Supporting collaborative writing tasks in large-scale distance education

Marc Burchart ¹ and Joerg Haake ²

¹Research Institute for Telecommunications and Cooperation e.V.
²Affiliation not available

December 7, 2023

Abstract

In distance education programs with a large number of students, the organization and facilitation of collaborative writing projects are particularly challenging. Teachers must specify the didactical design and group formation, supervise and support distributed groups, grade, and finally evaluate the learning experiences. Distributed student groups need their own workspace including both, support for a structured writing process including necessary instructions and materials as well as tools for collaborative text editing, communication, coordination, and providing formative and summative feedback. Current approaches to support collaborative writing in education are mostly based on the use of Web 2.0 applications, such as Wikis and Weblogs, or Collaborative real-time text editors, failing to support teachers and students appropriately. As a consequence, teachers often refrain from implementing collaborative writing projects in large scale distance learning courses. We introduce a process model of a collaborative writing project aimed at creating a summary of a research paper and present the architecture and implementation of a Collaborative Learning Platform implementing Collaborative Writing Activities by an extension of a Learning Management System and integrating it with a collaboration environment. The platform supports the phases and activities of the process model and provides distributed teachers and students with integrated support throughout the collaborative writing project lifecycle. Our experience shows that the platform provides a scalable, responsive, and robust environment for collaborative writing and is accepted by teachers and students. It provides the basis for the analysis of collaborative writing behavior and further research.
Supporting collaborative writing tasks in large-scale distance education

Marc Burchart and Joerg M. Haake

Abstract—In distance education programs with a large number of students, the organization and facilitation of collaborative writing projects are particularly challenging. Teachers must specify the didactical design and group formation, supervise and support distributed groups, grade, and finally evaluate the learning experiences. Distributed student groups need their own workspace including both, support for a structured writing process including necessary instructions and materials as well as tools for collaborative text editing, communication, coordination, and providing formative and summative feedback. Current approaches to support collaborative writing in education are mostly based on the use of Web 2.0 applications, such as Wikis and Weblogs, or Collaborative real-time text editors, failing to support teachers and students appropriately. As a consequence, teachers often refrain from implementing collaborative writing projects in large scale distance learning courses. We introduce a process model of a collaborative writing project aimed at creating a summary of a research paper and present the architecture and implementation of a Collaborative Learning Platform implementing Collaborative Writing Activities by an extension of a Learning Management System and integrating it with a collaboration environment. The platform supports the phases and activities of the process model and provides distributed teachers and students with integrated support throughout the collaborative writing project lifecycle. Our experience shows that the platform provides a scalable, responsive, and robust environment for collaborative writing and is accepted by teachers and students. It provides the basis for the analysis of collaborative writing behavior and further research.

Index Terms—Learning Environments, Collaborative Learning Tools, Peer Reviewing

I. INTRODUCTION

Collaboration skills are an essential requirement of today’s society [1], [2]. Therefore, it is not surprising that higher education programs usually include group projects in some courses, in which students learn collaboratively. Joint learning in these projects has positive impacts, such as improving students’ writing skills through Collaborative Writing (CW). Among other advantages, writing together can produce texts that are grammatically and lexically more accurate [3], [4], linguistically more complex, and more succinct than texts produced individually [5]. Furthermore, CW can not only improve student writing through collaborative activities [6]–[8] but often increase motivation [6], [9].

In distance education courses with a large number of participating students, the organization and facilitation of such group projects is especially challenging. Distance Learning Environments (DLEs) require the provision of dedicated shared workspaces for each group, including support for collaborative text editing, communication, coordination, and learning support through instruction, assessment, and feedback.

In practice, DLEs are provided by an e-learning platform offering “access to learning content and tests, communication, and collaboration tools for students, as well as course management and assessment facilities for teachers” [10, p. 21].

In distance learning environments, special attention must be paid to the diversity of students, so that joint learning and writing is possible both asynchronously and synchronously. Furthermore, CW in an online environment takes place in different phases that need to be defined by teachers and performed by students. To support smooth interaction within projects, the above mentioned shared workspaces and the tools provided by them must show sufficient responsiveness (e.g., quick display of text changes of remote project members), scalability (e.g., for large cohorts, large number of groups), and robustness (i.e., fault-tolerance, prevention of data loss, and error reporting). Furthermore, all software and IT systems must be compliant with applicable laws and regulations, such as the General Data Protection Regulation (GDPR in Europe).

For teachers, it can be time-consuming and difficult to facilitate collaborative learning. Student cohorts must be divided into reasonable groups before they can be assigned to newly created collaborative workspaces. Therefore, a Collaborative Learning Platform must support the formation of groups and the automatic provision of a collaborative workspace for each group. In addition, teachers need support for monitoring interactions in groups and for identifying groups that require their attention (e.g., due to deficiencies in communication or collaboration, dropout, or explicit requests for help).

Current approaches to supporting CW are mostly limited to providing student groups with tools for collaborative text editing, a way to communicate, and thus to coordinate their activities. In this context, the emphasis in education is on the use of Web 2.0 applications such as Wikis, Weblogs or Collaborative Real-Time Text Editors [11], [12]. However, these tools lack support for shared synchronous text editing [13], or do not comply with strict privacy policies of European universities, or lack adequate scalability and robustness [14]. Furthermore, these tools lack support for teachers with respect to automatic group formation, assignment of students to CW.

Marc Burchart is with the Research Institute for Telecommunications and Cooperation e.V., 44149 Dortmund, Germany (e-mail: marc.burchart@fernuni-hagen.de).

Joerg M. Haake is with the FernUniversität in Hagen, Faculty of Mathematics and Computer Science, 58097 Hagen, Germany (e-mail: joerg.haake@fernuni-hagen.de).
workspaces, defining learning phases, task instructions, and learning materials. In addition, these tools lack support for students with respect to learning and to perform different learning phases as well as to access task instructions and learning materials by all group members. Additional research is required to create more supportive workspaces for collaborative learning and writing in a group project (e.g., [15], [16]).

In general, none of the above-mentioned approaches provides teachers with holistic and scalable support for planning, setting up, running, and monitoring group projects that fully address the needs of students and teachers identified above. A solution is needed that makes it easier for teachers to create groups and that automatically assigns members to the dedicated group workspaces. Furthermore, it should be able to allow scalable and robust asynchronous and synchronous text editing and group communication, help monitoring the learning and writing process, and offer learning support in different learning and writing phases.

In this paper, we therefore present a Collaborative Learning Platform that addresses these deficits. We use the example of a CW project being part of a course on the introduction of students to scientific work in a distance education program for the Bachelor of Science in Psychology. In this project, student groups of eight analyze and summarize a typical psychology research paper to become familiar with typical structure, methods, and writing in psychology research. Each semester, up to 2,343 students are enrolled in this course.

In Section II we describe teacher guidance, phases of the writing process, and the process model of the collaborative summary writing project, followed in Section III by a requirements specification for a collaborative learning platform supporting this type of project. Section IV presents related research while in Section V, we present the architecture of the proposed collaborative learning platform. Section VI briefly discusses the implementation of the platform. In section VII we describe our experiences using the platform over three semesters with more than 4,500 students. Section VIII provides our conclusions and an outlook on future work.

II. LEARNING HOW TO COLLABORATIVELY WRITE A SUMMARY OF A RESEARCH PAPER

The CW project considered in this paper aims to improve the scientific writing skills of students enrolled in a large distance learning program by letting them write in groups a summary of a typical research paper in their discipline. For this, students must be divided into groups and carry out the writing process in a structured manner. Depending on the teacher’s instructions, this process is carried out either as individual writing of different parts of the text (horizontal-division writing) or as collaborative writing of the entire text document (reactive writing) [17].

A. Description of teacher’s guidance

In order to learn collaborative scientific writing, students need to be guided through the learning activity by teachers. They give instructions on how to solve a specific task, monitor and regulate the group activities, and intervene in the case of problems or poor performance by individuals, several group members, or the whole group.

Writing skills are improved by receiving formative feedback in the writing process [18]–[20]; more accurately, current research suggests that in academic writing both giving and receiving formative feedback can increase writing performance [19]. In addition, the review skills of the students are, of course, also increased. As indicated by [20] peer feedback within collaborative writing is more reflectively and constructively received by students than pure teacher corrections. Thus, significantly higher improvements in writing are observed.

At the end of the writing process, teachers review the performance of the students using summative evaluation as a retrospective assessment of learning [21]. For this purpose, text documents, as well as group and individual performance, must be considered. The aim is that teachers provide objective assessment, expert feedback and a comparison of group performance with that of the other groups.

B. Phases of the writing process

Usually, the collaborative writing process is divided into different numbers of phases. In the following, we present a few examples.

