VIDEOEDIT: AN EASY-TO-USE WEB-BASED VIDEO CREATION TOOL FOR THE CLASSROOM

Alexander Ohlei, Toni Schumacher, Michael Herczeg

Institute for Multimedia and Interactive Systems (IMIS), University of Luebeck (GERMANY)

Abstract

We are currently in the demanding process of the digital transformation in schools. This includes two major challenges. The first is the provision of the necessary hardware equipment and network connectivity for students and teachers. The second challenge is to find well-designed integrated software applications to support the learning processes inside and outside schools. We developed the Ambient Learning Spaces (ALS) platform as a didactic infrastructure for schools. In comparison to Learning Management Systems (LMS), ALS focuses on self-directed learning inside and outside school. The pervasive ALS platform consists of multiple interrelated teaching and learning applications (modules) that can be accessed through stationaries (e.g. PCs, multi-touch walls, dome projections) and mobiles (e.g. wearables, smartphones, tablets). During the development of the ALS platform, we studied how students and teachers can be enabled to create high quality media content with low effort. In schools we observed that almost all students had access to mobile phones, which allow them to record images, audio and video. To enable students to process these recordings, we developed VideoEdit, a web-based tool. We analyzed scenarios for video creation and processing in connection with diverse learning applications and implemented the video editing functions identified as mandatory. The VideoEdit application supports mobile upload functionality, so students can upload recorded files directly into the ALS platform without a need for USB cables or the like. Once the files are uploaded, they find an easy to understand set of functions such as cutting and merging files, adding new audio tracks, adding text overlays and some effects. Once the cloud-based rendering process is initiated, the resulting video file will become automatically available to all ALS applications through our cloud-based backend system NEMO (Network Environment for Multimedia Objects). Using the NEMO backend, all ALS frontend applications can access the same multimedia repository with device-specific rendering of image and video formats and resolutions. NEMO stores information in semantic structures using different databases to support search and discovery of media and related information available. In this publication, we present the architecture, design, and evaluation results of VideoEdit.

Keywords: Ambient Learning Spaces, Integrated Media Infrastructures, Didactic Infrastructures, System Architecture, Video Editing.

1 INTRODUCTION

In the current process of digital transformation in schools two major challenges exist. One is the provision of the hardware equipment and network connectivity for students and teachers. Many schools are not yet equipped with the necessary technical infrastructure but are in the progress of setting up Wi-Fi and electronic whiteboards. Another challenge is to find well designed integrated software applications to support the learning process inside and outside schools. In this area, mostly isolated educational applications are currently in use. We did extensive research on the development of integrated didactic infrastructures for schools and developed the *Ambient Learning Spaces (ALS)* platform as a possible system solution ([1], [2], [3]). ALS consists of multiple learning applications (modules) that can be accessed through stationaries (e.g. PCs, multi-touch walls, dome projections) and mobiles (e.g. wearables, smartphones, tablets).

During workshops in schools, when we evaluated the use of ALS applications, we discovered the need for an integrated tool that supports the video creation and editing process. Previously we used different tools for editing videos such as Windows Movie Maker or Adobe Premiere, but these systems pose many problems such as the need for a software installation before the main work can start. Furthermore, it is usually difficult to transfer self-recorded video files from students' devices onto the local PC due to restrictions with the USB connectivity and saving files on the computer in use. Sometimes students just do not find the files they previously copied to the computer. The video editors themselves usually also offer too many features and export options so it might take some time until students are able to use them properly. In many schools the computers get cleared upon shutdown

ISBN: 978-84-09-27666-0

and all the files are gone the next day, when the students like to continue their work. Therefore, we developed VideoEdit to make the process of video creation and editing less cumbersome.

2 AMBIENT LEARNING SPACES

In our research project *Ambient Learning Spaces (ALS)* we develop a digitally enriched body- and space-related learning environment for schools and museums ([1], [2], [3]). In these environments, users can interact with applications running on multiple interconnected digital devices in physical space to learn collaboratively. In the context of ALS, *Body- and Space-related Human-Computer Interaction* combined with *Cross-Device Interaction (XDI)* [4] builds the conceptual foundation.

In the backend, the *Network Environment for Multimedia Objects (NEMO)* is the platform for all ALS applications ([5], [6]). Inside NEMO, all media are stored in a context-specific semantic model. Additionally, logic elements for ownership, storytelling, and interactivity are stored in NEMO as well. Therefore, all ALS applications on mobile and stationary devices can access NEMO as one contextualized media and logic repository.

We designed the ALS system in a modular way so that schools can use multiple frontend applications developed in previous work to enable students and teachers to access the information layers through different media types such as tangible or mobile media as well as interactive multi-touch displays. In this publication we describe *VideoEdit* as a tool to easily create and edit videos that can be used in the ALS teaching and learning applications.

2.1 ALS System Architecture

The system architecture of the ALS learning applications consists of three logical subsystems (Fig. 1) [3]. The first subsystem includes the data management (ALS-Portal) and media editing systems (ALS-Tools), which can be used by students and teachers. The second subsystem is the NEMO backend system, which stores media files in a semantic database. The third subsystem consists of the frontend applications.

