Danielle S. McNamara. 2013. «Writing Pal: Feasibility of an Intelligent Writing Strategy Tutor in the High School Classroom». *Journal of Educational Psychology* 105 (4): 1010–1025. <https://doi.org/10.1037/a0032340>.
- Schwendimann, Beat A., Maria Jesus Rodriguez-Triana, Andrii Vozniuk, Luis P. Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet, und Pierre Dillenbourg. 2017. «Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research». *IEEE Trans. Learning Technol.* 10 (1): 30–41. <https://doi.org/10.1109/tlt.2016.2599522>.
- Shibani, Antonette 2017. *Combining automated and peer feedback for effective learning design in writing practices*. <https://opus.lib.uts.edu.au/handle/10453/115965>.
- Shibani, Antonette, Simon Knight, Simon Buckingham Shum, und Pamela Ryan. 2017. *Design and implementation of a pedagogic intervention using writing analytics*. <https://opus.lib.uts.edu.au/handle/10453/115878>.
- Shibani, Antonette, Simon Knight, und Simon Buckingham Shum. 2018. «Understanding Revisions in Student Writing Through Revision Graphs». *International Conference on Artificial Intelligence in Education*, 332-336. Springer, Cham. https://doi.org/10.1007/978-3-319-93846-2_62.
- Shibani, Antonette, Simon Knight, und Simon Buckingham Shum. 2019. «Contextualizable Learning Analytics Design». In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, 210–219. New York, NY, USA: ACM. <https://doi.org/10.1145/3303772.3303785>.

- Southavilay, Vilaythong, Kalina Yacef, Peter Reimann, und Rafael A. Calvo. 2013. «Analysis of collaborative writing processes using revision maps and probabilistic topic models». In *Proceedings of the Third International Conference on Learning Analytics and Knowledge – LAK '13*. New York, New York, USA: ACM Press. <https://doi.org/10.1145/2460296.2460307>.
- Storch, Neomy. 2005. «Collaborative Writing: Product, Process, and Students' Reflections». *Journal of Second Language Writing* 14 (3): 153–173. <https://doi.org/10.1016/j.jslw.2005.05.002>.
- Tapia, Freddy, Miguel Ángel Mora, Walter Fuertes, Jorge Edison Lascano, und Theofilos Toulkeridis. 2020. «A Container Orchestration Development that Optimizes the Etherpad Collaborative Editing Tool through a Novel Management System». *Electronics* 9 (5): 828. <https://doi.org/10.3390/electronics9050828>.
- Wininger, Michael. 2014. «Measuring the Evolution of a Revised Document». *Journal of Writing Research* 6 (1): 1–28. <https://doi.org/10.17239/jowr-2014.06.01.1>.

Danksagung

Ein besonderer Dank gilt Stefan Stürmer, Natalia Reich-Stiebert, Jan-Bennet Voltmer und Jennifer Raimann für ihre Unterstützung und ihr Feedback während der Entwicklung der hier beschriebenen Architektur.

Diese Forschung wird durch den Forschungsschwerpunkt «Digitalisierung, Diversität und lebenslanges Lernen. Konsequenzen für die Hochschulbildung» (D²L²) der FernUniversität in Hagen unterstützt.