[17] defined seven main activities to perform general writing tasks collaboratively. During the brainstorming phase students work together to develop ideas for the planned text. Subsequently, in the convergence of brainstorming phase the group members consider how they can realize the ideas developed. Within the following outlining activity phase the general direction of the text is determined, and text sections and subsections are defined. Thereupon, in the drafting phase a first version of the text is created as a draft, which in the following revision phase is read and annotated by peer students and/or other editors for possible improvements in grammar, style and content. Through the following revising phase, the group takes into the account the received comments and, if necessary, incorporates the proposed changes into their texts. In the last copyediting phase, usually performed by a single person, the texts of all members of the group are subjected to changes to achieve a common consistency.

[22] defines collaborative writing similarly, but as a four-step process with a brainstorming phase, planning and organization phase, (text) composition phase, and a review phase.

[23] defined five phases for a computer-mediated collaborative writing project to create a story script for 51 second-language learners. Each phase lasted one week. The collaborative planning phase was used to think and discuss the story and plan the joint writing. At the end of the phase, the title, the characters that appear, a summary, and the individual scenes, as well as the respective author of each scene, had to be documented. In the partitioned drafting phase each group member created the design of his/her assigned scene with details about the scenario, the dialogue of the appearing characters, the description of the story-line and all background information. The generated draftings were made available to all group members. During the peer revising phase each group member reviewed the scenes of all other team members. In this
process, each member could add or delete content and give feedback to discuss. In the \textit{peer-editing phase}, each member of the team corrected the texts of the other members. Text changes had to be discussed to develop the metalinguistic skills of the learners. In the last \textit{individual publishing phase} each member combines all the individually elaborated scenes to a single complete story and revised and edited the text according to his/her own preferences into a final version.

[6] investigated how groups of students self-structured their collaborative writing of an essay. After the topic of its work had been given to the group, the group members thought together about how they wanted to carry out their joint work, created an outline for their text document, and agreed on the form of dividing the work among the group members. Then they broke down the expected document into sub-sections to be produced individually by the group members, taking into account the different competencies of the members. After each group member had completed its part, the whole group combined these parts into one document. This document was proofread by all members and the uniformity of style and expression was checked before submitting it to the teacher.

\textbf{C. Process model of our collaborative summary writing project}

We defined the following phases and activities for the collaborative writing of a summary of a research paper (cf. Figure 1):

In the \textit{Preparation Phase}, teachers first need to define the \textit{didactical design} of the considered writing activity. This includes specifying the task and needed material (e.g., the research paper to be summarized) as well as specifying the instructions for the students (i.e., selection of the writing strategy and planning the individual writing phases including the timing). Furthermore, the specification of the review process (i.e., determining the frequency of the reviews and when they need to be done, designing questions for a guided formative feedback, and assigning peer groups to perform the review) and the criteria for a successful completion must be done. Finally, teachers must specify the group formation, for example, group size and formation process, such as manual formation by teachers or self-selection by students, or automatic formation by a \textit{group formation service} using random or criteria-based assignment to groups. After these specifications, the collaborative learning environment for each group must be \textit{set up} in the LMS providing access to task, material, instructions and tools. Students in the same group must be assigned to the same collaborative learning environment. Then the collaborative writing activity can start.

The \textbf{Collaborative Writing Phase} consists of two parallel activities: Collaborative Writing Activity performed by students and Monitoring Activity performed by teachers.

The \textbf{Collaborative Writing Activity} begins with a Text \textit{Production Activity} during which the group composes a first draft of the summary. It is assumed that students get to know each other earlier in the learning environment, as the collaborative writing activity is embedded in the course syllabus. The Text Production Activity contains the actual joint summary writing. Due to the diversity in distance education (i.e., the professional and personal contexts of the students), this can take place in either asynchronous or synchronous form. Regarding the different CW strategies described by [17], the didactics of our CW project focus on horizontal divisional writing, where writing is done individually in different parts of the text, and reactive writing, where students work together on the entire text document, taking into account text changes or feedback from other group members. The strategy for the respective task is always predefined in the instructions given by the teachers. Due to the time constraints of the course in which the CW activity will take place, teachers must specify in the Preparation Phase when and how often they demand that the created text be reviewed and revised in a cyclic sequence. If a review is not demanded after the last Text Production Activity the Collaborative Writing Activity ends and is followed by the \textit{Assessment Phase} (see below). Otherwise, it is followed by a Peer \textit{Review Activity}, in which each member of the group or another assigned group receives the current version of the text to give formative feedback, guided by the instructions. To do this, students have to answer a teacher-defined questionnaire and can give direct comments and annotations on self-selected parts of the text. As for the Text Production Activity, the instructions specify whether each student individually reviews a different part of the text or the whole text.
In parallel to the Collaborative Writing Activity, teachers perform a Monitoring Activity in which they monitor group behavior and intervene when they see critical behavior. They also act on requests for help from students.

The last phase of collaborative summary writing is the Assessment Phase, in which teachers either grade the individual performance of the students based on the horizontal task division or grade the group performance in the case of reactive writing. Teachers provide summative feedback and compare the group’s performance to the others.

Finally, in the Evaluation Phase teachers may evaluate the learning experience based on monitoring data and the results of the assessment phase. This may lead to improvements in future learning design. Additionally, researchers may use the above data to gain insights into learning processes, the use of the collaborative learning environment by students and teachers, and their effectiveness.

The transitions in Figure 1 define when documents are passed to the next phase. In each case, we specify who has access to which part of the document. Thus, there is no need for a certain kind of Submission Phase in which the text document must be submitted or handed over.

III. REQUIREMENTS SPECIFICATION

In order to support large-scale collaborative writing projects in distance learning, together with the phases defined in the previous Section II, a Collaborative Learning Platform (CLP) must meet the following 34 requirements, listed and labeled R1 through R34.

Teachers must be able to create Collaborative Writing Activities in online courses as defined in the Preparation Phase (cf. Section II). For this purpose, the CLP must allow teachers to specify the didactical design of the Collaborative Writing Activity (R1). The specification of the didactical design includes the task and material needed, instructions to be followed by the students, the frequency of reviews and, if necessary, questions for guided formative feedback used during the review (cf. Section II), peer group assignment for reviewing and criteria for a successful completion. In addition, the CLP must allow teachers to specify group formation (e.g., group size and formation process) to allow group formation for the collaborative learning (R2).

As part of the Distance Learning Environment the CLP must provide each student group with its own dedicated Virtual Workspace (VWS), so that its members can work together on a shared text artifact (R3). The CLP must integrate each CWA and each provided VWS into a Learning Management System (LMS), as LMS are commonly used to centralize, manage and organize learning activities and materials (R4). In this context, the CLP should allow students to use their LMS account for performing the entire CWA in order to avoid the need and problem of setting up accounts for external tools not controlled by the university (cf. R28) (R5).

The CLP must support the actual instantiation of the VWS for each group (set up), which provides access to tasks, materials, instructions, and tools. For this, students must be divided into groups of same or different sizes, which are gathered in a set of groups to be assigned to different activities. The CLP must provide an automatic or manual group formation process for teachers to perform the group formation of a large number of students (R6). As specified in Section II, the CLP must also support group formation by self-enrollment, random assignment, or teacher’s assignment (R7). To avoid an additional heavy workload for teachers, the CLP must automatically create a VWS for each formed group and assign the group members to it (R8).

In the Collaborative Writing Phase (cf. Section II), each group works on its collaborative writing task in its own VWS. The CLP must provide a group-based access control for each VWS and the tools provided by it to protect the work of a particular group from unwanted external influences and to allow teachers to evaluate and assess the performance of each group and its members (R9). To guide students through the intended phases of the group writing project (cf. Section II) and to support structured work, the CLP must provide each group with a VWS that conveys to students the necessary instructions about the overall process and each phase (R10). For the guide of the students through the intended phases, the CLP must also provide each group with a VWS that informs the students about the timelines of the phases (R11). To avoid confusion, the CLP should provide each group with a VWS that automatically provides all the functionalities and information required for the current phase, including information facilitating group awareness (R12).

To facilitate the Collaborative Writing Activity according to the process described in Figure 1, the CLP must provide each group with a VWS that has a writing tool for joint text production to support the Text Production Activity (cf. Section II) (R13). This writing tool must meet a number of requirements: It must allow students to collaboratively manipulate shared text documents (text artifacts) (R14), both asynchronously and synchronously due to the diversity of students in distance learning environments (e.g., different individual availability) (R15), and also simultaneously with all changes becoming visible to all group members as fast as possible to allow fluent collaboration (R16). For horizontal task division (cf. Section II), the time delay is not as critical due to the loosely coupled work. In case of reactive writing, groups usually employ a synchronous audiovisual channel, which enables a coordination of the joint text editing. Therefore, a delay in visibility of one or two words seems acceptable. To support collaboration and exchange of ideas the CLP must provide each group with a VWS that includes tools to support communication and coordination between group members (R17).