To add and remove media files the web-based ALS-Portal can be used. All data uploaded through the ALS-Portal is stored inside the connected NEMO instance. The ALS-Tools can be used to edit media and create new media from existing sources, such as videos and 3D objects. Media files created with the ALS-Tools are automatically stored inside NEMO and can also be accessed through the ALS-Portal. The communication with NEMO runs through https-secured web-service connections. The learning apps are creating or displaying media files.

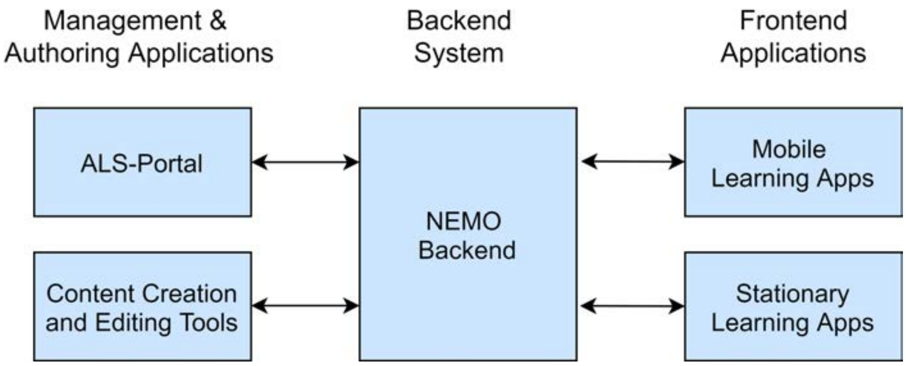

Figure 1. ALS System Architecture.
All parts of the ALS infrastructure are connected by a web service infrastructure.

2.2 ALS-Portal

The *ALS-Portal* is used to create and edit organizational information for all ALS learning applications through a regular web browser. After logging into the ALS-Portal, it presents a list of ALS applications that are available for the user account. Because of its architecture as a web application, there is no installation or configuration necessary on the devices that students or teachers use to access the ALS-Portal. It can be accessed using any kind of operating systems such as Windows, Linux, macOS, Android or iOS. Using this setup, the portal can easily be accessed on all network devices inside the school. Furthermore, the ALS-Portal can be configured to allow access through the internet. Students can access it from their homes or any connected place outside school to work on their task and projects. Each student has access to his or her own media repository and can share these media files with others so they can work on projects collaboratively.

2.3 The Integrated ALS Backend Platform NEMO

The *Network Environment for Multimedia Objects (NEMO)* is a service-based architecture (Fig. 2) providing:

- User Authentication (accounts and profiles for users and groups)
- Usage Tracking (anonymous tracking for user studies)
- Media Conversion (automatic data conversion for 2D, 3D, and video footage)
- Cognitive Services (automatic tagging and classification)
- Central Logic (store and retrieve media content considering ownership and context)
- Databases (relational, RDF, and file storage)