Depending on the teacher's specification, the Collaborative Writing Activity requires reviews. Thus, the VWS must support the Peer Review Activity (cf. Section II). The CLP must provide each group with a VWS that enables group members, members of a peer group, and teachers the possibility to give feedback on the produced text, both, in a collaborative as well as individual manner, so that they can provide formative feedback (R18). For this purpose, the CLP must provide each group with a VWS that provide members of the peer group with the functionality to answer a teacher-defined questionnaire for guided formative feedback (R19). In
addition, the CLP must provide each group with a writing tool that allows the annotation of selected text passages to group members, members of a peer group, and teachers so that they can provide formative feedback directly for the selected text passages (R29). The CLP must ensure that the number of revisions of the created text, as specified by the teachers, is complied with by a corresponding number of review (cf. Peer Review Activity) and rewriting (cf. Text Production Activity) activities before the Assessment Phase (cf. Section II) (R21).

To support the Monitoring Activity (cf. Section II), in which the teacher monitor group behavior and intervene when they see critical behavior, the CLP must allow teachers to monitor group behavior, evaluate group interactions, and identify groups that need teacher attention, e.g. due to deficits in communication, collaboration, dropouts, or due to explicit requests for help (R22). This implies that the CLP must offer students the ability to contact their teachers, for example, when problems or questions arise about instructions (R23). On the other hand, the CLP must support teachers to communicate with groups and their members at any time, for example, for feedback, problem solving, or answers to questions (R24). The CLP also must grant teachers full access to each VWS and its tools for the provision of quality assurance, feedback, problem solving, and assessment (R25).

For the Assessment Phase (cf. Section II), the CLP must allow teachers in a horizontal writing activity to grade each student’s performance, or in a reactive writing activity to grade the group performance, and to provide general (summative) feedback to the groups and it’s students (R26).

For the Evaluation Phase (cf. Section II), the CLP must support teachers and researchers in evaluating the learning experience by providing access to monitoring data and to the results of the assessment phase (R27).

The CLP and the required tools must be in accordance with applicable laws and regulations, such as the General Data Protection Regulation (GDPR), since the FernUniversität in Hagen, as a public German university, is subject to legal requirements for the proper use of all procedures and IT applications (R28). In addition, the CLP should not require students to explicitly transfer data or documents between the tools employed in the CWP, in order to minimize the risk of data loss or using wrong documents (R29). In order to limit the costs for licenses induced by large number of students, the CLP should avoid the use of software that needs to be commercially licensed (R30).

For a comprehensive and fully functional system, the CLP must meet the following general requirements that facilitate effective and efficient use of the system. The CLP and required tools must provide sufficient scalability to allow collaboration in large-scale distance learning courses (R31). The CLP and required tools must also ensure responsiveness to facilitate fluent and effective collaboration within large-scale distance learning courses (R32). The CLP and required tools must provide robustness to avoid demotivation due to system failures, breakdowns, or data loss (R33). The CLP must immediately inform the system administration in the event of serious technical problems for rapid human intervention (R34).

IV. RELATED WORK

In education, a wide variety of different software solutions are used to support collaborative writing tasks. In general, these tools help students communicate with each other, collaborate on common text artifacts, and coordinate their activities. We distinguish the following categories of solutions: (A) Wikis, Blogs & Forums, (B) Repository-based solutions, (C) Collaborative real-time Text Editors, (D) Collaborative Peer-to-Peer text editors, (E) Microservice-based solutions, (F) Frameworks, and (G) Shared Workspaces. In the following, we discuss how current solutions in these categories address the requirements identified in the Section III.

A. Wikis, Blogs & Forums

Frequently employed solutions use Wikis (e.g., [24]–[28]). They allow multiple users to write or edit documents, primarily asynchronously, and can also be used for coordination and communication [29]. After saving the changes, a new version of the document is automatically published for all group members to access. Due to their asynchronous usage, they are especially suitable for being used in a course with numerous students and provide adequate responsiveness. By default, Wikis do not support multiple authors with synchronous (near real-time) text editing capability [13], [30], which can lead to conflicts in the version history. These conflicts must be resolved manually by users [31]. Thus, asynchronous and synchronous text edits and fluent collaboration are not fully supported. The same limitation also applies to other asynchronous collaborative text creation tools such as Weblogs (used e.g. by [32]–[34]), or Forums (used e.g. by [35], [36]).

B. Repository-based Solutions

Himmelstein et al. [37] describe the Manubot software, which allows large-scale collaborative creation of manuscripts. Documents are written using the lightweight markup language Markdown and stored in a GitHub repository. Revisions of text changes can follow a specific workflow where feedback can be provided and discussed. This can lead to additional text changes. Among others, the software can convert manuscripts to common formats (such as HTML, PDF, and DOCX), publish changes continuously as HTML on GitHub, automatically manage the bibliography, and allow citation using persistent identifiers (such as DOI, ISBN, and URLs). Although writing can be done synchronously, each students individual changes are only visible after the student commits and pushes the changed document to the GitHub repository. Thus, a fluent synchronous collaboration, where all changes are visible to all members as fast as possible, is not fully supported.

C. Collaborative real-time text editors

Collaborative real-time text editors (such as Google Docs, Overleaf, Microsoft Office 365, Collabora, OnlyOffice, Zoho Writer, CodiMD, Dropbox Paper, Authorea, and Etherpad Lite) allow web-based synchronous or asynchronous near-real-time editing of text documents. The editors differ from the previously mentioned solutions particularly in the fact that
they display the text changes to all group members in near-real time (WYSIWIS - "What You See Is What I See"). Most editors have a near real-time chat and offer the possibility of leaving comments on marked text passages. In the field of education, the proprietary editor Google Docs (e.g., [38]–[40]) and the open source text editor Etherpad Lite (e.g., [41]–[43]) were frequently used. When measuring the performance of Google Docs and Etherpad Lite, [14], [44] discovered that the central server solution of these editors, utilized to maintain document consistency, becomes a bottleneck when used by many users and groups. This bottleneck leads to a significant delay (larger than 10s) between the execution of a text change by a user and its visibility to other users, thus limiting fluent collaboration, sufficient scalability, and responsiveness.

Kumar et al. [45] describes an open source collaborative text editor named LiteDoc. The architecture is based on a distributed system. Each document is divided into different sections, which are stored in an emulated atomic single-writer multi-reader (SWMR) register. Using the distributed architecture and the SWMR register, the authors expect a better (almost linear) scalability, among others, by avoiding the above-mentioned bottleneck of a single-server solution. In addition, the editor only allows one user to write to a particular section at a time. This makes computationally intensive algorithms for consistency maintenance redundant, likely resulting in increased performance and enhanced data consistency. As the text editor is still under development, it is not yet usable. The restriction of one author per particular section at a time constrains the possibility of synchronous, fluent, collaborative reactive writing on the same section. Annotation or communication facilities are not mentioned.

D. Collaborative peer-to-peer text editors

To make synchronous web-based writing more delay-tolerant, collaborative peer-to-peer (p2p) text editing systems have been developed (e.g., [46]–[48]). In contrast to the centralized approach, each peer has a local copy of the text document and sends its text modifications directly to all others, making the system more scalable and fault-tolerant [46]. To our knowledge, there is currently no mature solution for a p2p text editor that can be considered for use in a large-scale distance education setting. The reasons for this are as follows: Firstly, many existing architectures are only usable under the specific conditions assumed during their development or perform poorly under realistic workloads [49]. Second, time-independent asynchronous editing of text artifacts must be supported, e.g., by allowing peers to receive changes from offline peers later. New approaches, for example, the p2p collaboration software PushPin (which is currently under development), try to overcome this limitation by using a so-called storage peer, which is a Unix daemon that persists the data of the offline peers on a central server [50].

E. Microservice-based solutions

As a scalable centralized approach, [51] describes a microservice architecture using Docker software. The architecture allows the deployment of multiple Etherpad Lite instances, each with its own dedicated database instance, to increase scalability and robustness. This architecture focuses on the provision of an Etherpad Lite instance for each registered user, allowing him or her to create and share text documents via a special link with other users, and therefore does not support collaborative learning groups.

F. Frameworks

All above-mentioned solutions aim to provide a basic (partially responsive, scalable, and fault tolerant) solution for collaborative text editing, communication, and coordination support. [35] describes a framework which uses the LMS Moodle [52] in combination with the Google Docs collaborative text editor and the Stormboard collaborative workflow platform to offer students the possibility to write essays collaboratively. However, the solution is difficult for teachers to manage (e.g. manually create a shared group workspace, assign the group members to it, and provide easy access to the group documents) and does not offer good usability for students. For example, in the LMS Moodle, students have to extract their text document from Google Docs and upload it to the Moodle forum. Therefore, more research is needed to offer an improved workspace for collaborative writing within the LMS. Furthermore, contrary to distance learning, online collaboration was accompanied by some face-to-face meetings of students and teachers.

G. Shared workspaces

Synchronous audio-video conferencing systems such as Zoom [53], plugNmeet [54], or Microsoft Teams [55] can be used in combination with collaborative writing editors to extend the workspace with direct communication ability (e.g., [56]). They do not provide any own collaborative writing capability (some use external editors like Etherpad Lite). Furthermore, as pure communication tools, they have no possibility to support the learning process, such as providing feedback or assessment functionalities.

[15] describes the iWrite environment for supporting collaborative learning activities for large numbers of students using Google Docs. The software allows teachers to group students manually. It automatically creates and assigns students to Google Docs group documents and includes peer review functionality, which provides PDF snapshots of submitted document drafts to defined peers, tutors, or lecturers, and offers them the opportunity to write a feedback. The environment supports three different phases: a Draft Writing Phase, a Review Phase, and a Revision Phase, in which feedback can be used for text improvements. At the end of the last phase, a PDF copy of the document is stored for teacher use, and the Google Docs document is automatically set to read-only.

Google Docs does not comply with the strict requirements of European information security and privacy regulations, such as the General Data Protection Regulation [57]. Furthermore, there are ethical concerns about user privacy and tracking while using Google Docs (e.g., [58], [59]). Additionally, as mentioned above, [14], [44] identified possible significant delays when using Google Docs, which may lead to insufficiently fluent collaboration.
The VTIE CWE [60] is a prototype of a Collaborative Writing Environment (CWE) that allows students to write scientific reports in the context of school education. In the preparation phase, it allows teachers to define tasks for collaborative report writing, to manually divide students into learning groups, and to assign an individual section of the report to be developed to each of them. The tool includes a so-called ScrapBook which allows students to share their research results (collected texts, images, links, and notes) with the other group members, and a simple WYSIWYG editor, which allows each student to create and edit his or her specific section. The creation of the section goes through several cycles until the text is finalized. Each of these cycles consists of two phases in which the respective student (1) creates, edits, and submits his/her text, after which (2) the other team members (or the teacher) give feedback on the created text. However, the solution does not allow automatic creation and assignment of a VWS, and does not offer the possibility of reactive writing of the whole text.

[61] developed a groupware tool called TC3 (Text Composer, computer supported and collaborative) to facilitate collaborative writing between pairs of students. It provides relevant information sources, a private notepad for each user, a chat with the possibility of viewing the history, and a simple (turn-taking) shared word processor that displays the changed text in near real time. However, TC3 does not offer support for large-scale synchronous joint writing and learning, such as feedback or assessment functionality, nor does he turn-taking shared word processor allow synchronous fluent collaboration.

The CURE platform [16] provides rooms as shared workspaces for collaborative learning. The rooms contain educational material, as well as tools for synchronous and asynchronous communication, coordination, and collaboration. Different rooms and links between them can be used to structure the learning process. Rooms can be made accessible to all members of the dedicated group or to members with role-specific access permissions. The platform also supports manual group formation and group maintenance. However, CURE did not address synchronous collaborative writing, scalability to large numbers of concurrent groups, as well as automatic creation and assignment of rooms to groups.

Table I shows the fulfillment of the requirements defined in Section III by the categories of solutions discussed above. We chose the strongest solution within the category to assess the fulfillment of requirements.

To the best of our knowledge, there is currently no effective solution for supporting collaborative writing in large-scale distance education courses. A highly scalable and robust platform is needed, which creates and automatically assigns groups to dedicated workspaces, in which they can collaborate asynchronously and simultaneously in near real-time and which offers support for communication and learning through instructions, feedback, and assessment. Teachers and researchers must be able to monitor group behavior to provide feedback, problem solving or providing answers to questions, and to evaluate the learning experience. In this context, we must comply with relevant European data protection laws and regulations, e.g., GDPR. A key aspect of this compliance is to ensure that third parties (e.g., Google) do not have data access.

### V. A Learning Platform for Distributed Collaborative Writing Tasks

In this Section, we describe a Collaborative Learning Platform (CLP) that addresses all requirements identified in Section III. According to [10, p. 21] “the functionality of e-learning platforms typically includes access to learning content and tests, communication, and collaboration tools for students, as well as course management and assessment facilities for teachers”. In this sense, we define our solution as a platform, since it enables collaborative writing for large-scale distance courses by providing learning material, communication, collaboration tools, and feedback functionalities for students, as well as management, monitoring, feedback, grading, and assessment functionalities for teachers. The following Section explains the architecture of the CLP (see Figure 2), described in the order of the previously defined phases (cf. Section II). Section VI delves into details of the implementation.

#### A. Preparation Phase

In the Preparation Phase, teachers can create interactive Collaborative Writing Activities (CWA) for the online courses of the LMS by defining didactical designs for each and entering them via the CWA Config (CWAC) (cf. R1).
The CWAC supports the specification of manual group formation (self-enrollment or teacher assignment) (cf. R2), which is carried out using LMS functionalities, and automatic group formation by using a group formation service that performs random or criteria-based group composition and returns the set of groups with their group members. The CWAP stores the group formation using the Core APIs of the LMS (cf. R6), and the CWAC contains the resulting group specification.

In the CWAC, teachers also specify the task, the instructions, and the links to the materials needed to perform the task (e.g., PDF files, web links, HTML texts, and pictures). Since the Collaborative Writing Activity consists of Text Production Activities (TPA) and Peer Review Activities (PRA), the teacher can specify in the CWAC the sequence of the activities to be performed. For each activity, the teacher can specify a unique name, the type of activity (TPA or PRA), the instructions, and a start and end date. The CWAC ensures that the first activity is a Text Production Activity. In the case of Peer Review Activities, the teacher needs to additionally specify the review assignment (which reviews which other group) and can optionally define a questionnaire for guided formative feedback. The questionnaire consists of a list of questions that, depending on the definition, can be answered either on a rating scale or in the form of a free text answer of limited text length.

With the configuration shown above, the overall Collaboration Writing Phase (CWP) and each of its activities are defined (cf. R10) as well as the timing of the automatic phase transitions (cf. R11), the number of iterations to be performed before the Assessment Phase (cf. R21), and the teacher-defined questionnaires for peer reviews (cf. R19). Finally, teachers choose the type of grading (none, scale, point, percent or free text) they want to use for the respective CWA in the Assessment Phase (cf. R26).

Once the CWAC is fully specified, the CWAP is able to automatically create a VWS for each group and assign the group members to it (cf. R8). This finishes the Preparation Phase and the Collaborative Writing Phase can begin at the start date defined by the first text production activity.

B. Collaborative Writing Phase

When students log into the LMS and enter their online course, all course-related activities are displayed, including all Collaborative Writing Activities. If they select a CWA, the CWAP ensures that the VWS of the respective group is presented to the user (cf. R9) containing all necessary instructions on the overall CWP and its phases (cf. R10, R11).

The CWA enables asynchronous and synchronous collaboration on a shared text artifact (cf. R15). As an LMS is designed to primarily support asynchronous learning of individual students in a scalable manner (cf. R31), it is not suitable to support the required synchronous collaboration in student groups (cf. R31) with sufficient responsiveness (cf. R32) in a scalable manner. Therefore, a Collaboration Environment (CE), which is specifically designed for scalable synchronous collaboration in a large number of student groups, is used to augment the LMS. Therefore, the proposed CWP consists of a combination of an asynchronous LMS and a synchronous CE (see Fig. 2). The LMS provides and manages the CWA, the VWS, the assessment and the phase transitions while the CE offers the tools for the actual synchronous and asynchronous collaborative writing, communication, and coordination, which are embedded in the VWS at the LMS side (cf. R14, R17, R15). Since the CE handles user authentication for tool login, a single login to the LMS suffices (cf. R5).

The CE provides each group with a Group Instance (GI) of the VWS. The GI includes all the tools needed to perform the process as specified in the CWAC. For the collaborative summary writing project, a Collaborative Text Editor (CTE) supports both TPA and PRA (cf. R20). To provide the groups with the required protected VWS and the teachers with full access, the CTE provides a group-based access control (cf. R9). To monitor group behavior and evaluate group interactions (cf. R22), the CTE stores all user interactions within the CTE in the Group Instance Database. The CTE also provides at least one communication channel (e.g., a text chat) so that the group can communicate and coordinate (cf. R17). Although the CTE is designed to facilitate synchronous collaboration between group members, it can also be used for asynchronous collaboration (cf. R15).