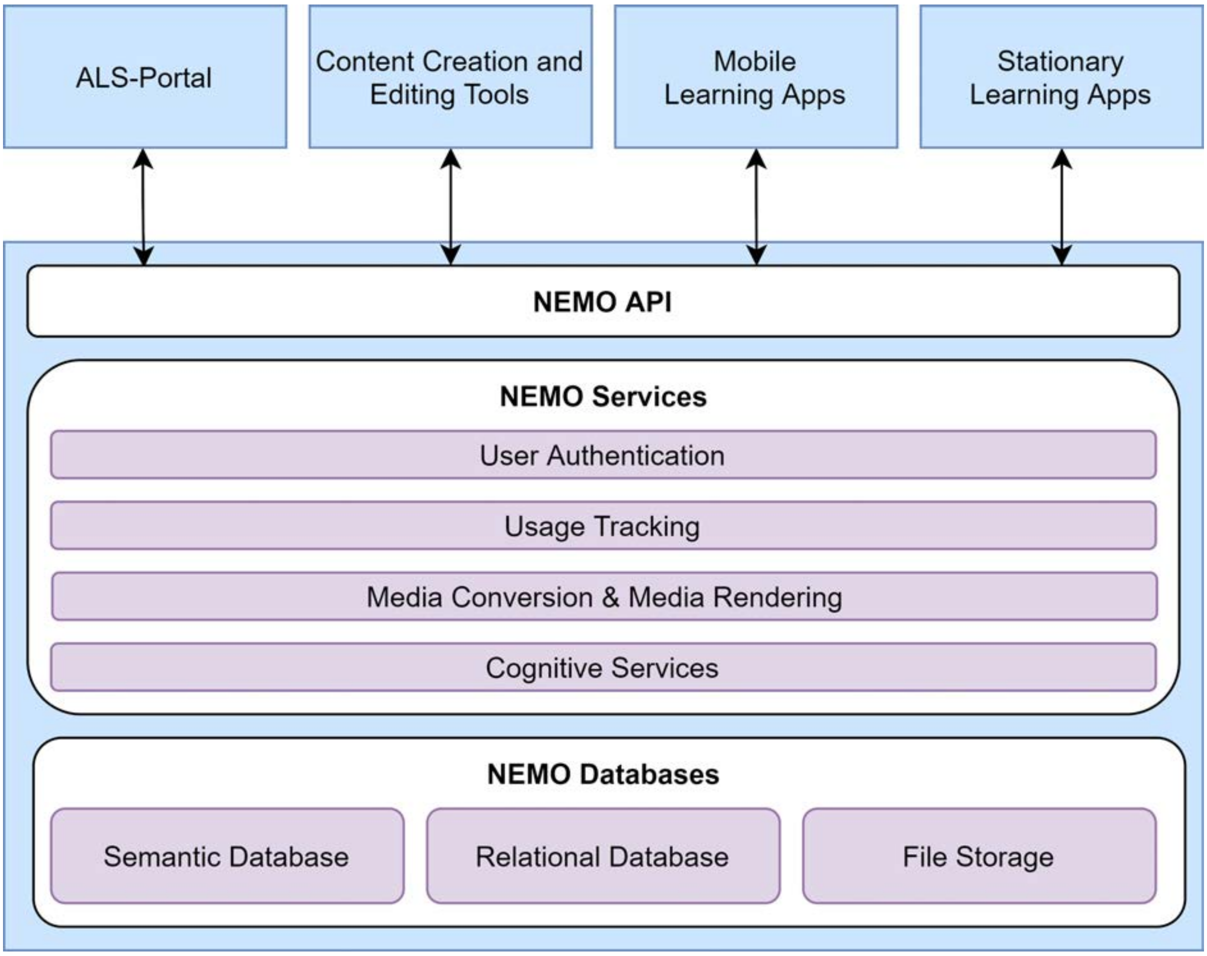

Figure 2. ALS System Architecture.

Media created and bound to objects representing the learner's worlds are defining a semantic web and a model of the application. Several media in different formats for different devices can represent such an object. NEMO enables the learners to reuse media created or collected through one frontend application in other frontend applications for related but different learning contexts. This is important to create a growing repository of media that can be abstracted and enriched to *knowledge representations* through annotations (attributes), links (relations), and classifications (ontologies).

3 FEATURES NEEDED FOR VIDEO EDITING

The ALS environment includes multiple mobiles and stationary frontend applications mainly used by the learners and in some cases by the teachers, to initiate, and organize the learning process. All of these applications are connected to NEMO which enables object and media sharing between the ALS applications and tools. In the following sub-sections, we will outline a possible workflow for school projects using ALS applications. We will also describe the ALS frontend applications that present video files and list the features students and teachers require for editing video files in a way that the resulting videos fit as well as possible into the application domains.

3.1 Workflow in Project-Oriented Teaching with ALS

During project-oriented teaching, the teacher defines the topic for the class, based on the curriculum. Depending on the topic, the teacher can choose between the different ALS applications that students may apply during the project.

In the next step, the students supported by the teacher create a concept for the planned project. A part of the concept is the informational content and media types that the students shall work with. Possible media types are text, images. 3D models, audio or video recordings taken from books, the internet, or recorded from physical spaces mainly outside of school.

Depending on the project topic, students will create media in or outside school using certain ALS applications. In the next section we will describe some of the most used ALS learning modules that deal with video footage, the focus of this paper.

During the review phase of the project, media files that are of good quality and fit to the planned concept are selected by the students. After the review, the students can prepare the media files for their projects. In case the video footage selected by the students, which is either recorded by themselves or provided by other sources, they might need to cut, merge or edit the video files. This can be done using VideoEdit, the video editor presented in this paper.

After preparation of the media files, students can finish the project. The finished projects can then be delivered and presented in the class or any other context inside or outside school.

3.2 MoLES

A mobile ALS application that supports the collection of information and media is the mobile app *Mobile Learning Exploration System (MoLES)* ([7], [8], [9], [10]). The app allows creating tasks (challenges) to be solved by students inside and preferably outside school. They may use their own smartphones or tablets [11] to solve these tasks by collecting information in form of various media like text, photo, audio or video.

The information and media collected will be tagged and automatically transmitted to NEMO. If there is no connectivity in the field, the media will be stored offline and synchronized with NEMO as soon as possible. When the tasks have been solved, the students return to school to review, structure and edit the information and media collected, which is already stored in a project repository on NEMO waiting to be inspected, selected, structured and processed for the presentations.

3.3 InfoGrid

InfoGrid is an augmented reality (AR) app for educational purposes ([11], [12]). Teachers or students can create new *InfoGrid* projects using the ALS-Portal through a web browser at school or home. Using the ALS-Portal students or teachers can upload images as AR targets which will later be recognized by the InfoGrid app. For any AR target inside the project, an AR overlay such as an image, video, 3D overlay or more complex interaction structures, *Dynamic Asset Collections* [13], can be assigned.