Since CTE must ensure fluent collaboration by providing enough fast response for one group, the architecture of the CE supports vertical and horizontal scalability. Depending on the CE and GI resource needs, horizontal scaling can be achieved by exploiting the multicore architecture of modern server processors on one server. Vertical scaling can be archived by
distributing the GIs among a pool of server nodes. Vertical scaling can be used to avoid server or node overload. Isolating GI execution also contributes to robustness (cf. R33) since errors in one GI do not affect other GIs. With sufficient computing resources (processor power or number of nodes), the architecture allows scalability (cf. R32) to a large number of groups and users (cf. R31). In this way, sufficient responsiveness can be achieved for each GI (cf. R16).

Each GI contains a Group Instance Database (GIDB) that persists and manages all data generated by its CTE. Since one group produces only a relatively small amount of data, a low-resource distributed Database Management Systems (DBMS) is sufficient, reducing the resource needs of each GI. As a result, only one CTE is affected by GIDB breakdown, which increases robustness (cf. R33).

In case of a CTE crash, the state of the database may be corrupted. In this case, access to a previous consistent state is needed. For this purpose, the Fusion Database (FDB) is used as a backup database for the previous state of each GIDB. With the FDB, this previous state can be restored to the GIDB, increasing the robustness of the system with respect to data loss (cf. R33) if the GIDB becomes unrecoverable, for example due to data corruption after a crash. The frequency of the backup can range from continuously to as soon as the CTE is no longer in use or to every few minutes. A lower frequency saves system resources, which contributes to scalability (cf. R31). If the backup process between a GIDB and FDB crashes, the CLP administration is immediately informed by the Group Management (see below). If the FDB gets corrupted, the data from the GIDBs have to be backed up again.

When a student enters the LMS and open the CWA, the CWAP requests the Group Management (GM) component to provide access to the group-specific CTE and embeds a CTE, provided by the respective GI, into the VWS of the student with the activity-related document already open.

The GM has the task of creating and managing the various GIs. To minimize the use of hardware resources, it automatically starts and stops GIs when they are required or no longer used, respectively. This increases scalability (cf. R32) and responsiveness (cf. R31) of CE, since resources are only occupied by GIs that are currently in use. The shutdown of unused GIs increases the robustness by avoiding errors that can be caused by their components (cf. R33). The GM also authorizes LMS users in the CTE by means of a group-based access control (cf. R5), only granting access to group members, teachers, or peer group members (cf. R9). To enhance robustness (cf. R33), the GM also acts as a watchdog, restarting the GI in the event of an error. When a problem arises that is unsolvable for the GM, both in its own context and in the GI context, it immediately informs the system administration (cf. R34). The GM uses its own GM Database (GMDB) to store data necessary for the management of GIs.

After the embedding of the CTE is completed, communication between the CWAP and the CTE is performed using the Reverse Proxy (RP) of the CE. As a central intermediary, the proxy is able to distribute user access to the corresponding CTE (and to the GM) by making them accessible under certain web paths. As a load-balancing mechanism, it is essential for the scalability of the system (cf. R31) and also increases security since it supervises access to CTEs and GM (cf. R28).

So far, we have described how the CE supports TPA. When a Peer Review Activity (PRA) begins, members of the peer review group get access to the respective GI of the group to be reviewed with CTE functionalities. These are limited to reading and annotating the text, and responding to comments on the annotations (cf. R20). In the PRA, the reviewed group cannot change the text created in the collaborative writing phase. In addition, the CWAP presents the teacher-defined questionnaire – as specified in the CWAC – to each member of the peer review group for creating separate guided formative feedback (cf. R19). CWAP also ensures that TPA and PRA are performed in the specified sequence (cf. R21).

In order to support the Monitoring Activity of the teachers, the CWAP sends the assignment of the CWA group, the questionnaires, the answers to the questionnaire, and the assignments of the peer group to the GM, which stores them in the FDB. This ensures that the FDB contains a complete record of all activities of all groups that can be used to monitor group behavior, evaluate group interactions, and identify potential group problems (cf. R22). Furthermore, the CWAP with the help of the GM ensures that teachers can directly access the VWS of each group without restriction and during each phase (cf. R25). In the VWS, teachers can directly communicate with students (cf. R24), while students can directly communicate with teachers using the functionalities of the LMS (cf. R23).

C. Assessment Phase

Teachers employ the gradebook provided by the LMS for the grading of all students as specified in the CWAC. For this purpose, the CWAP modifies the gradebook (provided by the LMS) so that teachers can input the grade in the gradebook for each student in the specified form (cf. R26). Additionally, teachers can provide summative feedback using the LMS gradebook (cf. R18).

Teachers can evaluate the CWA text product at any time by exploiting the full access to the VWS of each group and its embedded CTEs (cf. R25, R29). To gain insight into group behavior and group interactions (cf. R22), teachers can analyze the FDB data.

D. Evaluation Phase

In order to evaluate the learning design and learning experience of the students, teachers and researchers can access all CWP monitoring data, all CWA text artifacts (through VWS), and the assessment results (cf. R27). Insights obtained from this information may help to improve the learning design.

The CLP ensures compliance with the General Data Protection Regulation (GDPR) when processing personal data (cf. R28), since the CLP operates and stores data on servers located within the organization (e.g., the university), unauthorized access by third parties is restricted, and access control is used.

VI. IMPLEMENTATION

The implementation of the conceptual architecture of the Collaborative Learning Platform (cf. Figure 2) uses the LMS
Moodle [52] to implement the LMS functionalities for establishing, organizing and running the proposed collaborative writing project (cf. R4), while the CE part is implemented using the Docker infrastructure [62] (see Figure 3).

A. Learning Management System

The FernUniversitaet in Hagen uses the LMS Moodle [52] as the main computer-supported learning environment for distance learning. Therefore, we use it to implement the LMS functionalities for setting up, organizing, and running the proposed collaborative writing project (cf. R4).

The Moodle database is used to store data about users, courses, course content, learning activities, plug-in data, communication, and LMS configuration.

Moodle provides a set of core APIs to allow Moodle plug-ins to access Moodle functionalities and data structures. The following APIs are used by our implementation of the Collaborative Writing Project: Data manipulation API, Group API, Access API, Web Service API, Gradebook API, and Form API. The data manipulation API enables consistent and secure reading and writing to the Moodle database.

In Moodle a Learning Activity defines a feature in which students learn by interacting with each other, other students, and / or teachers (cf. [63]). Examples of learning activities are Quiz, Forum, Wiki, and Feedback. We define the Collaborative Writing Activity (CWA) as a new type of learning activity that supports the specified phases of collaborative summary writing. For the implementation of a learning activity, Moodle requires the specification of a Activity Module Plug-in.

B. Collaborative Writing Activity Plug-In

For the Collaborative Writing Activity, we developed the Collaborative Writing Activity Plug-In (CWAP) that extends Moodle so that teachers can create interactive Collaborative Writing Activities (CWAs) within online courses. This plug-in, among other tasks, allows teachers to create and edit the configuration for the respective collaborative writing activity, the CWA Config (CWAC). Teachers can define the didactical design of each CWA by specifying the respective CWAC (cf. R1) using standard Moodle functionalities. The CWAP is realized as a Moodle Activity Module Plug-In, such as the Quiz plug-in, Forum plug-in, etc. The activity plug-ins in Moodle have their own setting. The CWAP setting is used to store the URL of the Group Management (GM) component of the CE and a token to authorize access to the GM.

1) Configuration: The CWAP uses the Moodle Core Form API to create and manage a CWAC form that is used to configure the settings of a Collaborative Writing Activity (CWA). Form API enables the creation and management of consistent and secure web forms in Moodle.

In the CWAC form, the teacher can choose between manual group formation (self-enrollment or teacher assignment) or automatic group formation (cf. R6). The current CWAP implementation uses the Moodle Core Groups API to assign a set of groups to the CWAC. Teachers can assign such a set of groups with the LMS Moodle course functionalities or let students self-enroll to the groups via a Moodle Assign activity.

Current Moodle versions also allow to define a set of groups by automatic random group formations (cf. R6). Additionally, external plug-ins, like [64], also allow to define set of groups by criteria-based group composition (cf. R6).

In the CWAC form, teachers also specify the task, the instructions, and the links to the materials needed to perform the task (cf. R1). The CWAP uses a Moodle-integrated HTML editor to define the task, the instructions, and the links to the materials as an HTML text string.