Upon start, the InfoGrid mobile app presents a list of available tours. Once the user selects a project, the app downloads the necessary information to the mobile device. As soon as the download is complete, the user can see an optional intro regarding the selected project. When the mobile device is facing one of the prepared targets in the physical space, the app renders the augmentation through the display of the mobile device. In case of audio and video augmentations, the app streams the audio and video data from the connected NEMO repository.

3.4 Rendering 3D Models from Video and Images

In case students like to work with 3D objects, they can use 3D models from public repositories or create them on their own using a special ALS module, the *NEMO Converter 3D (NOC3D)* of the NEMO media conversion layer, as well as the *3DEdit Authoring System*. This conversion module can generate 3D objects from of existing photos or videos of physical 3D objects ([14], [15], [16]).

The videos used for NOC3D should be prepared well to contain good reference images of the recorded object that shall be transformed to a 3D model. Therefore, editing the video footage will usually be necessary.

3.5 InteractiveWall and InteractiveTable

Additionally to the mobile applications there is a flexible setup in number and size of stationary large screen systems like the *InteractiveWall (IW)* ([17], [18], [19]), often placed in school foyers and other public school spaces or in the classroom. The IW is a multi-touch interactive screen with a capability of up to 50 simultaneous touch events like when several students are interacting with the content at the same time. The IW comes in a variation as an *InteractiveTable* with the additional capability of recognizing not only touch events, but also *Tangible Objects (Fiducials)* placed on the surface.

The IW provides so-called *MediaGalleries* that show raw collections (media from the field) or collections of already selected, grouped, tagged and classified media bound to objects of the learning domain. From these, the students may create presentations or documentations under supervision and guidance of their teachers or parents at home.

Learning applications supporting semantic modeling about knowledge entities *(SemCor)* and chronological structures *(TimeLine)* connected to media objects can be used on the IW as well [20]. These applications can contain video material that has been created previously by the students.

3.6 HyperVid

HyperVid is a web-based hyper-media system with which students and teachers can link video fragments to a hypervideo (Fig. 3). The finished hypervideo can be presented on the InteractiveWall. It can also be integrated in webpages, e.g. the homepage of the school. HyperVid promotes networked thinking and supports joint learning with time-based multimedia [21].

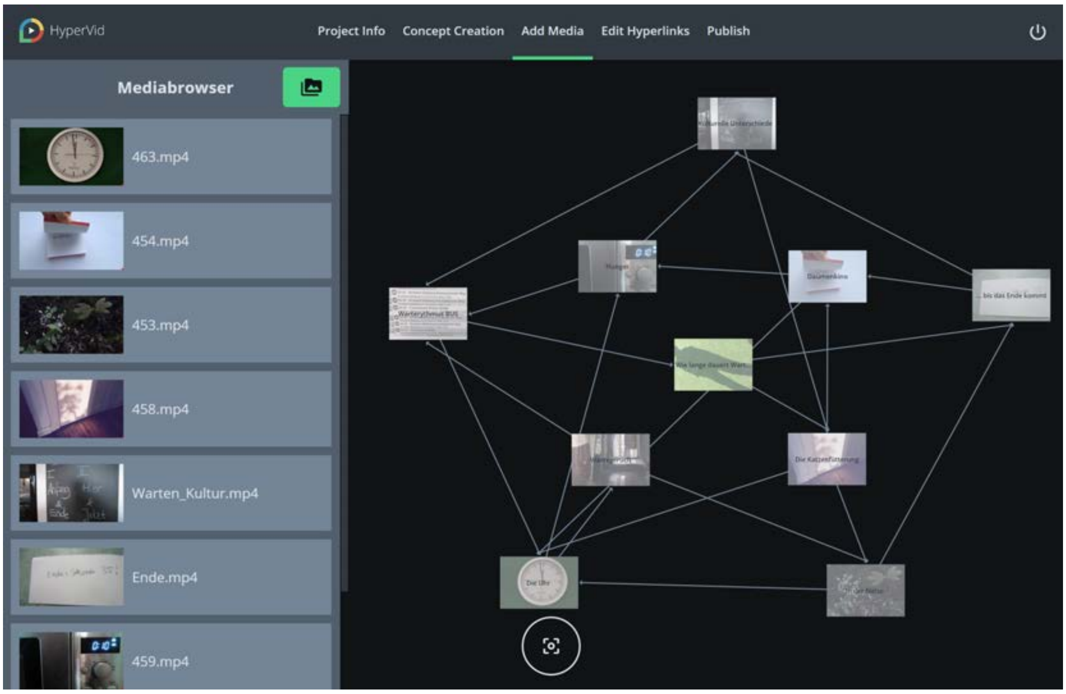

Figure 3. HyperVid frontend showing the creation of a new hypervideo consisting of multiple interlinked video fragments.

3.7 Video Editing Features for ALS

During workshops and formative evaluations with the ALS applications, we identified several features for a video editor within ALS (named VideoEdit) to support the work of students and teachers.