Since the CWA is composed of Text Production Activities (TPA) and Peer Review Activities (PRA), teachers can specify in the CWAC form the sequence of activities to be performed. For each activity, teachers can specify a unique name, the activity type (TPA or PRA), the instructions, and a start and end date. The CWAP uses the Form API to create the individual form elements.

In the case of Peer Review Activities teachers specify in the CWAC form the assignment of the reviewing groups to the groups to be reviewed, and a questionnaire for guided formative feedback (cf. R19). The CWAC form allows to define a list of questions with the respective rating type (rating scale or free-text answer of limited text length).

Finally, teachers choose in the CWAC form the type of grading (none, scale, point, percent, or free text). The CWAP uses the Moodle Core Gradebook API to define in the Moodle gradebook the grading type for the respective CWA in the Assessment Phase (cf. R26).

Except for the grading specification, all CWAC data is stored in the Moodle database via the data manipulation API.

2) Execution: If a user opens a CWA, the CWAP reads the respective CWAC from the Moodle database and displays...
the current state of the CWA. To do so, the CWAP uses the assignment of the user to the respective group, as stated in the CWAC. This ensures that each group is provided with its own Virtual Workspace (VWS) (cf. R8).

The VWS consists of three pages: one page for displaying the task, the instructions, and the links to the materials, another page for the TPA, and a third page for the PRA. Thus, the user is provided with all functionalities and information of the current phase (cf. R10).

Figure 4 exemplary shows the VWS of a student currently in the Collaborative Writing Phase and performing a Text Production Activity. The upper left area shows a timetable (1) that informs the student about the start time and end time of all phases specified in the CWAC (cf. R11). The example shows two phases, a Text Production Phase followed by a Peer Review Phase. Users can switch between the three different pages (Instructions & Materials, Text Production, Peer Review) by using a navigation menu (2).

The Etherpad Lite editor is used to implement the Collaborative Text Editor (CTE) (cf. R14). The toolbar (3) contains basic text format settings, such as font selection, font size, italic, bold, and selection of unordered lists or ordered lists. Another toolbar (4) allows the user to export or import the opened document, save an important version, view the editing history, or see who is currently online. The actual text document (5) is shown below the two toolbars. In the text document, different colors indicate different authorships. Thus, each color is associated with an author, facilitating group awareness (cf. R17). The text, highlighted in yellow, represents the anchor of an annotation. The content of the annotation (6) is displayed on the right side. The integrated text chat (7) is shown at the bottom right of the screen. It can be used for asynchronous and synchronous communication and coordination (cf. R17).

The CWAP defines methods for (1) retrieving the task, the instructions and the links to the materials needed to perform the task from the Moodle database, (2) determining the current phase (in the case of the Collaborative Writing Phase: the time sequence of the various activities, the unique names, the activity type (TPA or TRP) and the instructions of the activities from the configuration stored in the Moodle database, (3) requesting a URL and user session token to access the respective group CTE from the Group Management component (GM) via the Reverse Proxy (RP), (4) requesting a URL and user session token to access the group CTE of a group to be reviewed from the GM via the RP, (5) accessing a teacher-defined questionnaire of a PRA, and (6) accessing the answers of a user to a teacher-defined questionnaire of a PRA from the CWAP. The data returned by the CWAP methods are cached by the CWAP as long as the user continues to use the CWAP. The CWAP uses a method to store the answers of a user to a teacher-defined questionnaire of a PRA in the Moodle database via the data manipulation API and to send it to the Group Management (GM). The GM then stores the user’s answers in the Fusion Database.

All of the above CWAP methods check if the current phase and the user’s permissions allow the execution of the respective CWAP method. If this is not the case, the execution is aborted, and an error message is displayed to the user.

C. Collaboration Environment

The Docker [62] infrastructure is used for deploying containers, virtual networks, and volumes: Containers are used to isolate the components of the CE against each other to support vertical and horizontal scalability and robustness (cf. R31, R33). Virtual networks are used to support and secure required communication between the different containers. Volumes are used to maintain persistent data used by the database components (i.e., the Group Management Database, the Fusion Database, and the Group Instance Database). The volumes allow restoring the data of containers which have become inaccessible, e.g., due to runtime errors, deadlocks, or infinite loops. An inaccessible container can be immediately recreated and the data restored by mounting the respective volume of the container again (cf. R33).

1) Group Instance: A Group Instance is implemented as a pair of containers: one container for the Collaborative Text Editor and one container for the Group Instance Database. To increases robustness because in the event of a CTE crash the CTE can be restarted or recreated using the existing Group Instance Database (GIDB) (cf. R33). A virtual network is established for communication between the two containers. Due to the isolation of CTE and GIDB and restricted accessibility of the GIDB, safety is increased.

To implement the Collaborative Text Editor (CTE), the real-time editor Etherpad Lite [65] was selected due to its free and open-source usability in compliance with the Apache License 2.0 (cf. R30), its simple operability, ease of deployment, and high customizability. It allows a learning group a fluent collaborative near real-time editing (cf. R16, R14) of text documents in a synchronous or asynchronous manner (cf. R15) and offers an integrated document-based near-real-time chat (cf. R17). Etherpad Lite also supports the extension by existing or self-developed plug-ins and is open-source. It also fosters group awareness by a list of online users and text color markings indicating different authors (cf. R12).

Group-based access control is implemented by the CTE Etherpad Lite ep_auth_sessions Plug-in [66]. When a user
accesses the CTE, this plug-in verifies the validity of the user session token provided as a URL parameter to authorize the user and grant him access to restricted group pads.

To implement the required monitoring functionality (cf. R22) an Etherpad Lite Tracking Plug-In (TP) was developed to log the following events for each user: (1) the student’s connection times to the CTE, (2) the times in which the browser tab containing the CTE is focused or not for the user, respectively, (3) the event when the student opens or closes the CTE chat, (4) changes of the scrolling positions in the text document, and (5) the scrolling positions in the chat. By combining the Etherpad Lite log data (e.g. chat messages and change sets for text or format modifications) with the TP generated data, teachers can monitor reading and writing behavior and analyze group interactions (cf. R22).

The required annotation functionality (cf. R20) was implemented by the Annotation Plug-In (ANP). The implementation of the ANP is based on the Etherpad Lite plug-in `ep_comments_page` [67], which allows color highlighting of user-selected text passages as well as shared discussions by using comments on selected text passages, and suggesting changes that may alter the text. During a Peer Review Activity of the Collaborative Writing Project, the peers are encouraged to provide feedback through text comments, but are not allowed to change the text being reviewed. Thus, the so called "suggested change" functionality of the `ep_comments_page` was removed from the ANP.

The GIDB was implemented using the document-oriented DBMS software CouchDB [68], because it is well suited for horizontal scalability (cf. R31). Furthermore, as a distributed database, CouchDB is robust to network or hardware failures (cf. R33). It persists and manages all data generated by the respective CTE (Etherpad Lite, including TP and ANP).

2) Reverse Proxy and Group Management: The Reverse Proxy (RP) and the Group Management (GM) components are combined in a single container because the GM component needs to modify the configuration of the RP to reflect the hostname, port number, http protocol and webpath of currently running Collaborative Text Editor containers within the virtual network of the RP and GM container and all CTE containers. The RP was realized using the software Nginx [69], which is considered to be powerful, stable, and resource-saving.

The Group Instance Management (GIM) was implemented using the cross-platform JavaScript runtime environment Node.js [70] together with the script language TypeScript [71]. Because Node.js works asynchronously and event-driven, it is highly scalable and can handle many concurrent requests [72]. It is only active when requests are made and thus saves system resources and increases scalability due to its non-blocking behavior (cf. R31). To give the GM access to Docker, the Node.js package Dockerode [73] is used, which allows interactions with the Docker API. The GM also provides an API that allows the CWAP methods to send requests to the GM. The API is implemented with the Node.js web framework Express [74]. Express allows to handle HTTP and HTTPS requests. It is considered to be fast, minimal, and robust (cf. R33). Both requests and responses are sent in JSON format.

The tasks of the GM are to create and manage the GIs. Upon a first request of the CWAP to access a specific CTE, the GM creates the configurations of CTE container, GIDB container, GIDB volume, and virtual network between CTE and GIDB containers. Then it requests Docker to create the corresponding containers, volumes, and virtual networks. The configurations with the resulting Docker IDs of containers, volume, and virtual network are stored in the Group Management Database. Next, the GM instructs Docker to start the CTE and GIDB containers and initializes the CTE with respect to the group, the requesting group member, the text document, and the token of the user session. In case of an error the administrator of the CLP is informed; otherwise, the GM updates the Reverse Proxy configuration by adding a so-called `webpath` for the newly created CTE. Finally, a URL containing the host address of the RP, the `webpath` to the respective CTE, and the token of the user session is returned to the CWAP.