In case students want to work with video files, these files will either originate from external sources like the internet or from self-recording with mobile phones. Therefore, there needs to be an easy-to-use function to import the files into VideoEdit. It is also necessary to support multiple video file formats so that all major types of mobile devices are supported. The rendering of the video files should be server-based so the application can even be used on thin and mobile clients that have little computing power. Thin clients like tablets might have touch-only input, so VideoEdit needs be able to be controlled by mouse input or by using touch input.

When manually recording video files with a mobile phone, usually in the beginning or at the end of the recording, there will be some shaking or unwanted noise. To remove these unwanted parts of the file it is necessary to be able to cut the video. It is also sometimes necessary to merge multiple video files, to add a new audio track, or to add text to sections of the video files. In case recordings have been recorded too quiet or too loud it is necessary to increase or decrease the volume.

Sometimes students only have pictures available as material to work with. Then it needs to be possible to create videos from these images or from combinations of images and videos and combine them with self-recorded audio files. When working with images, effects like zooming in make the presentation more vivid. A smooth transition between images and videos also improves the video quality and should also be available.

Finally, the resulting videos files should be available to all ALS applications without the need for uploading them again. The resolution of the video file should adapt to the technical requirements of the target application and target device. Mobile applications like InfoGrid for instance only need a low resolution of videos while the MediaGallery or HyperVid on a UHD-based InteractiveWall need video files in a high quality. It should also be possible to download the videos so they can be placed on the school website or other platforms.

3.8 Feature List

We created a list of required features based on our findings as a software specification for the implementation of VideoEdit:

- file import of local files and mobile upload functionality;
- file encoder to support the import of all common video formats;
- cut and merge functionality of videos and images;
- adding, positioning, deleting audio tracks, and adjusting the volume;
- video effects: smooth transition between files, zoom-in effect on images and cropping of borders inside the movies, changing the volume of the video between 0 and 200%;
- mouse and touch input for the frontend;
- adding text overlays for introduction texts or subtitles;
- server-based rendering with HD and SD output resolutions and export.

4 VIDEOEDIT

Based on the features required we developed VideoEdit in a user-centered design with multiple formative evaluations in schools.

4.1 System Architecture

VideoEdit is a web-based application that can be used on Chrome and Edge web browsers. The frontend was developed with HTML, CSS, and JavaScript; the backend with ASP.NET. The NEMO Video Encoder was developed with C#, .NET and uses the FFmpeg library for video encoding and decoding. The NEMO Data Storage stores the raw media files and the output files.

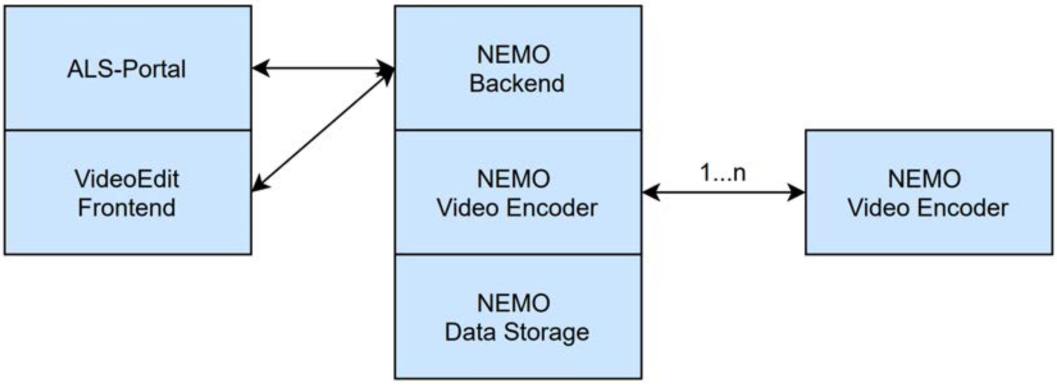

Figure 4. VideoEdit system architecture

The VideoEdit frontend takes parameters and raw files for the editing process from the user and sends them via secured web services to the NEMO backend. All the rendering will be done on the rendering server. Multiple rendering servers can be installed for higher load. All rendering servers can be configured to access the same NEMO data storage repository and pick unfinished projects to be rendered. The number of rendering servers can be dynamically changed at any time (Fig. 4).

After the cloud-based rendering process is complete, the resulting video file will become automatically available for all ALS applications through the cloud-based backend system NEMO.

4.2 Frontend Design

The frontend of VideoEdit is divided into three areas (Fig. 5). The first area is for file handling (upper left); the second is the preview area that also includes the options menu (upper right); the third is the timeline area, where all components of the new media file are displayed (bottom).

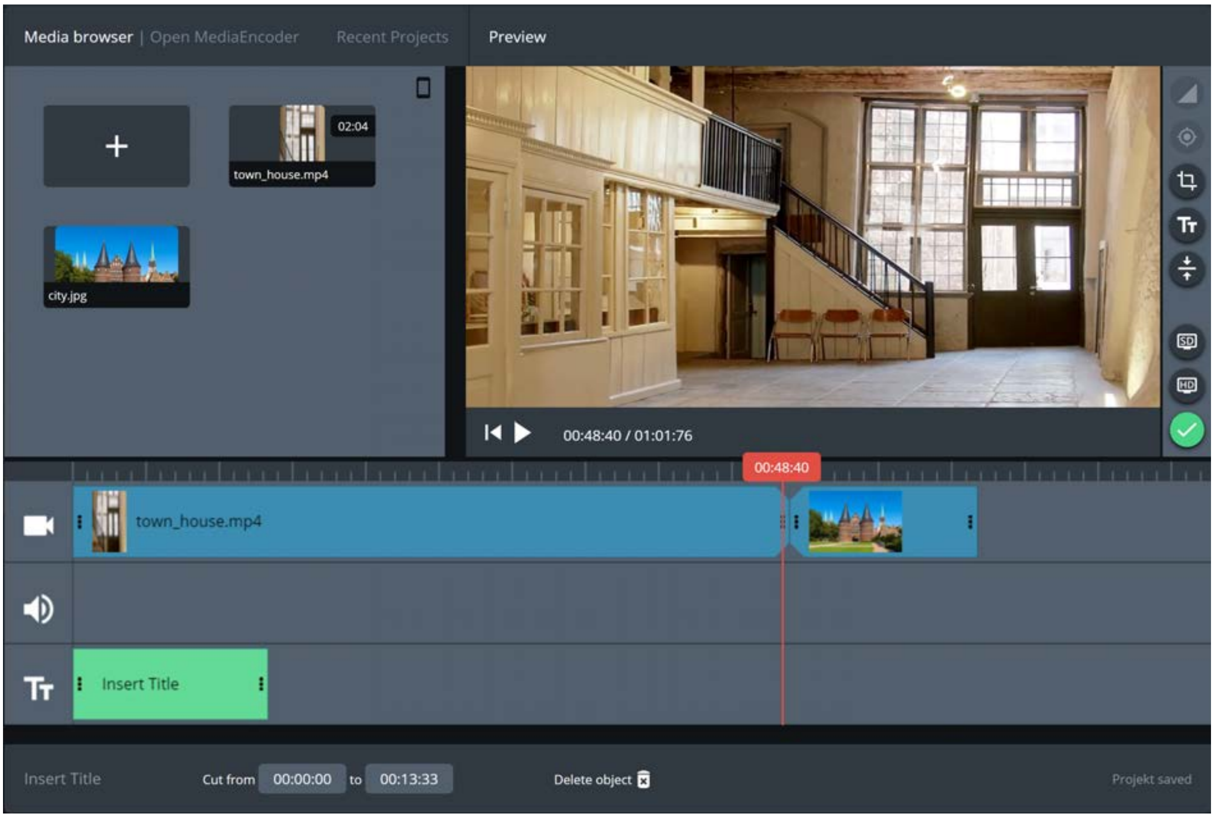

Figure 5. VideoEdit showing the creation of a new video from of a video merged with an image.

When starting VideoEdit, it first asks for a username and password. After logging in, a dialogue box is shown, which provides the options of loading a previous project, setting up a new project or starting the tutorial for the system.

When creating a new project, the user has first to upload the files that the system shall process. The system natively supports .mp4 and .mov video files, .mp3 audio files and .jpg or .png image files. In case the file the user wants to work with exists in a different format, the MediaEncoder can be used to transform all file types, which are supported by the FFmpeg library, into the .mp4 format (Fig 6 left). The user can open the MediaEncoder by clicking the corresponding link in the upper left area of VideoEdit.

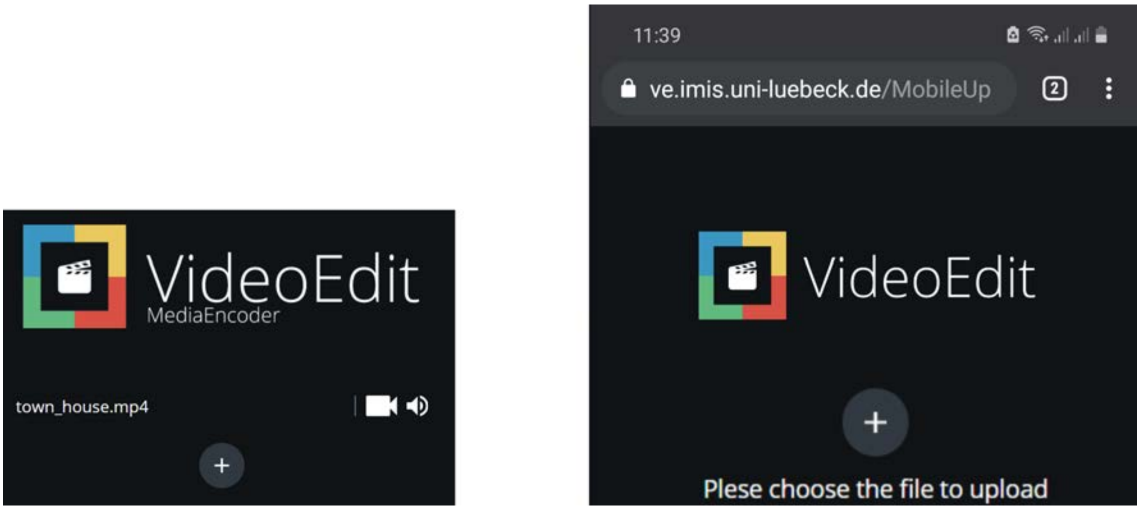

Figure 6. (left) The MediaEncoder converts any video format into an mp4 file and creates a separate audio only mp3 file. (right) VideoEdit mobile upload. Files can be uploaded into the VideoEdit project from any mobile device.