Upon a following request of a CWAP to access a specific CTE, the GM checks whether the CTE and GIDB containers are running. If not, the GM instructs Docker to start the CTE and GIDB containers, and the GM updates the Reverse Proxy configuration by adding a `webpath` for the started CTE. In both cases, the GM updates the CTE with respect to the requesting group member and user session token. Finally, a URL containing the host address of the RP, the `webpath` to the respective CTE, and the token of the user session are returned to the CWAP.

To minimize the usage of hardware resources, the GM implements a watchdog pattern (e.g. [75]) that automatically stops GIs when they are not used. For this purpose, the GM creates one watchdog per GI. The watchdog uses the `HTTP-Client-Library Axios` [76] of Node.js to check if the respective CTE (Etherpad Lite) is in use by examining the Etherpad Lite statistics page. If it is not in use, the GM stops this CTE container and the corresponding GIDB container using Docker. To increase robustness (cf. R33), the watchdog periodically checks (currently 30 seconds) if the respective CTE and GIDB containers are running. If not, the watchdog uses Docker to start the respective containers. Degradation of responsiveness (cf. R32) of the CE due to too many GIs running on one host is avoided since resources are only occupied by GIs that are currently in use. The shutdown of unused GIs increases robustness, because errors related to the CTE or GIDB of the stopped GI are avoided by preventing unintended execution beyond their use (cf. R33). To further increase robustness, the GM ensures that when a CTE container is stopped, the CouchDB replication feature is used to synchronize the respective GIDB with the FDB, before the GIDB is stopped. To allow the GM to interact with the GIDB, the Node.js package Nano [77] is used.

When a problem arises that is unsolvable for the GM (e.g., docker-related errors, database conflicts, or errors during replication), it immediately informs the system administration by using the Telegram Messenger (see below) (cf. R34).

3) Group Management Database: The Group Management Database is implemented in a separate container for safety reasons. Communication between GM and GMDB is secured by a separate virtual network. The DBMS PostgreSQL [78] is used to store management data for the GIs, including for
each group: groupID, configurations of the CTE container, the GIDB container, the volume of the GIDB, the virtual network between the CTE and the GIDB containers, and the Docker IDs of the containers, volume, and virtual network.

4) Fusion Database: The Fusion Database is also implemented in a separate container for safety reasons. Communication between GM and FDB, as well as between FDB and GIDB, is secured by separate virtual networks. The FDB is implemented using the DBMS CouchDB due to its replication feature, which the GM uses to synchronize the respective GIDB with the FDB when stopping the CTE, before its GIDB is stopped.

5) Telegram Messenger: The Telegram Messenger is used to inform the system administration when a problem arises that cannot be solved automatically by the GM (cf. R34). The Node.js package Telegraf [79] is used by the GM to send text messages to a Telegram channel [80]. The channel is set up by the administration of the CE to receive messages over the messenger, e.g. on the smartphone. The administration of the CE specifies two environment variables of the GM, a chat ID and a security token, used to access the respective channel.

VII. EXPERIENCE OF DEPLOYMENT IN LARGE TEACHING COURSES

The collaborative learning platform described in this paper has been deployed so far in winter semester 2021/22, summer semester 2022, and winter semester 2022/23 in a course on introduction to scientific work in the bachelor’s degree program of psychology at the FernUniversitaet in Hagen (the German Distance Learning University). In this course, teachers formed student groups of eight members each to collaboratively write a summary of a given research article.

The teacher divided the summarizing task of the research article into four parts (cf. Figure 5): (1) the theory part, (2) the methods part, (3) the results part, and (4) the discussion part. For each part, the teacher created a dedicated Collective Writing Activity. The students of a group were instructed to write each part collaboratively in a form of reactive writing. The students had six weeks to complete the summaries. In the first three weeks, the theory part and the method part had to be written followed by the results and discussion part in the last three weeks. Finally, all parts had to undergo peer review.

In winter semester 2021/22, only one Text Production Activity for each part was used without the Peer Review Activity using the Collaborative Learning Platform. In the summer semester 2022 and winter semester 2022/23, each part was reviewed by another peer student group. Here, each student in the review group could annotate passages of the text to be reviewed and answer the peer review questionnaire. The questionnaire, prepared by the teacher, contained simple questions that were to be answered on a rating scale (e.g., "Are the results briefly summarized?", "Is the submitted text structured and do its parts build on each other logically?", "Does the linguistic style follow the presented guidelines?"). Therefore, in each CWA exactly one Text Production Activity was defined, followed by one Peer Review Activity. The two CWAs in both three-week periods used the same start and end dates for each activity, allowing concurrent work. After the six-week period ended, the students could not make changes anymore, and the peer-review began followed by assessment. Due to a scheduled written exam at the end of the course, the teachers did not use the ability to give summative feedback. Only active participation was considered a necessary requirement for a successful completion of the collaborative summary writing task.

Teachers assigned the peer review groups to the writing groups randomly using the Collaborative Writing Activity Plug-In. The peer review provided the students with formative feedback as teachers only gave summative feedback after assessment. Due to time constraints, no final TPA was added after the PRA to allow students to improve their text.

In the Preparation Phase of the course, the teachers created the respective Collaborative Writing Activities and manually assigned students to the Moodle course groups. An informed consent for using their data for both, performing the collaborative summary writing task and for research purposes was obtained by students and teachers before using the platform.

A. Fulfillment of requirements

The fulfillment of the functional requirements identified in Section III has already been discussed in Sections V and VI. Teachers were able to create Moodle course groups for a large number of enrolled students (see Table II). The Collaborative Writing Plug-in automatically created a virtual workspace for each group and assigned the group members to it. To provide each group with its own Group Instance, the Group Management component provided a dedicated pair of Etherpad Lite and CouchDB (Group Instance Database) instances for each participating group. To minimize the use of hardware resources, the Group Management component automatically started and stopped the Group Instance components when they were used or not used, respectively. The Nginx Reverse Proxy was able to distribute the various user accesses to the different Etherpads with sufficient performance. For the large number of up to 2,343 students (see Table II), this approach worked well. As shown in Figure 6, user activities led to a maximum of 800 messages processed per minute (red line) without performance problems when executing all containers of the Collaboration Environment on a single server hardware.

In case of performance problems, the described architecture allows for the distribution of containers across several servers (nodes). This shows that the necessary scalability has been achieved (cf. R31).
Table II shows the semester-based statistical data of Collaborative Summary Writing activities. The number of students enrolled in the project course varies strongly between winter (~2,000 to 2,400 students) and summer semesters (~350 students) due to high school graduation cycles, as most students begin their studies in the fall after completing high school in the summer. Because students are not required to successfully complete the course and can freely choose in which semester they take the written exam, it is also typical that not all enrolled students actively participate in the course. A student is considered participating if the student performed at least one activity in the Collaborative Text Editor (e.g., login, editing, or commenting). Depending on the semester, the percentage of participation varies between just over 50% in the summer semesters and up to 86% in the winter semesters. 11 teachers supervised 2,343 students in the winter semester 2021/22 to 333 in winter semester 2022/23, resulting in a ratio of 213 students per teacher and demonstrating the need for platform support.

The platform proved to be reliable and fully operational during its use in the three semesters. It was consistently accessible and no major problems were detected or reported by the students.

A small number of students initially had problems accessing the activity-related group documents in the Collaborative Text Editor (CTE), i.e., Etherpad Lite due to an ad blocker or strict browser policies that blocked cross-site authentication between the Moodle CWAP plug-in and the CTE. However, these problems could be solved in cooperation with the course teachers by recommending students to use different browsers or to disable the ad blocker.

In case of errors, the Group Management component was able to restart the respective Group Instances within seconds (see table II). The current state of each Group Instance

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>SEMESTER-BASED STATISTICS OF CW ACTIVITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participation and Supervision</td>
<td></td>
</tr>
<tr>
<td>no. of course enrolled students</td>
<td>2,243</td>
</tr>
<tr>
<td>no. of participating students</td>
<td>2,024</td>
</tr>
<tr>
<td>% of participating students</td>
<td>86.4%</td>
</tr>
<tr>
<td>no. of formed student groups</td>
<td>296</td>
</tr>
<tr>
<td>no. of participating groups</td>
<td>295</td>
</tr>
<tr>
<td>% of participating groups</td>
<td>99.7%</td>
</tr>
<tr>
<td>no. of supervising teachers</td>
<td>11</td>
</tr>
<tr>
<td>Sessions (threshold = 30min)</td>
<td></td>
</tr>
<tr>
<td>no. of student sessions</td>
<td>74,122</td>
</tr>
<tr>
<td>no. of grp. sessions (≥ 2 stud.)</td>
<td>64,361</td>
</tr>
<tr>
<td>percentage of group sessions</td>
<td>13.2%</td>
</tr>
<tr>
<td>no. of teacher sessions</td>
<td>31</td>
</tr>
<tr>
<td>Text changes</td>
<td></td>
</tr>
<tr>
<td>no. of logged chng. sets</td>
<td>3,475,429</td>
</tr>
<tr>
<td>added chars/chng. set (25%ile)</td>
<td>1</td>
</tr>
<tr>
<td>added chars/chng. set (50%ile)</td>
<td>up to 2</td>
</tr>
<tr>
<td>added chars/chng. set (95%ile)</td>
<td>up to 6</td>
</tr>
<tr>
<td>Communication &amp; Feedback</td>
<td></td>
</tr>
<tr>
<td>no. of chat messages</td>
<td>6,948</td>
</tr>
<tr>
<td>no. of annotations</td>
<td>2,908</td>
</tr>
<tr>
<td>no. of annotation replies</td>
<td>1,553</td>
</tr>
<tr>
<td>no. of completed review quest.</td>
<td>N/A</td>
</tr>
<tr>
<td>System</td>
<td></td>
</tr>
<tr>
<td>no. of automatic GI restarts</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Database was successfully backed up each time an Etherpad Lite instance was stopped. Tests indicated that the data could also have been recovered from the FDB to the respective GIDB. Therefore, the intended robustness (cf. R33) was achieved. Further tests showed that the Group Management component informs the system administration by telegram when problems arise that could not be solved by the Group Management component itself, fulfilling (cf. R34).

B. Usage Experiences
The high percentage of participating groups (97.8%) (cf. table II) indicates that students accepted the Collaborative Learning Platform. The percentage of participating students lies in the usual range for the respective course. The high number of sessions and logged text change sets implies actual usage of the system for writing. Since 98% of the text change sets consists of fewer than seven added characters, extensive copy-paste behavior could not be observed.

The sessions of the groups were identified on the basis of the log of the respective group’s Etherpad Lite. All consecutive activities of a certain user in the log, which were less than 30 minutes apart, are considered a single user session (e.g., [81], [82]). Group sessions were determined as the temporal overlap of individual sessions of users in the same group. They denote time intervals when multiple group members simultaneously worked on the text document in the same Etherpad Lite instance. Similarly, all non-overlapping student sessions of a group denote situations in which students worked on a text document individually. The low percentage of group sessions shows that students mainly used an asynchronous writing strategy with few synchronous group sessions. Overall, students used the CLP to produce the required summary following the collaborative writing process outlined in the instructions and specified as a sequence of CWAs.

The increasing number of teacher sessions (from 31 in winter semester 2021/22 to 333 in winter semester 2022/23) with a slightly decreasing number of teachers (11 in winter semester 21/22 to 9 in summer semester 2022 and winter
students, organization and facilitation of collaborative writing sessions or writing strategies.

comments per user, or process mining to, e.g., identify user descriptive statistics to describe, e.g., added characters or to, e.g., count nouns, spelling errors, topic coverage, using behavior by using natural language processing approaches strategies (e.g., the absence of massive copy-paste behavior need to align work, family time, and studying.

which can be expected from distance learning students who It is also visible that there were always some students working, sions during the six-week period of the collaborative summary work phases. Figure 7 shows the number of user-student ses-

sessions during the six-week period of the collaborative summary writing task. Not surprisingly, students are more active prior to the submission deadlines after three or six weeks, respectively. It is also visible that there were always some students working, which can be expected from distance learning students who need to align work, family time, and studying.

Data on text changes allowed to analyze text production behavior. For example, data on sessions allowed to analyze collaboration behavior such as timing and sequence of asynchronous and synchronous work phases. Figure 7 shows the number of user-student ses-

The feedback of all teachers shows that they regard the Collaborative Learning Platform as useful. No serious problems were experienced. Continued use of the platform is planned.

Teachers could send a request at any time to the CLP administrator to receive the text documents, annotations, and chats of the groups as well as the data of the Fusion Database to evaluate group interactions. This happened at the end of each semester mostly for research purposes.

From a researcher perspective, the platform provides facilities to conduct studies and collect data. Until today, several research groups have already used the platform to work on research questions related to social psychology, learning analytics, and collaborative writing patterns.

The data collected by the CLP enabled teachers and researchers to analyze collaborative writing behavior. For example, data on sessions allowed to analyze collaboration behavior such as timing and sequence of asynchronous and synchronous work phases. Not surprisingly, students are more active prior to the submission deadlines after three or six weeks, respectively. It is also visible that there were always some students working, which can be expected from distance learning students who need to align work, family time, and studying.

Data on text changes allowed to analyze text production strategies (e.g., the absence of massive copy-paste behavior as discussed earlier) or in future the analysis of text editing behavior by using natural language processing approaches to, e.g., count nouns, spelling errors, topic coverage, using descriptive statistics to describe, e.g., added characters or comments per user, or process mining to, e.g., identify user sessions or writing strategies.

VIII. CONCLUSIONS

In distance education programs with a large number of students, organization and facilitation of collaborative writing projects are challenging. Teachers need support for preparing and setting up, running, monitoring, and evaluating large numbers of collaborative writing groups.

Current approaches in education to support collaborative writing projects rely on the use of Web 2.0 applications, such as Wikis, Weblogs, Forums, or Collaborative real-time text editors (e.g., [11], [12], [24]), and are mostly limited to provide student groups with tools for collaborative text editing, a way to communicate and thus to coordinate their activities. Unfortunately, these approaches do not provide holistic support for the above-mentioned phases. During preparation and set up of a large-scale collaborative writing project, support for group creation and student assignment, specification of learning support for different learning phases, and providing each group with its own shared workspace including task instructions and learning materials are missing. During running and monitoring of the collaborative writing projects, current tools lack either support for shared synchronous text editing [13] or do not comply with requirements imposed by European and national regulations on security, privacy, and data protection, or lack suitable scalability and robustness [14].

The contribution of this paper includes the specification of a four-phase collaborative summary writing project, the specification of requirements on a collaborative learning platform aiming at support for large-scale collaborative writing projects, a conceptual architecture of a collaborative learning platform addressing these requirements, the description of the implementation of this conceptual architecture based on Moodle, Etherpad Lite, and Docker, and experiences from using the collaborative learning platform in three semesters with a maximum of 2024 students per semester. The presented solution makes it easier for teachers to create groups, automatically assigns members to the dedicated group workspaces, enables scalable and robust asynchronous, as well as synchronous text editing and group communication, respects the European information security and data protection requirements, helps to monitor the learning and writing process, and offers learning support in different learning and writing phases.

Our deployment of the system and the gained experiences show that the described Collaborative Learning Platform provides a scalable and robust environment for Collaborative Writing Projects which is accepted by teachers and students. Due to its robustness, scalability, and wealth of interaction data stored, it provides the basis for analysis of collaborative writing behavior, which is useful for evaluation by both teachers and researchers.

By the end of 2023, we plan to release the implementation of our developed platform as open source, accessible at https://github.com/mburchat/cwap, https://github.com/mburchat/ce.

In the future, we plan to analyze the impact of horizontal scalability by distributing group instances across different computer nodes. In addition, we are currently working on analyzing the writing and collaboration behavior of students. Furthermore, we plan to extend the collaborative learning platform with a dashboard for both teachers and students, to support monitoring and self-regulated learning. Lastly, we are exploring a Group Agent supporting teachers in monitoring and acting on problems that are detected in group work.
REFERENCES


Marc Burchart received the Bachelor of Arts in Business Administration from the South Westphalia University of Applied Sciences, Germany, in 2015 and the Master of Science in Practical Computer Science from the FernUniversität in Hagen, Germany, in 2020. He is a researcher in the Research Institute for Telecommunications and Cooperation e.V., Dortmund, Germany, and a research member of the FernUniversität’s Center of Advanced Technology for Assisted Learning and Predictive Analytics (CATALPA). His main research interests are in adaptive collaborative systems with a focus on collaborative writing in educational settings.

Joerg M. Haake received the PhD degree in Computer Science from the Technical University Darmstadt, Germany. He is professor of cooperative systems in the Faculty of Mathematics and Computer Science, FernUniversität in Hagen, Germany, and deputy scientific director of the university’s Center of Advanced Technology for Assisted Learning and Predictive Analytics (CATALPA). Current research activities comprise support for collaborative learning and working, adaptivity in collaborative learning and working environments, and support for interdisciplinary collaboration. He is editor-in-chief of the e-learning and education Journal (eleed) and director of the faculty’s institute for continued education.