Next to a regular file upload from the local device, VideoEdit supports mobile upload, so users can upload recorded files from their mobile devices directly into the ALS platform without a need for USB cables or the like. To upload the file, the user first needs to click on the mobile upload icon in the file handling area. Then a QR code will be displayed, which can be scanned on the mobile device. Scanning the QR code brings up a webpage (Fig. 6 right) on the device that allows uploading files into the project repository.

VideoEdit was implemented with a set of options that support the video creation process (Fig. 7). When button a is activated, a fading animation is added for the transition between two elements in the video timeline. Button b sets a focal point inside an image. The focal point creates a zoom animation towards this point, which can be used to animate still pictures. Button c crops black borders from the video mainly to support the creation of video files for augmented reality contexts that should exactly match existing target areas. Option d adds a text layer to the video. Button e toggles, where the text will be displayed. The top area of the video, the center or the bottom can be selected.

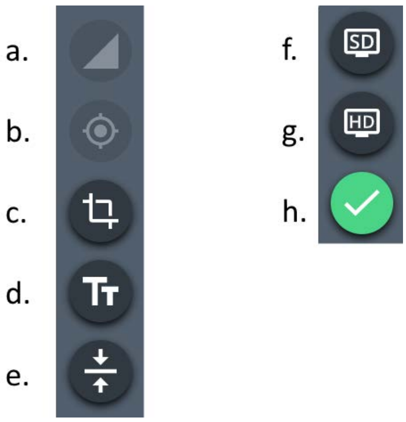

Figure 7. VideoEdit menu buttons; (left) options menu, (right) rendering and download menu.

After editing the project, all changes will be saved automatically. When the user finished setting up the video file, the rendering process can be initiated by clicking the button h. An indication window informs the user that the rendering is in progress and the browser window can be closed. The file will automatically become available once the rendering process has finished. In case the user wants to make changes to the file that was recently rendered, the same button can be used to re-render the video with new settings. Once the rendering process is completed successfully the user has the additional option to download the resulting video in SD (720p) resolution with button f or in HD (1080p) resolution with button h.

4.3 Evaluations of VideoEdit

VideoEdit was developed in a user-centered development process. Multiple formative evaluations with teachers and students have been carried out. We also used the feedback we collected during workshops to extend and improve the functionality of the system. Several hundred videos have already been created and rendered using VideoEdit in ALS. The rendering speed scales with the performance of the server. On a system running Windows 10 Professional with 64 GB RAM, AMD Ryzen 9 3950X CPU and RTX2080 graphics card the rendering process of a video file into multiple formats (HD and SD resolution) with a length of 1 minute takes approximately 23 seconds.

5 SUMMARY

Ambient Learning Spaces (ALS) is a prototypical teaching and learning environment for a wide variety of learning contexts inside and outside of school.

In this paper, we presented the technical architecture and the frontend design of VideoEdit. We found that the easy-to-use system opens new ways of creating and editing media image and video-based media in schools and supports the digital transformation in schools. Connected to the NEMO semantic media repository allows the reuse of media in different contexts and for different interaction devices. In this integrated and pervasive teaching and learning environment, all created video files become available in any other ALS learning applications to be viewed or reused for new projects and media creations.

ALS is currently piloted for three schools and three museums, but has been made available to a larger number of institutions, who tried out the applications and authoring modules over several years. Improvements and extensions are ongoing.

ACKNOWLEDGEMENTS

We developed and evaluated the ALS system described in the ongoing research project “Ambient Learning Spaces” supported from 2009–2021 by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG). We also thank our school and museum project partners for their continuous support and their permission to do our field research in their institutions.

REFERENCES

[1] T. Winkler, F. Scharf, C. Hahn, and M. Herczeg, "Ambient Learning Spaces," in *Education in a Technological World: Communicating Current and Emerging Research and Technological Efforts* (A. Méndez-Vilas, ed.), Formatex Research Center, pp. 56-67, Badajoz, Spain, 2011.

[2] M. Herczeg, T. Winkler, and A. Ohlei, "Ambient Learning Spaces for School Education," in *Proc. of iCERi 2019*, pp. 5116–5125, IATED, 2019.

[3] M. Herczeg, A. Ohlei, and T. Schumacher, “Ambient Learning Spaces: A Connecting Link between Digital Technologies and Computer-Supported Pedagogy,” in *Proc. of INTED 2021*, same issue, IATED, 2021.

[4] F. Scharf, C. Wolters, M. Herczeg, and J. Cassens, “Cross-Device Interaction”, in *Proc. of AMBIENT 2013*, pp. 35-41, IARIA, 2013.

[5] B. Feldner, S. Günther, F. Schmitt, T. Winkler, and M. Herczeg, “A Dolphin is a Dolphin is a Dolphin? Multimedia Enriched Learning Objects in NEMO,” in *Proc. of ICALT 2009*, pp. 29–31, IEEE Computer Society, 2009.

[6] S. Lob, J. Cassens, M. Herczeg, and J. Stoddart, "NEMO - The Network Environment for Multimedia Objects," in *Proc. of the First Intl. Conf. on Intelligent Interactive Technologies and Multimedia*, pp. 245–249, ACM, 2010.

[7] S. Günther, T. Winkler, and M. Herczeg, "Mobile Learning with Moles: A Case Study for Enriching Cognitive Learning by Collaborative Learning in Real World Contexts," in *Proc. of ED-MEDIA 2008*, pp. 374–380, AACE, 2008.

[8] T. Winkler, M. Ide-Schöning, and M. Herczeg, "Mobile Co-operative Game-based Learning with Moles: Time Travelers in Medieval," in *Proc. of SITE 2008*, pp. 3441–3449, AACE, 2008.

[9] T. Winkler, S. Günther, and M. Herczeg, "Moles: Mobile Learning Exploration System," in *Proc. of SITE 2009*, pp. 348–351, AACE, 2009.

[10] T. Winkler and M. Herczeg, "The Mobile Learning Exploration System (MoLES) in Semantically Modeled Ambient Learning Spaces," in *Proc. of IDC 2013,* pp. 348–351, ACM, 2013.

[11] M. Herczeg, A. Ohlei, and T. Schumacher, "Ambient Learning Spaces: BYOD, Explore and Solve in Physical Context," in *Proc. of iCERi 2020*, pp. 7979-7989, IATED, 2020.

[12] A. Ohlei, D. Bouck-Standen, T. Winkler, and M. Herczeg, "InfoGrid: Acceptance and Usability of Augmented Reality for Mobiles in Real Museum Context," in *Mensch und Computer 2018 - Workshopband*, pp. 340–344, de Gruyter, 2018.

[13] A. Ohlei, T. Schumacher, and M. Herczeg, „An Augmented Reality Tour Creator for Museums with Dynamic Asset Collections", in *Proc. of the AVR 2020*, LNCS 12243, pp. 15-31, Springer, 2020.

[14] D. Bouck-Standen, A. Ohlei, V. Daibert, T. Winkler, and M. Herczeg, "NEMO Converter 3D: Reconstruction of 3D Objects from Photo and Video Footage for Ambient Learning Spaces," in *Proc. of AMBIENT 2017,* pp. 6–12, IARIA, 2017.

[15] D. Bouck-Standen, A. Ohlei, S. Höffler, V. Daibert, T. Winkler, and M. Herczeg, "Reconstruction and Web-based Editing of 3D Objects from Photo and Video Footage for Ambient Learning Spaces," in *Intl. J. on Advances in Intelligent Systems*, vol. 11, no. 1/2, pp. 94–108, 2018.

[16] A. Ohlei, L. Bundt, D. Bouck-Standen, and M. Herczeg, "Optimization of 3D Object Placement in Augmented Reality Settings in Museum Contexts. Augmented Reality, Virtual Reality, and Computer Graphics," in *Proc. of AVR 2019 Part II,* 208–220, Springer, 2019.

[17] T. Winkler, M. Ide, and M. Herczeg, "InteractiveSchoolWall: A Digital Enriched Learning Environment for Systemic-Constructive Informal Learning Processes," in *Research Highlights in Technology and Teacher Education*, pp. 117–126, AACE, 2012.

[18] T. Winkler, M. Ide, C. Hahn, and M. Herczeg, "InteractiveSchoolWall: A Digitally Enriched Learning Environment for Systemic-Constructive Informal Learning Processes at School," in *Proc. of ED-MEDIA 2014*, pp. 2527–2537, AACE, 2014.

[19] T. Winkler, D. Bouck-Standen, M. Ide, A. Ohlei, and M. Herczeg, "InteractiveWall 3.1 - Formal and Non-Formal Learning at School with Web-3.0-based Technology in Front of Large Multi-touch Screens" in *Proc. of ED-MEDIA 2017*, pp. 1317–1326, AACE, 2017.

[20] M. Herczeg, T. Schumacher, and A. Ohlei, "Ambient Learning Spaces: Discover, Explore and Understand Semantic Correlations," in *Proc. of iCERi 2020*, pp. 7990-7999, IATED, 2020.

[21] T. Winkler, M. Ide, and M Herczeg, "The Use of Hypervideo in Teacher Education", in *Proc. of AUCEi Intl. Conf. 2013,* Florida, USA, AUCEi, 2013.